
Mapbender Documentation

Release 3.3.0

The Mapbender Team

Mar 11, 2024

CONTENTS

1	Installation	1
2	Mapbender Quickstart: First steps with Mapbender	19
3	Elements	37
4	Backend	235
5	Customization / Tips and tricks	259
6	FAQ - Frequently Asked Questions	285
7	Architecture	291
8	Development	303
9	Version history	321

INSTALLATION

We provide installation instructions for different operating systems. The installation with Symfony describes how you can use Symfony build-in mechanisms to take a quick look at Mapbender.

Installation and update guide for Mapbender users:

1.1 Installation on Ubuntu/Debian

Mapbender is shipped with a preconfigured SQLite database which includes preconfigured applications (the database is located under **<mapbender>/app/db/demo.sqlite**).

Hint: For productive use PostgreSQL is recommended. You can find the necessary configuration steps in chapter *Optional > Mapbender Deployment on PostgreSQL*.

1.1.1 Requirements

- PHP >= 7.4
- **Apache installation with the following modules activated:**
 - mod_rewrite
 - libapache2-mod-php
- **PostgreSQL Installation**
 - It is recommended to use a PostgreSQL database for Mapbender.
 - It is recommended to create a database user to access the Mapbender database.

Nginx can also be used as web server (this will not be discussed in detail here).

1.1.2 Preparation

Installation of mandatory PHP extensions:

```
sudo apt install php-gd php-curl php-cli php-xml php-sqlite3 sqlite3 php-apcu php-intl openssl php-zip  
↳ php-mbstring php-bz2
```

- Please check the [FAQ - Frequently Asked Questions](#) for further PHP settings.

1.1.3 Unpack and register to web server

Download the current Mapbender version and unzip it into `/var/www/mapbender` or a different location:

```
wget https://mapbender.org/builds/mapbender-starter-current.tar.gz -O /var/www/mapbender-starter-  
↳ current.tar.gz  
tar -xzf /var/www/mapbender-starter-current.tar.gz -C /var/www  
mv $(ls -d /var/www/*/ | grep mapbender) /var/www/mapbender/
```

1.1.4 Configuration Apache 2.4

Create the file `/etc/apache2/sites-available/mapbender.conf` with the following content:

```
Alias /mapbender /var/www/mapbender/web/  
<Directory /var/www/mapbender/web/>  
Options MultiViews FollowSymLinks  
DirectoryIndex app.php  
Require all granted  
  
RewriteEngine On  
RewriteBase /mapbender/  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteRule ^(.*)$ app.php [QSA,L]  
</Directory>
```

Activate the site and reload Apache:

```
a2ensite mapbender.conf  
service apache2 reload
```

1.1.5 Directory rights

```
sudo chown -R :www-data /var/www/mapbender  
  
sudo chmod -R ug+w /var/www/mapbender/app/logs  
sudo chmod -R ug+w /var/www/mapbender/app/cache  
sudo chmod -R ug+w /var/www/mapbender/web/uploads  
  
sudo chmod -R ug+w /var/www/mapbender/app/db/demo.sqlite
```

1.1.6 First steps

The Mapbender installation can now be accessed under **http://[hostname]/mapbender/**. User data by default:

username: "root", password: "root"

Troubleshooting is available via the following command (must be executed in the application directory):

```
app/console mapbender:config:check
```

Hint: Please note that config:check will use the php-cli version. The settings may be different from your webserver PHP settings. Please use `php -r 'phpinfo();'` to show your PHP webserver settings.

Congratulations! Mapbender is now set up correctly and ready for further configuration. Find Information about the first steps with Mapbender in the [Mapbender Quickstart](#).

1.1.7 Optional

LDAP

To use the optional LDAP-connection, following PHP-LDAP-extension is required:

```
sudo apt install php-ldap
```

Mapbender installation with PostgreSQL

Configuration of PostgreSQL database for productive use:

Requirements: - configured PostgreSQL database - database for Mapbender configuration - PostgreSQL database user to access the database with create database right

Installation PHP-PostgreSQL driver

```
sudo apt install php-pgsql
```

Configuration of database connection (app/config/parameters.yml):

```
database_driver: pdo_pgsql
database_host: localhost
database_port: 5432
database_name: mapbender
database_path: ~
database_user: postgres
database_password: secret
```

For further information: [YAML Configuration \(Configuration and Application files\)](#).

Initialisation of the database connection:

```
cd /var/www/mapbender
app/console doctrine:database:create
app/console doctrine:schema:create
app/console mapbender:database:init -v
bin/composer run reimport-example-apps
```

Create root user for access:

```
app/console fom:user:resetroot
```

Find further information in *[Details of the configuration of Mapbender](#)*

Mapbender installation with MySQL:

Similar to configuration with PostgreSQL.

Installation MySQL driver:

```
apt install php-mysql
```

Following parameters (parameters.yml) need to be adapted:

```
database_driver: pdo_mysql
database_port: 3306
```

To initialize your database connection, see PostgreSQL.

1.2 Installation on Windows

For a quick installation (e.g. on a test system), use the MS4W-Installer (<https://ms4w.com/download.html>).

Read on for a detailed description on a productive system.

Mapbender needs a database to store information. The Mapbender download package contains a SQLite database that is ready to use. For production we recommend the use of a PostgreSQL database.

1.2.1 Requirements

- PHP NTS >= 7.4 (<https://windows.php.net/download/>)
- Apache installation (<https://www.apachelounge.com/download/>, run as service with these modules):
 - mod_rewrite
 - mod_fcgid
- PostgreSQL Installation (<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>)
 - It is recommended to use a PostgreSQL database for Mapbender.
 - It is recommended to create a database user to access the Mapbender database.

Nginx can also be used as web server, but it will not be discussed in this manual.

1.2.2 Configuration PHP

Unzip the Zip archive, for example under `c:\php`.

- It needs to be checked if the following variables (`php.ini`) are set correctly:

```
sys_temp_dir
upload_tmp_dir
date.timezone
```

- the path from PHP-bin directory to the PATH variable (Windows environment variable) needs to be set
- activate the required PHP extensions in the `php.ini` configuration file:

```
# php.ini
extension=php_curl
extension=php_fileinfo
extension=php_gd
extension=php_intl
extension=php_pdo_pgsql
extension=php_pdo_sqlite
extension=php_pgsql
extension=php_openssl
extension=php_mbstring
extension=php_zip
extension=php_bz2
```

- Please check the [FAQ - Frequently Asked Questions](#) for further PHP settings.

1.2.3 Extract Mapbender and register to web server

Download the current Mapbender version (<https://mapbender.org/builds/mapbender-starter-current.zip>) and unzip it into `c:\mapbender\`

1.2.4 Configuration Apache

A subfolder “`conf.d`” must be prepared in the directory `<apache>/conf`.

Add the following code at the end of file `httpd.conf`:

```
# Include directory conf.d
Include "conf/conf.d/*.conf"
```

Create file `<apache>\conf\conf.d\mapbender.conf` with:

```
Alias /mapbender c:/mapbender/web/
<Directory c:/mapbender/web/>
    Options MultiViews FollowSymLinks
    DirectoryIndex app.php
    Require all granted

    RewriteEngine On
    RewriteBase /mapbender/
    RewriteCond %{REQUEST_FILENAME} !-f
```

(continues on next page)

(continued from previous page)

```
RewriteRule ^(.*)$ app.php [QSA,L]
</Directory>
```

Restart Apache.

1.2.5 mod_fcgid

Create file `<apache>\conf\conf.d\fcgi.conf` with:

```
LoadModule fcgid_module modules/mod_fcgid.so

FcgidInitialEnv PHPRC "c:/php/"
FcgidInitialEnv PATH "c:/php;C:/WINDOWS/system32;C:/WINDOWS;C:/WINDOWS/System32/Wbem"
FcgidInitialEnv SystemRoot "C:/Windows"
FcgidInitialEnv TEMP "C:/WINDOWS/TEMP"
FcgidInitialEnv TMP "C:/WINDOWS/TEMP"
FcgidInitialEnv windir "C:/WINDOWS"

FcgidPassHeader Authorization
FcgidIOTimeout 1200
FcgidConnectTimeout 1200
FcgidBusyScanInterval 1200
FcgidBusyTimeout 1200
FcgidErrorScanInterval 1200
FcgidIdleScanInterval 1200
FcgidIdleTimeout 1200
FcgidZombieScanInterval 1200
FcgidMaxProcesses 1000
FcgidOutputBufferSize 64
FcgidProcessLifeTime 3600
FcgidMaxRequestsPerProcess 10000
FcgidMinProcessesPerClass 0
FcgidFixPathinfo 0
MaxRequestLen 200000

<Files ~ "\.php$" >
    Options Indexes FollowSymLinks ExecCGI
    AddHandler fcgid-script .php
    FcgidWrapper "c:/php/php-cgi.exe" .php
</Files>
```

1.2.6 Configuration PostgreSQL

The configuration of the Mapbender database is done in the file `app/config/parameters.yml`.

For more information on the database configuration, see [YAML Configuration \(Configuration and Application files\)](#).

```
database_driver: pdo_pgsql
database_host: localhost
database_port: 5432
database_name: mapbender
database_path: ~
```

(continues on next page)

(continued from previous page)

```
database_user: postgres
database_password: secret
```

Open the windows shell and initialize the database connection with the following commands:

```
cd c:\mapbender
php.exe app/console doctrine:database:create
php.exe app/console doctrine:schema:create
php.exe app/console mapbender:database:init -v
php.exe bin/composer run reimport-example-apps
```

To administrate Mapbender you need a user. Create root user for access:

```
php.exe app/console fom:user:resetroot
```

Find further information in [Details of the configuration of Mapbender](#)

1.2.7 First steps

The Mapbender installation can now be accessed under **[http://\[hostname\]/mapbender/](http://[hostname]/mapbender/)**.

Check if the alias is working

- <http://localhost/mapbender/>

username: "root", password: "root" (if you use the SQLite database shipped with Mapbender)

Troubleshooting is available via the following command (must be executed in the application directory):

```
php.exe app/console mapbender:config:check
```

Hint: Please note that config:check will use the php-cli version. The settings may be different from your webserver PHP settings. Please use `php -r 'phpinfo();'` to show your PHP webserver settings.

Further information can be found at [app/console mapbender:config:check](#).

Congratulations! Mapbender is now set up correctly and ready for further configuration. Find Information about the first steps with Mapbender in the [Mapbender Quickstart](#).

1.3 Update Mapbender to a newer Version

To update Mapbender you have to do the following steps:

- get the new version from <http://mapbender.org/builds/>
- save your configuration files (parameters.yml and config.yml) and your old Mapbender (files and database)
- replace the new files
- merge your configuration files (check for new parameters and changes)
- update your Mapbender database

- copy the screenshots from your old Mapbender version from `/web/uploads/` to the folder `/web/uploads` of your new installation
- Templates: If you are using your own template, you have to compare your scripts with the new scripts (are there any changes?)
- print templates: if you use your own print templates: copy them back to `app/Resources/MapbenderPrintBundle/templates/`.
- Import the demo applications either via `bin/composer run reimport-example-apps` or via the web administration
- At [Installation on Ubuntu/Debian](#) under the section **Unpack and register in your Web-Server** you can see how the config file for the Apache Alias should look like
- That's all! Have a look at your new Mapbender version

Hint: Please follow the migration instructions on specific Mapbender versions, see the [Migration Guide](#).

1.3.1 Update Example for Linux

Have a look at the steps as commands

```
# Download the new version
wget http://mapbender.org/builds/mapbender-starter-current.tar.gz /tmp/build_mapbender/
tar xzf /tmp/build_mapbender/mapbender-starter-current.tar.gz

# save the old version
mv /var/www/mapbender /var/www/mapbender_save

# get the code of the new version
cp -R /tmp/build_mapbender/mapbender-starter-v3.3.2 /var/www/
mv /var/www/mapbender-starter-v3.3.2 /var/www/mapbender

# copy your old configuration files to the new version
cp /var/www/mapbender_save/app/config/parameters.yml /var/www/mapbender/app/config/parameters.yml
cp /var/www/mapbender_save/app/config/config.yml /var/www/mapbender/app/config/config.yml

# manual step
# merge parameters.yml, config.yml and if used mapbender.yml
# if you use screenshots: copy the screenshots from the old version back to mapbender/web/uploads
# if you have individual templates: merge the templates with the new Mapbender version
# if you use your own print templates: copy them back to mapbender/app/Resources/MapbenderPrintBundle/
↪ templates/

# change the accessrights and owner of the files
sudo chmod -R ugo+r /var/www/mapbender
sudo chown -R :www-data /var/www/mapbender

# Update your Mapbender database
cd /var/www/mapbender/
app/console doctrine:schema:update --dump-sql
app/console doctrine:schema:update --force

# Import the mapbender demo applications
bin/composer run reimport-example-apps
```

(continues on next page)

(continued from previous page)

```
# Update the symbolic links
app/console assets:install web --symlink --relative

# change the access rights and group of the files
sudo chmod -R ug+r /var/www/mapbender
sudo chown -R :www-data /var/www/mapbender

# You have to set write permission to app/cache and app/logs.
sudo chmod -R ug+w /var/www/mapbender/app/cache
sudo chmod -R ug+w /var/www/mapbender/app/logs
sudo chmod -R ug+w /var/www/mapbender/web/uploads
```

1.3.2 Update Example for Windows

```
# Download the new version http://mapbender.org/builds/

# Save the old version (files and database)

# Copy the configuration files (parameters.yml and config.yml) to your new Mapbender version.
# You have to check the configuration files for changes (new parameter, other changes)

# Call the app/console commands with php.exe
# You have to open a windows console to send the commands
c:
cd mapbender

# Update your Mapbender database
php.exe app/console doctrine:schema:update --dump-sql
php.exe app/console doctrine:schema:update --force

# Notes for MS4W users:
#   - be sure to first execute setenv.bat to properly set the required paths for PHP
#   - you may have to also pass the extension you need, at the commandline, for example:
#       php -d extension=C:\ms4w\Apache\php\ext\php_pdo_pgsql.dll app/console
↪ doctrine:schema:update --dump-sql

# Import the applications from mapbender.yml to your database to get to know about the latest
↪ developments
php.exe bin/composer run reimport-example-apps

# Export files to the web-directory
php.exe app/console assets:install web

# Delete your cache and the logdateien at mapbender/app/cache und mapbender/app/logs

# if you use screenshots: copy the screenshots from the old version back to mapbender/web/uploads
# if you have individual templates: merge the templates with the new Mapbender version
# if you use your own print templates: copy them back to mapbender/app/Resources/MapbenderPrintBundle/
↪ templates/
```

Installation guide and details for developers:

1.4 Installation of Mapbender using Symfony built-in webserver

Mapbender is built on the [Symfony](#) Framework and therefore can make use of the [Symfony built-in webserver](#). This setup allows a quick test of Mapbender without an integration into an external webserver.

Hint: The use of the Symfony built-in webserver is not suitable for production environments.

In this document we assume that the SQLite database is used.

- Please check the installation documentation for [Linux](#) respectively [Windows](#).
- Download the current Mapbender version <https://mapbender.org/builds/>.
- Extract Mapbender in an arbitrary directory.
- Start the Symfony webserver.

```
app/console server:run
```

The command runs a local web server. By default, the server listens on 127.0.0.1 address and the port number is automatically selected as the first free port starting from 8000.

Now Mapbender is available on the local machine with the address <http://127.0.0.1:8001/>. Please note that Mapbender runs in the developer mode per default.

If you run the command several times Symfony will choose the next available port.

You also could change the default address and port by passing them as an argument:

```
app/console server:run 127.0.0.1:80002

[OK] Server listening on http://127.0.0.1:8002

// Quit the server with CONTROL-C.

[Mon Jan 31 15:56:57 2022] PHP 7.4.3 Development Server (http://127.0.0.1:8002) started
```

1.5 Git-based installation

We have documented the installation from the Git-sources directly in the README.md of the Mapbender-Starter Git-repository:

<https://github.com/mapbender/mapbender-starter/blob/master/README.md>.

We have created a Contributing Guide for our developers which describes the necessary steps to develop with Mapbender and also explains how to install the software via Git. This documentation is directly available in the Git-Repository:

<https://github.com/mapbender/mapbender-starter/blob/master/CONTRIBUTING.md>.

Overview: The Git-based installation is for the development of Mapbender or for project-specific customization that you can save in a cloned Mapbender-Starter. There you can store your own printing-templates, applications or own bundles. With Composer you are able to load the modules Mapbender, FOM, OWSPProxy, Digitizer and all the other versions that you need. During an update of Mapbender you have to compare the status of your Mapbender-Starter. Basic customizations can be done by checking the versions in `composer.json`.

1.5.1 Quick primer on using Git

On using Mapbender branches and Git, please look at the [Contributing Guide for Developers](#).

1.6 Details of the configuration of Mapbender

1.6.1 Configuration steps

In the following, we will describe the configurational steps of Mapbender a bit further. Configuring your Mapbender installation consists of the following six steps:

- Creating the database
- Creating the database schema
- Copying the bundles' assets to the public web directory
- Creating the "root" user
- Initializing the database
- Loading the applications of the mapbender.yml to your database

All that can be done using the console utility provided by [Symfony](#), on which the Mapbender framework is built upon. There's a mayor caveat though you should understand before continuing:

Note: The console utility will write files in the app/cache and app/logs directories. These operations are made using the user permissions of whatever user you're logged in with. This is also true for the app/db directory and the SQLite database within. When you open the application from within the browser, the server PHP process will try to access/write all these files with other permissions. So make sure you give the PHP process write access to these files. See last step below.

Note: Important: The following steps assume that you are in the directory above the app directory (notice that for git installation that means mapbender/application/, else mapbender/).

```
cd mapbender/  
or for git based installation  
cd mapbender/application
```

Adapting the configuration file

Database connection parameters are stored together with some more configuration parameters in the file app/config/parameters.yml.

More Information: [YAML Configuration \(Configuration and Application files\)](#).

Creating the database

Symfony can attempt to create your database, this works of course only if the configured database user is allowed to do so. Call the console utility like this:

```
app/console doctrine:database:create
```

Creating the database schema

Symfony will create the database schema for you:

```
app/console doctrine:schema:create
```

Copying the assets bundles

Each bundle has its own assets - CSS files, JavaScript files, images and more - but these need to be copied into the public web folder:

```
app/console assets:install web
```

Alternatively, as a developer, you might want to use the symlink switch on that command to symlink instead of copy. This will make editing assets inside the bundle directories way easier.

```
app/console assets:install web --symlink --relative
```

Creating the administrative user

The first user - which has all privileges - must be created using the command:

```
app/console fom:user:resetroot
```

This will interactively ask all information needed and create the user in the database.

Alternatively, there is a silent mode you can use, if you want to use a script to install Mapbender and don't want to be asked for all parameters:

```
app/console fom:user:resetroot --username="root" --password="root" --email="root@example.com" --silent
```

Initialize the database

Initializing the database can be done using the command:

```
app/console mapbender:database:init
```

Importing applications from application/app/config/applications

It is possible to (re-)import applications from the applications folder into the database with the command:

```
bin/composer run reimport-example-apps
```

1.6.2 Configuration files

The configuration files are located under **app/config**.

Find more information in: *YAML Configuration (Configuration and Application files)*.

1.6.3 Production and Development environment and Caching: app.php and app_dev.php

Mapbender provides two environments: a production environment for the general operation and a development environment in which the application can be tested. This concept follows the “environments” in the *Symfony framework*.

The production environment is called with the URL <http://localhost/app.php>, the development environment with the URL http://localhost/app_dev.php. The call with app_dev.php is and should only be available from localhost.

There are differences in the behaviour of app.php and app_dev.php:

- The cache mechanism of the development environment behaves differently: Not all files are cached, thus code changes are directly visible. Therefore, the usage of app_dev.php is always slower than the production environment.

In detail, the development environment of Mapbender does not cache the CSS, JavaScript and Translation files, among others.

The production environment caches all these files and puts them into the app/cache folder.

- The development environment gives out error messages and stack traces to the user interface. The production environment logs them into the file app/log/prod.log.
- The development environment shows the Symfony Profiler. This tool logs things that are important for developers but are not supposed to be visible for common users.

The directory app/cache contains the cache files. It contains directories for each environment (prod and dev). But the mechanism of the dev-cache, as described before, behaves differently.

If changes of the Mapbender interface or the code are made, the cache directory (app/cache) has to be cleared to see the changes in the application.

1.7 Migration Guide

Hint: This page gives migration instructions on specific Mapbender versions. For general tips on updating, see the [Update Mapbender to a newer Version](#) page instead.

1.7.1 Migration to Mapbender 3.3.4

- Please update the database schema. Run:

```
app/console doctrine:schema:update --force
```

- Next, you need to configure CSRF tokens to be used during login to the **csrf_token_generator** form_login section in security.yml as follows:

```
form_login:
  check_path: /user/login/check
  login_path: /user/login
  csrf_token_generator: security.csrf.token_manager
```

- For productive environments, it is important to install an SSL certificate. After that, set the parameters.cookie_secure variable in your parameters.yml to true. This ensures that the Login cookie is only transmitted over secure connections.

1.7.2 Migration to Mapbender 3.3

- Make sure you have PHP >= 7.4 or PHP 8.x
- Provide a backup of your database.
- Update your database schema to 3.3 with `app/console doctrine:schema:update --force`
- config.yml: Please note that in the doctrine connection configuration variables must be set with quotes, for example '%database_driver%'
- CAUTION: Please note that the eye at application is used from (3.2.x) onwards to make the application available for the anonymous user (public access). Before 3.2.x, the eye/checkbox at security was used to publish an application.

To update from 3.2.x to 3.3.x should be quite easy.

Note: If you update from a version < 3.2, you have to follow the steps described at the *Migration to Mapbender 3.2* section below.

New features

Styling

Styling is now possible via variables that can be passed to your application.

- [Mapbender](#)
- Create your own scss file, see this [Demo Bundle SCSS Example](#).
- Modify your template - add function `getSassVariablesAssets` and refer to your scss file see, see this [Demo Bundle Template Example](#).

Sketch

- Sketch now supports to draw a circle with a defined radius. You draw the circle first and then edit the circle and define a radius.
- Sketch now allows to define colors to be offered to draw. You can also activate a color picker

FeatureInfo

- FeatureInfo Highlight now allows to style the fill and stroke and opacity

1.7.3 Migration to Mapbender 3.2

You can migrate older Mapbender installations to Mapbender 3.2.

Check the [Update Mapbender to a newer Version](#) Guide.

- Make sure you have PHP $\geq 7.1.0$ and PHP < 8
- Provide a backup of your database.
- Update your database schema to 3.2 with `app/console doctrine:schema:update --force`
- CAUTION: Please note that the eye at application is from (3.2.x) used to make the application available for the anonymous user (public access). Before the eye /checkbox at security was used to publish an application.

Some elements may not work after the update and may need a closer look.

Update map_engine_code

If it makes sense, update all applications to `map_engine_code` current.

```
Update mb_core_application set map_engine_code = 'current';
```

SearchRouter

In the Workshop bundle, you can find a [Demo](#)

1. deprecated empty: use placeholder instead
2. For text and choice you have to define the full class-path.

You also find information at [Best Practices Page](#)

You can update the configuration with the following SQL.

```
Update mb_core_element set configuration =
replace(configuration,'s:6:"choice",'s:53:"Symfony\Component\Form\Extension\Core\Type\ChoiceType")
where class = 'Mapbender\CoreBundle\Element\SearchRouter';

Update mb_core_element set configuration =
replace(configuration,'s:4:"text",'s:51:"Symfony\Component\Form\Extension\Core\Type\TextType")
where class = 'Mapbender\CoreBundle\Element\SearchRouter';

Select configuration from mb_core_element where class = 'Mapbender\CoreBundle\Element\SearchRouter';
```

3. For choice: Please note that key or value are passed flipped that means value and the key- see also [Best Practices Page](#)

```
choices:
  Bonn - this is the value not the key: Bonn
  Cologne - this is the value not the key: Cologne
  Siegburg - this is the value not the key: Siegburg
```

SimpleSearch

SimpleSearch element was improved. You can now define the projection of the result that comes from the Solr Service. Mapbender will then transform the result to the projection of the map.

SimpleSearch Supports Nominatim, Photon from version 3.2.5 - see workshop demo applications

1. Define sourceSrs in your SimpleSearch definition. If not defined, the default ("EPSG:4326") should be chosen. You can find an example at [this Configuration file](#)

```
sourceSrs: 'EPSG:25832'
```

2. query_ws_replace: From version 3.2.8 on, set query_ws_replace or modify the code as described in [this issue comment](#)

```
query_ws_replace: +
```

BaseSourceSwitcher

Please note that on start of an application, all WMS are activated where the root-Layer is activated.

Before 3.2, it was possible to activate all Basesource and only the first WMS was visible on start.

Template / CSS

CSS change. Plus, there will be a big redesign in backend and frontend in the upcoming versions.

- Check the workshop Bundle for the changes
- Define your template as desktop-template

Digitizer

Digitizer is available for Mapbender $\geq 3.2.2$. The new Digitizer Version is 1.4. Some functionality is not updated to 1.4 already (e.g. cluster).

- See [list of deprecated features](#)
- See also [digitizer php file](#)
- You can find a demo in the [Workshop bundle](#)
- `maxResults` - is supported again to limit the number of features that are loaded to the application (if not defined all features will be used) (digitizer $\geq 1.4.9$)
- **For font definitions, see [issue 1308](#)**
 - `fontSize`: 38 definition without px
 - `labelXOffset`: 18 (not supported in 3.2.3)
 - `labelYOffset`: 18 (not supported in 3.2.3)
- **Types that are not supported in 3.2.4**
 - upload
 - select with multiselect
 - coordinates
- Clustering not implemented in 3.2.x
- Style definition is limited not all OL2 styles can be defined
- Support styling features with icons (interpret `externalGraphic`, `graphicWidth`, `graphicHeight` properties) (Mapbender $\geq 3.2.7$)
- Support data placeholder syntax in `externalGraphic` (e.g. `"/bundles/projectbundle/images/${type}.png"`) (Mapbender $\geq 3.2.7$)
- Save NULL for empty fields, works for int/float/decimal columns but not for double precision (Mapbender $\geq 3.2.6$, see [issue 1355](#))
- Save NULL for empty fields for text fields does not work. Mapbender saves `"` instead (see [issue 1385](#))
- supports `printable: true`

There is a new style called `unsaved`.

```
unsaved:
  strokeWidth: 3
  strokeColor: "#f0f0f0"
  fillColor:   "#ffff"
  fillOpacity: 0.5
  pointRadius: 6
```

(continues on next page)

(continued from previous page)

```
label: 'Neu - bitte speichern'  
fontColor: red  
fontFamily: 'Arial, Courier New, monospace'  
fontColor: red  
fontSize: 38  
fontWeight: bold
```

WMS Layer visibility

Make sure that your WMS provides a proper extent for all supported EPSG-codes (this is used and saved in table mb_wms_wmslayersource Spalten latlonbounds und boundingboxes). Else it can happen that a layer is not requested for the given extent of your map.

Sketch

Redlining was renamed to Sketch ($\geq 3.2.3$).

```
Update public.mb_core_element set class = 'Mapbender\CoreBundle\Element\Sketch',  
title = 'mb.core.sketch.class.title'  
where class = 'Mapbender\CoreBundle\Element\Redlining';
```

FeatureInfo

- showOriginal deprecated - parameter not available anymore (from 3.2.3).
- highlighting: true - new $\geq 3.2.3$ highlights the geometry if you have WKT integrated in the feature-info result - see [issue 1287](#) and also this [FeatureInfo blog post](#)

Print - Print queue

- Mapbender supports print queue
- see [Queue blog post](#)



MAPBENDER QUICKSTART: FIRST STEPS WITH MAPBENDER

Mapbender is a web based geoportal framework to publish, register, view, navigate, monitor and grant secure access to spatial data infrastructure services. Management interfaces empower administrators who need to maintain and categorize map and feature services and grant access to individuals, groups and other services. Mapbender is written from the ground up using modern web technologies. The foundation is laid by Symfony. On the client-side expect to find OpenLayers.

With this code base, we will continue the Mapbender idea of being a Geoportal framework. Key features of Mapbender are:

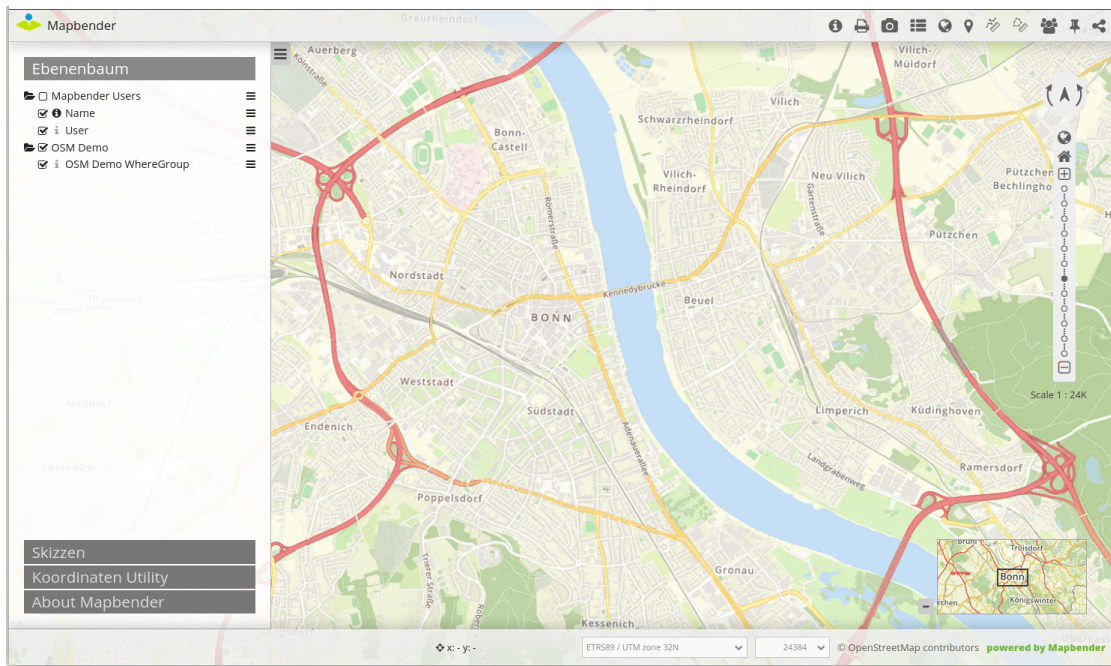
- Applications can be setup, configured and styled right from within the browser.
- Services (e.g. WMS) can be managed inside a service repository and linked to applications.
- Rights management are easy to maintain, for individual users and groups, whether you store them inside the database or in an LDAP.
- Search modules can be configured.
- Applications for digitalization can be setup.
- Mobile template can be used to provide applications for smartphones and tablets.

You will need nothing but a web browser for this quickstart.

This quickstart describes how to:

1. Start Mapbender
2. Create an application
3. Insert elements into an individual application
4. Configure sources
5. Manage users and groups
6. Use the rights management
7. Start an application at a defined position

This is how a Mapbender application can look like:



2.1 Installation

This quickstart explains the basics of Mapbender and serves as a quick introduction after your first successful installation. For the installation of Mapbender have a look at [Installation](#).

2.2 1. Start Mapbender

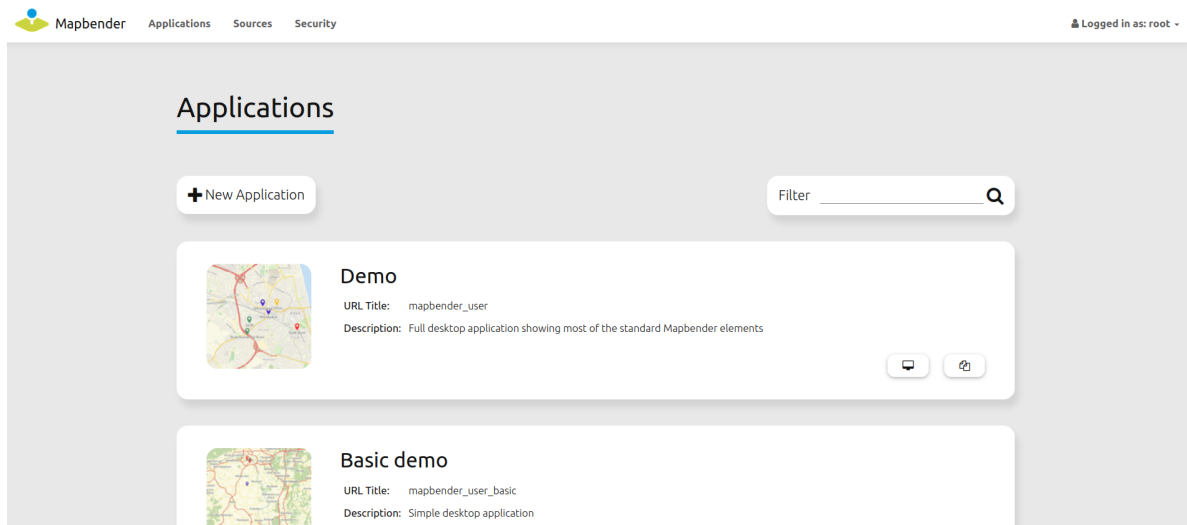
1. Choose Mapbender from the start menu (if a shortcut was already created) or visit <http://localhost/mapbender/app.php> (this address can be slightly different depending on how the Apache Alias was created in the file `/etc/apache2/sites-available/mapbender.conf`, more information at [Installation](#)).
2. The application should then appear in your browser window.

If you have any difficulties running Mapbender, please check whether your Apache web server and your PostgreSQL database are running without errors.

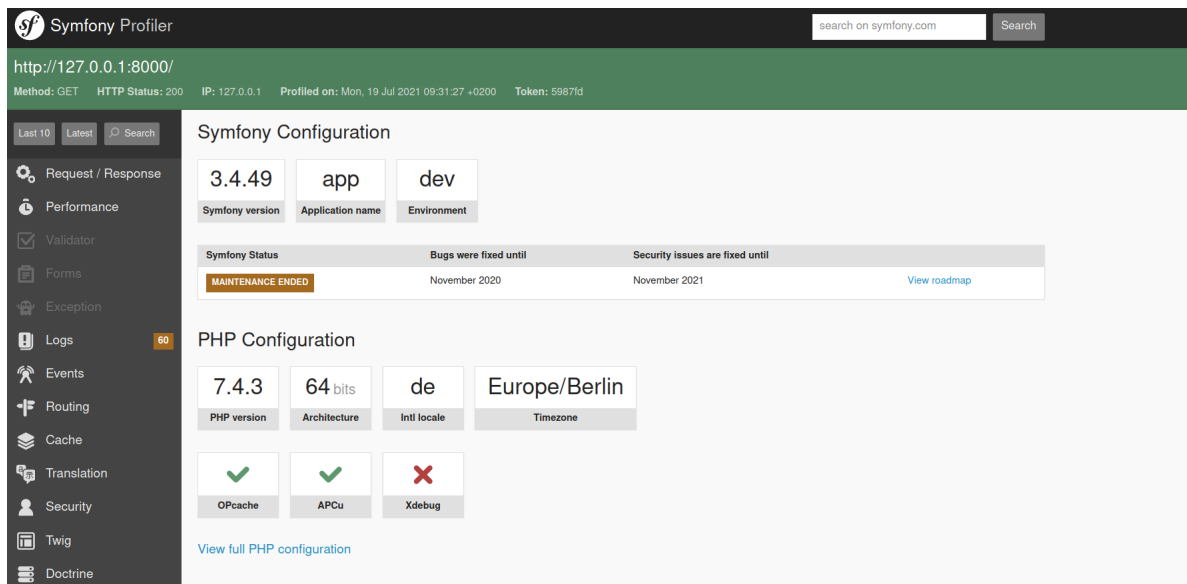
2.2.1 Start Mapbender in developer mode

Symfony offers a developer mode with lots of information about your application (logging, exceptions, database queries, memory usage, time and more). This mode is only available from localhost.


- Start the developer mode: http://localhost/mapbender/app_dev.php



- Have a look at the information that is offered in the developer mode.









2.2.2 Mapbender Backend

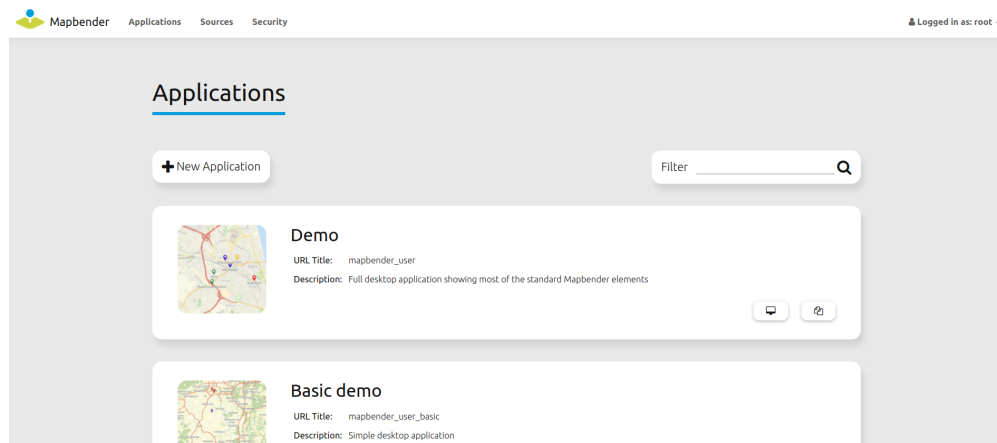
1. After a successful Mapbender startup, the application overview page in the backend will appear. The applications are listed with a screenshot, title, URL title and description.
2. You can open an application by click on the title, the screenshot or via the  button.
3. A log-in is required to gain access into Mapbender's administration backend. In order to do so, click on **login** at the top-right of the login page. You can login with the user that was generated during installation. This could be root with the password root - this is the default user and password that you get after installation of Mapbender. Please change the root password if you want to run a productive environment. Please don't delete the user root.

After a successful login you will be directed to the Mapbender administration backend.

2.2.3 Application overview

The application overview site displays a list of all available applications. The root user has access to the following functions:

- title, URL title and description
- preview screenshot for the application (if provided)
- filter textfield for application search
- option to create new applications
-  link to the application
-  button to duplicate the application
-  button to export the application
-  button to edit the application
-  button to make the application available for anonymous users
-  button to delete the application



2.3 2. Create an individual application

There are three different options to create an application:



An application can be created out of an already existing one. This can be done via a click on the button in the application overview. The application will receive the same title and URL title with the appendix “*imp*”. All previously defined elements and configurations will be transferred as well. Another possibility is the import of an application. Further information can be found under [YAML Configuration \(Configuration and Application files\)](#).






Furthermore, new applications can be created from scratch. The required steps are explained in the following:

1. Select the option “*New Application*” in the application overview.
2. After that, select a template in order to define the layout of your application. The options are: Fullscreen, Fullscreen alternative, Mapbender Mobile template. It is also possible to define your own template and assign it to a new application.

Tip: Please note that the style-, icon- and layout-configurations are set up via css- and twig-files. Read more about template generation under [How to create your own Template?](#).




1. Define a title, URL title and a description (optional). Title and URL title can be identical. However, the URL title has to follow the usual URL syntax.
2. A thumbnail can be uploaded as well. It will appear next to the application title on the application overview page. For this, select “*Select File*” below the thumbnail section.
3. Under the section Map engine, choose your preferred OpenLayers version to manage the application’s map.
4. Set a tick at “*persistent map state*”, to make certain map parameters and configurations persistent. Further information can be found in [Share](#).
5. Click “*save*” to save and create your application. It is now possible to add elements (e.g. map, navigation bar, legend) and services to your application.

It is possible to edit applications: In order to do so, navigate to the application overview. Here, you can


click on a  Button for every application with corresponding editing rights. Furthermore, applications can be viewed in the Frontend via a  Button. They can also be exported (), deleted () or give public access ().

2.4 3. Add elements to your application

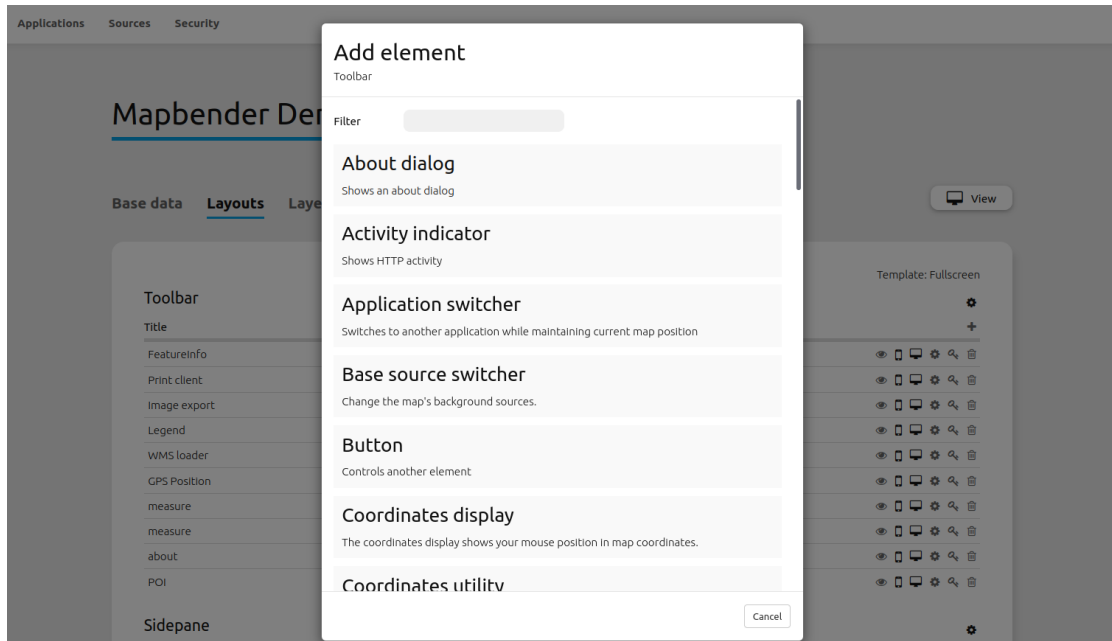
Mapbender applications consist of a Top toolbar, Sidepane, Map area and Footer. A variety of different elements can be added into these areas.

1. Choose Applications  Button .
2. Click on the  Button to get an overview over the elements Mapbender provides in the corresponding area.
3. Choose an element from the list. Notice that you have different areas in your application. Make sure to add the element to a region that makes sense. Not all elements can be added to all regions.

4. Configure the element. Notice: When you select an element, for example **map**, you see that the element has a set of attributes. Each element offers individual attributes for configuration.
5. You can change the position of an element via drag & drop within and between regions.

6. Have a look at your application. To open your application, click on the  Button.

Now you should have an idea about how easy it is to change a Mapbender application.



In the following, you find a complete list of all elements and their functionalities. For a more detailed description, please have a look at the corresponding chapters in the *Table of Contents*.

- *About Dialog* : Shows information about Mapbender in an about dialog
- *Activity Indicator* : Shows HTTP activity
- *View Manager* : Saves map states for later restoration
- *Application Switcher* : Switches to another application while maintaining the current map position
- *Image Export* : Exports the current map view (format options: png or jpeg)
- *Button* : Integrate another element as a button
- *Copyright* : Shows terms of use
- *Data Manager* : erzeugt und speichert Sachinformationen in einer Datenbank
- *Digitizer* : Create and manage spatial data
- *Dimensions handler* : Manage sources with a time dimension
- *PrintClient* : Renders a Print dialog
- *Layertree*: Gives an overview of map layersets and layers
- *Simple Search* : erstellt eine Einfeldsuche
- *GPS Position* : Renders a button to show the GPS position
- *HTML*: Offers free definition of HTML to integrate pictures, texts or links

- *Base source switcher* : Changes the map's background sources
- *FeatureInfo*: Gives information about sources
- *Map*: Creates the map element in which layersets and layers are integrated into
- *Coordinates Utility* : Transforms coordinates to different SRS and navigates to them on the map
- *Coordinates Display*: Shows the map coordinates of your mouse position
- *Legend*: Displays legend of active themes on the map
- *Line/Area Ruler* : Enables to measure a line/area and display its length/area in a dialog
- *Link*: Links to an external URL
- *Scale Display* : Displays the current map scale
- *Simple Search* : Enables a configurable search on JSON sources (e.g. Solr)
- *Scale Bar*: Displays a small line indicator representing the current map scale
- *Scale Selector*: Displays and changes a map scale
- *POI* : Creates a POI for sharing
- *Navigation Toolbar* : Provides a floating control to pan and zoom in the map
- *SRS Selector* : Changes the map's spatial reference system
- *Sketches* : Enables a drawing tool with different shapes
- *Search Router* : Enables a configurable search via SQL
- *Share URL* : Shares the current map view via URL
- *WMS Loader* : Loads a WMS via a getCapabilities-Request
- *About Dialog* : Shows information about Mapbender in an about dialog
- *Overview* : Provides an overview map





2.4.1 Try it yourself

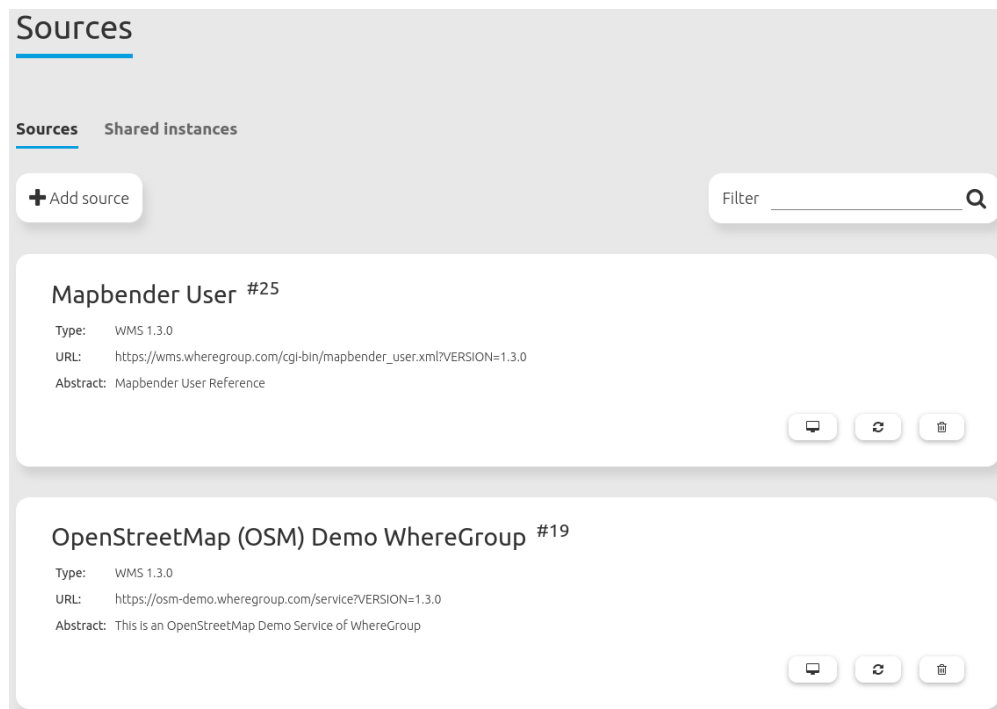
- add a Map to the Map area of your application
- add a Layertree to the Map area of your application
- add a button to the Top toolbar and combine it with the Layertree of your application
- add the Navigation Toolbar to the Map area
- add a Copyright and change the copyright text
- add an SRS Selector to the footer

2.5 4. Configure Sources

Mapbender can handle sources of the type OGC WMS or OGC WMTS / TMS. Via a click on Sources, one can navigate to an overview of all uploaded sources. There is a second list called “*Shared instances*” which only provides sources of the type shared. Further information about bound and shared instances can be found here: [Layerset](#).

The sources pages provides a user with the following functions:

-  add data source
-  show data source
-  update data source
-  delete data source
- filter via text to search for sources

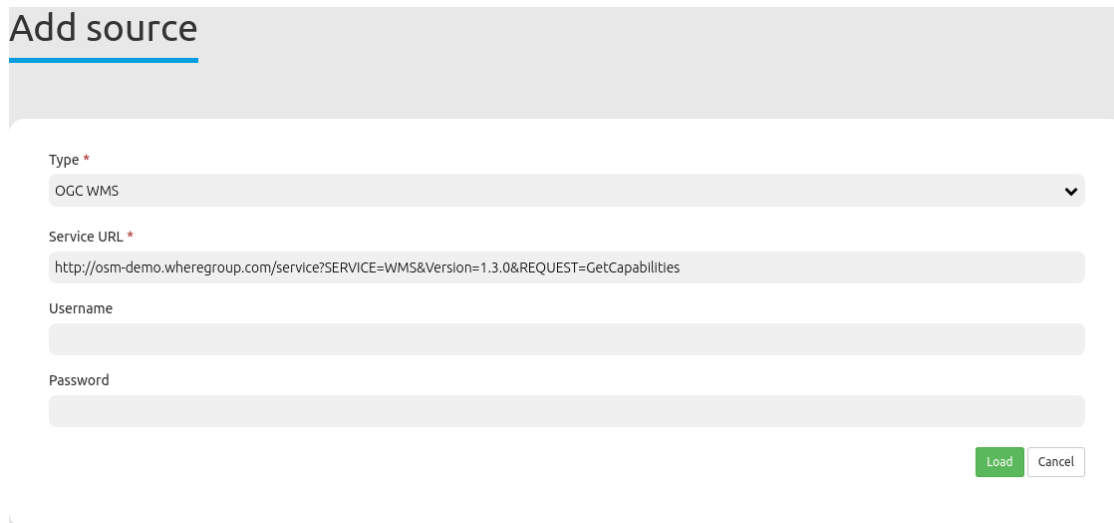


2.5.1 Load sources

Mapbender allows the integration of OGC Web Map Services (WMS) and Web Map Tile Services (WMTS). The versions 1.0.0 and 1.3.0. are supported. A source provides a XML, when the getCapabilities document is requested. This information is read by Mapbender. The client receives all necessary information about a source via this XML.

Tip: You should check your capabilities document in your browser before uploading the service.




1. To upload a source, click on Add source.
2. Define the “Type” of the source: OGC WMS or OGC WMTS / TMS.
3. Provide the link to the getCapabilities URL in the field *Service URL*.
4. Define username and password (in case your source requires it).
5. Click on Load to upload the service into the repository.
6. After a successful upload, Mapbender will provide an overview of the WMS information.

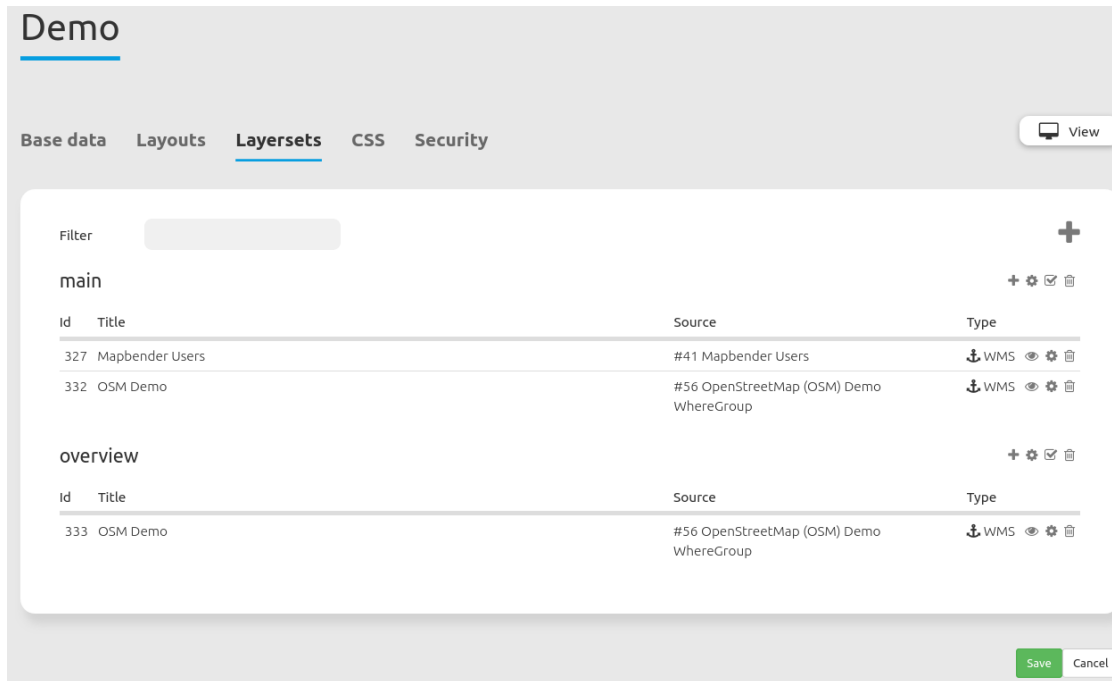


The screenshot shows the 'Add source' form. It has a title 'Add source' with a blue underline. Below the title are four input fields: 'Type' (a dropdown menu with 'OGC WMS' selected), 'Service URL' (a text field containing 'http://osm-demo.wherogroup.com/service?SERVICE=WMS&Version=1.3.0&REQUEST=GetCapabilities'), 'Username' (an empty text field), and 'Password' (an empty text field). At the bottom right of the form are two buttons: a green 'Load' button and a white 'Cancel' button with a grey border.

2.5.2 Add sources to an application



After uploading a service, it can be integrated into one or several application(s).

1. Navigate to your application overview page. Click on the  Button of the desired application and navigate to the tab *Layersets*.
2. In the *Layersets* section you can integrate uploaded sources into your application. Click on  Button next to the filter function to create a layerset. All layers have to be assigned to at least one layerset. Provide a name for it (e.g. “main” for the main map and “overview” for the overview map).
3. Now you can add layers to the layerset. Click on the  Button next to the desired layerset.
4. The order of the layers can be changed via drag & drop.



2.5.3 Source configuration

Sources can be individually configured. This can be useful if you, for instance, don't want to display all layers, change the order or titles of the layers, prevent a layer's feature info output or adjust the scale in which the layers are visible.

1. Click on Application  Layersets  Edit instance to configure an instance.
2. You can now change the instance configuration.
3. The order of the layers can also be changed via drag & drop.

Origin URL: `https://wms.wherogroup.com/cgi-bin/mapbender_user.xml`
Save Back

Title *

Opacity

Format *

Infoformat

Exceptionformat

Tile buffer

BBOX factor

Dimensions
+

Vendor specific
+

☒ BaseSource
☐ Proxy
☒ Transparency
☐ Tiled

Layer ordering *

Title	Minsc	Maxsc	active	select allow on	info allow on	toggle allow on
Mapbender Users			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>
User			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
Name			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Save Back

Source configuration:

- Title: Name of the application
- Opacity: Opacity in percentage
- Format: Format of the getMap-Requests
- Infoformat: Format of the getFeatureInfo-Requests (text/html für die Ausgabe als HTML wird empfohlen)
- Exceptionformat: Format for error messages
- Tile buffer: This parameter is valid for tiles services and specifies if additional tiles should be requested. If the user pans the map, these tiles are already downloaded and visible. The higher the value the more tiles are requested (default: 0).
- BBOX Factor: This parameter is valid for non-tiled WMS services. You can specify the size of the returned map-image. A value greater than 1 will request a bigger map-image (default: 1.25).
- BaseSource: Should the service be handled as BaseSource (BaseSources can be shown/hidden in the layertree)
- Proxy: If active, the service will be requested by Mapbender and not directly

- Transparency: default is active, the source is without a transparent background if it is deactivated (getMap-Request with Transparent=FALSE)
- Tiled: you can request a WMS in tiles, default is not tiled (may be a good choice if your map is very big and the WMS service does not support the width/height)
- Layer ordering: Handles the order of the layers in the service. Can be set toStandard (reversed) and QGIS (same order).

Dimensions:

This function is relevant for sources with a time dimension. Further information can be found under [Dimensions handler](#).

Vendor Specific Parameter:

You can define Vendor Specific Parameters in a layerset instance to add them to a WMS request. This principle follows Multi-Dimensions in the WMS specification.

You can use Vendor Specific Parameters in Mapbender for example to add the user- and group information of the logged-in user to a WMS request. You can also add hard coded values.

The following example shows the definition of the parameter "group", which transfers the group-value of the logged-in user.

- Vstype: Mapbender specific variables. Group (groups), User (users), Simple
- Name: Parameter name of the WMS request
- Default: Default value
- Hidden: If this value is set, requests are send via a server so that the parameters are not directly visible

Currently, the element can be used to transfer user- and group information, e.g. for a user the \$id\$ and for groups the value \$group\$.

Layer configuration:

- title: layer title
- min./max. scale: scale scope
- active on/off: activates/deactivates a layer completely
- select allow: layer is active when the application starts
- select on: selectable in geodata explorer
- info allow: layer info is active when the application starts

- info on: layer provides feature info requests, info default activates the feature info functionality
- toggle allowed: allows opening of folder at application start
- toggle on: open folder on start of the application
- layer ordering: allows to order layer according to the Standard or QGIS configuration.
- more information (...): opens a dialog with detailed layer information:
- ID: ID of the layer
- Name: layer name of the service information (for getMap-Requests)
- Style: if a WMS provides more than one style you can choose a different style than the default style.

2.5.4 Try it yourself

1. Load a source into Mapbender.
2. Add a source to your application.
3. Change the configuration of your source.

Here is an example source:

- WhereGroup OSM WMS: <https://osm-demo.wherogroup.com/service?SERVICE=WMS&VERSION=1.3.0&REQUEST=getCapabilities>

2.6 5. User and group management

Access to Mapbender requires authentication. Only public applications can be used by everyone.

A user can get permissions to access one or a set of applications and services.

2.6.1 Create a user

1. To create a user, go to Security ▢ Users ▢ Add new user.
2. Choose a name for your user.
3. Provide an email address for the user.
4. Choose a password for your user and repeat it in the Confirm password field.
5. Save your new user. It is still possible to alter user information later on.

Applications Sources Security

New User

Base data Profile Groups Security

User name *

E-Mail *

Choose password *

Confirm password *

☒ activated

You can provide more information about the user in the tab **Profile**. In the **Groups** and **Security** tabs it is possible to assign the user additional parameters, e.g. the membership to a group.

Applications Sources Security

Edit User

Base data Profile **Groups** Security

Filter

Name 1 groups of 3 selected

☒ A-Team

☐ B-Team

☐ C-Team

2.6.2 Create a group

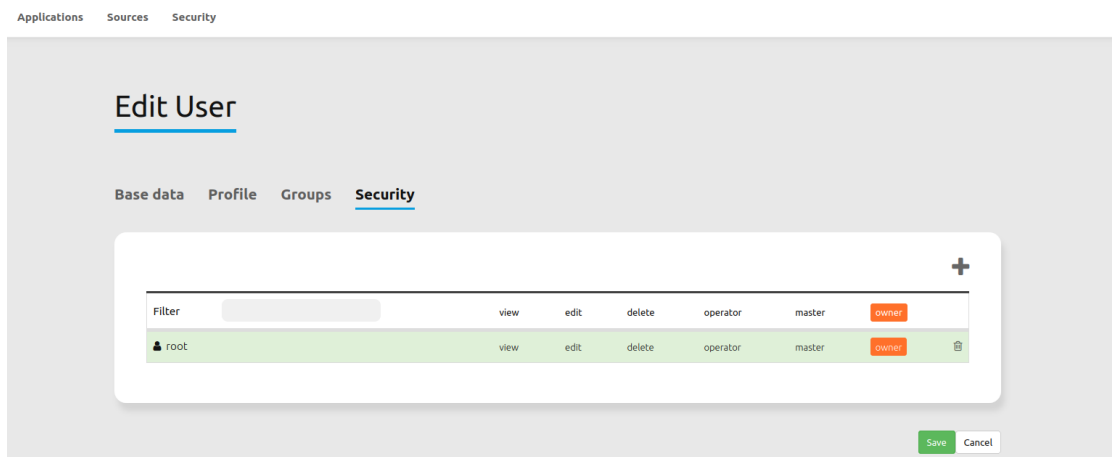
1. Create a group by Security ▸ Groups ▸ Add new Group.
2. Define a name and a description for your group.
3. In the tab Users, assign users to your group.
4. Save your new group.

2.7 6. Rights management


Mapbender provides different rights. They refer to the Symfony ACL System.

- view: Whether someone is allowed to view the object.
- edit: Whether someone is allowed to make changes to the object.
- delete: Whether someone is allowed to delete the object.
- operator: Whether someone is allowed to perform all of the above actions.
- master: Whether someone is allowed to perform all of the above actions and in addition is allowed to grant any of the above permissions to others.
- owner: Whether someone owns the object. An owner can perform any of the above actions and grant master and owner permissions.

Assign roles to a user by Security ▸ Users ▸ Edit your User ▸ Security.



2.7.1 Assign an Application to a User/Group

1. Edit your application via Application  Edit-Button.
2. Choose Security

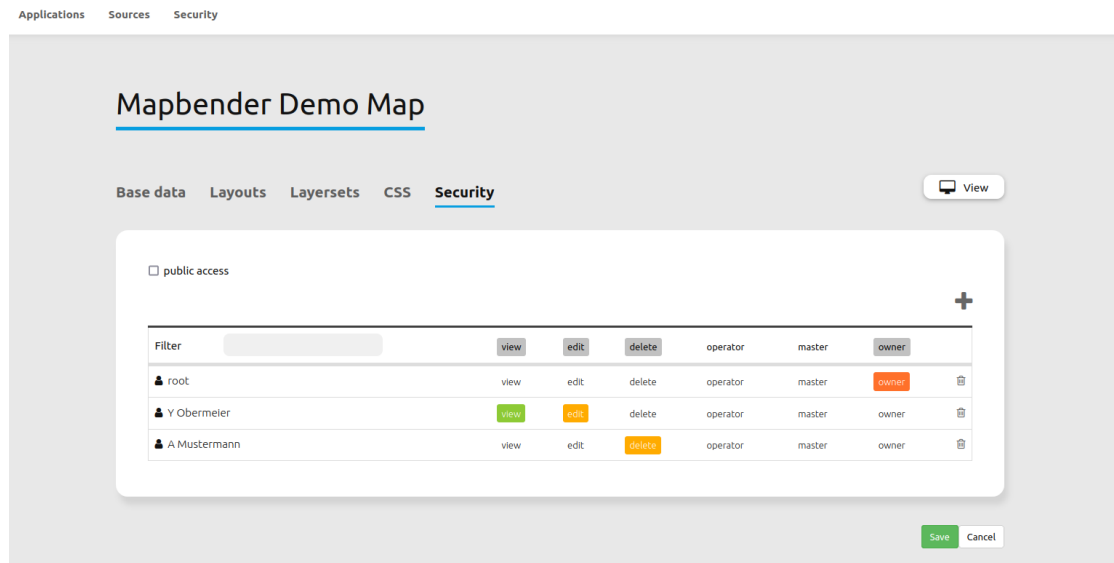
3. Make your application accessible to the public by Security ☐ public access. Alternatively, one can

use the



Button. If this option is activated, also anonymous users will gain access to the application.

4. Alternatively to public access you can set permissions for specific users/groups.



Test your configuration. Logout from Mapbender by Logout. Login again as the new user.

2.7.2 Assign elements to a User/Group

Per default, all elements are accessible to users/groups if they have access to that particular application. This can be modified for each element.

1. Edit your application by pressing the



Edit Button.

2. In your application settings, choose Layouts.

3. Every element has a



Acl button.

4. Choose the



Acl button from the element that should be only available for special users/groups.

5. Assign one or more users or groups to the element. Then, set permissions like view, edit, delete, operator, master, owner
6. Test your configuration.

2.8 7. Start Application at a defined position

You can open an application at a defined location. This can be done by a POI. You also can add texts in the request.

You can pass one or more POIs in the URL. Each POI has the following parameters:

- point: coordinate pair with values separated by comma (mandatory)
- label: Label to display (optional)
- scale: Scale to show POI in (optional, makes only sense with one POI)

If you pass more than one POI, the map will zoom to 150% of the POIs bounding.

To pass a single POI, use the following URL format:

- ?poi[point]=363374,5621936&poi[label]=Hello World&poi[scale]=5000

2.9 What's next?

This is only the first step on the road to using Mapbender. There is a lot more functionality you can try.

Mapbender Website: <https://mapbender.org/>

You find tutorials at: <https://doc.mapbender.org>

Get involved in the project: <https://mapbender.org/en/community/>

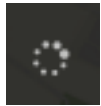
ELEMENTS

3.1 Base elements

3.1.1 Activity Indicator

The activity indicator element provides a simple widget showing background activity (Ajax calls and pending map tile requests). In the default configuration it uses a Font symbol.

The activity indicator can be implemented in footer, sidepane and toolbar. Per default it looks like this:



Configuration

Add element

Footer - Activity indicator

Title	Activity indicator
Tooltip	mb.core.activityindicator.class.title
Activity class	mb-activity
Ajax activity class	mb-activity-ajax
Tile activity class	mb-activity-tile

BackAddCancel

- **Title:** Title of the element.
- **Tooltip:** The text entered as a tooltip will be indicated by hovering over the element with the mouse cursor a longer time.
- **Activity class:** CSS class to indicate activity (Ajax or tile).
- **Ajax activity class:** CSS class to indicate Ajax activity.
- **Tile activiy class:** CSS class to indicate tile loading activity.

YAML-Definition:

This template can be used to insert the element into a YAML application.

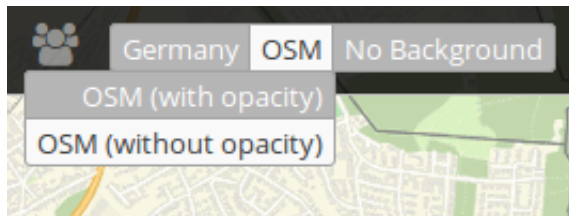
```

activityClass: mb-activity           # CSS class to indicate activity (Ajax or tile)
ajaxActivityClass: mb-activity-ajax # CSS class to indicate Ajax activity
tileActivityClass: mb-activity-tile # CSS class to indicate tile loading activity

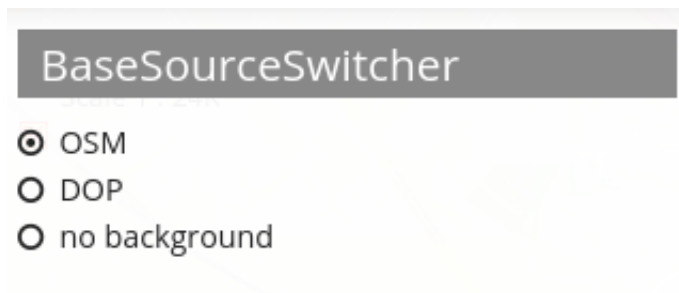
```

3.1.2 Base source switcher

With this element you can switch between different predefined layers (BaseSources), e. g. background maps. The Base source switcher is a button group to change the map's background sources. For every sourceset a button will be displayed in the client. Only one source set can be active at the same time. You have the possibility to define groups. All sourcesets of the same group will be listed in a dropdown list with the group name as title.



The Base source switcher can also be used in the sidepane. The definition of groups is not possible in the sidepane.




Configuration

Preparation: In order to be able to configure the Base source switcher, you have to define Service instances as BaseSource (checkbox "Basesource" on). You can define the layer as a BaseSource in the tab "Layersets" in the backend of the application. You have to edit the layer and set a checkmark at "BaseSource". Please note that on start of an application all sourcesets with an activated root layer are active.

OSM Demo (with Opacity)

WMS Instance 3/21 (Bound Instance)

Source 

[Convert to shared instance](#)

Origin URL: `https://osm-demo.wherogroup.com/service` Save Back

Title *

Opacity

Format *

Infoformat

Exceptionformat

Dimensions

Vendor specific

☒ BaseSource


☐ Proxy

☒ Transparency


☐ Tiled

Layer ordering *

Configuration with selected root layer - sourceset is active on start:

Title	Minsc	Maxsc	active	select		info		toggle	
				allow	on	allow	on	allow	on
 OSM Demo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ OSM Demo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>




Configuration with selected root layer - sourceset is not active on start:

Title	Minsc	Maxsc	active	select		info		toggle	
				allow	on	allow	on	allow	on
 OSM Demo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ OSM Demo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The configuration of the element itself occurs in two steps:

1. Create a Base source switcher element with Title, Tooltip (and, if configured in Map area: Position)
2. Add sourceset(s) with one or more sources and definition of a group (optional)
 - **Title:** Title of the element.
 - **Tooltip:** The text entered as a tooltip will be indicated by hovering over the element with the cursor.
 - **Instancesets:** One or more instancesets can be defined. Select one or more instances and assign a title and group (optional).

Edit element

Title	Base source switcher		
Tooltip	Base source switcher		
Instancesets *	+		
Title *	OSM (With Opacity)		
Group	OSM		
Instances *	<div>5 NW DOP</div> <div>OSM Demo</div> <div>4 NW Overlay für DOP</div> <div>OSM Demo (without Opacity)</div>		
Title *	OSM Demo (Without Opacity)		
Group	OSM		
Instances *	<div>5 NW DOP</div> <div>OSM Demo</div> <div>4 NW Overlay für DOP</div> <div>OSM Demo (without Opacity)</div>		
Title *	DOP		
Group			
Instances *	<div>5 NW DOP</div> <div>OSM Demo</div> <div>4 NW Overlay für DOP</div> <div>OSM Demo (without Opacity)</div>		
Title *	no background		
Group			
Instances *	<div>5 NW DOP</div> <div>OSM Demo</div> <div>4 NW Overlay für DOP</div> <div>OSM Demo (without Opacity)</div>		

Save Cancel

In the configuration example either one, none or several entries per instanceset can be selected. You can create groups, which are then grouped together in the drop-down list. It's possible to move an instanceset in the configuration window via drag and drop.

- **Title:** Title of the BaseSource.
- **Group:** Optional group name.
- **Position:** Position (only when used in Map area). Options: 'left-top', 'right-top', 'left-bottom', 'right-bottom'
- **Instances:** Sources for the BaseSource.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: 'BaseSourceSwitcher'           # title
tooltip: 'BaseSourceSwitcher'        # text to use as tooltip
target: map                           # Id of Map element
anchor: 'right-bottom'                # Position (only when used in Map area). Options:
↪ 'left-top', 'right-top', 'left-bottom', 'right-bottom'
sourcesets:                           # List of sourcesets
  - { title: sourceSetName, group: groupName,
      sources: [sourceId]}            # sourceset: title,
                                     # group: (optional) group name to group of
↪ sourcesets by "group name"          # sources list of sources

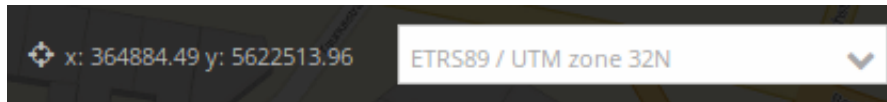
sourcesets:
  - { title: sourceSetName, group: groupName,
      sources: [sourceId]}
```

3.1.3 Coordinates Display

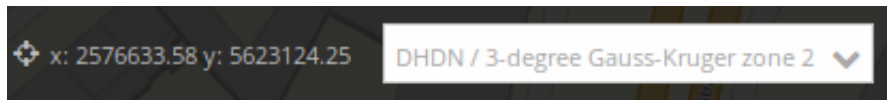
The coordinates display element shows your mouse position in map coordinates. The coordinates are dependent on the selected spatial reference system which may be changed in the [SRS Selector](#).

The coordinates display for different coordinate systems looks like this:

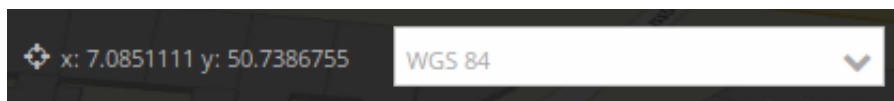
ETRS89 / UTM zone 32N (EPSG:25832):



DHDN / 3-degree Gauss-Kruger zone 2 (EPSG:31466):



WGS 84 (EPSG:4326):



WGS 84 / Pseudo-Mercator (EPSG:3857):



Configuration

 A screenshot of the "Edit element" configuration dialog. The dialog has a title bar "Edit element" and a list of configuration options on the left: "Show label", "Title", "Num digits *", "Empty", "Prefix", "Separator", and "Position *". Each option has a corresponding input field on the right. "Show label" is a checkbox. "Title" is a text field containing "coordinates display". "Num digits *" is a spinner field set to "2". "Empty" is a text field containing "x: - y: -". "Prefix" is a text field containing "x:". "Separator" is a text field containing "y:". "Position *" is a dropdown menu set to "Bottom right". At the bottom right are "Save" and "Cancel" buttons.

Option	Value
Show label	<input type="checkbox"/>
Title	coordinates display
Num digits *	2
Empty	x: - y: -
Prefix	x:
Separator	y:
Position *	Bottom right

- **Show label:** Show a label which incorporates the title and appears next to the coordinates.
- **Title:** Title of the element. It will appear next to the coordinates if "Show label" is activated.
- **Num digits:** Number of decimal digits of the coordinates.
- **Empty:** Text displayed when the mouse is not on the map (default: 'x = - y = -').
- **Prefix:** Prefix in front of the X coordinate (default: '= x').
- **Separator:** Separator in between the X coordinate and Y coordinate (default: 'y= ').
- **Position:** Position (only when used in Map area). Options: 'left-top', 'right-top', 'left-bottom', 'right-bottom'

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
numDigits: 2           # the number of digits each coordinate shall have when being rendered
↪(default: 2)
target: map            # id of Map element to query
label: true            # true/false to label coordinates display (default: false)
empty: 'x= - y= - '    # show this text if the mouse is not on the map
prefix: 'x= '          # show prefix before x coordinate
separator: ' y= '       # show separator before y coordinate
anchor: 'right-bottom'  # Position (only when used in Map area). Options: 'left-top', 'right-top
↪', 'left-bottom', 'right-bottom'
```

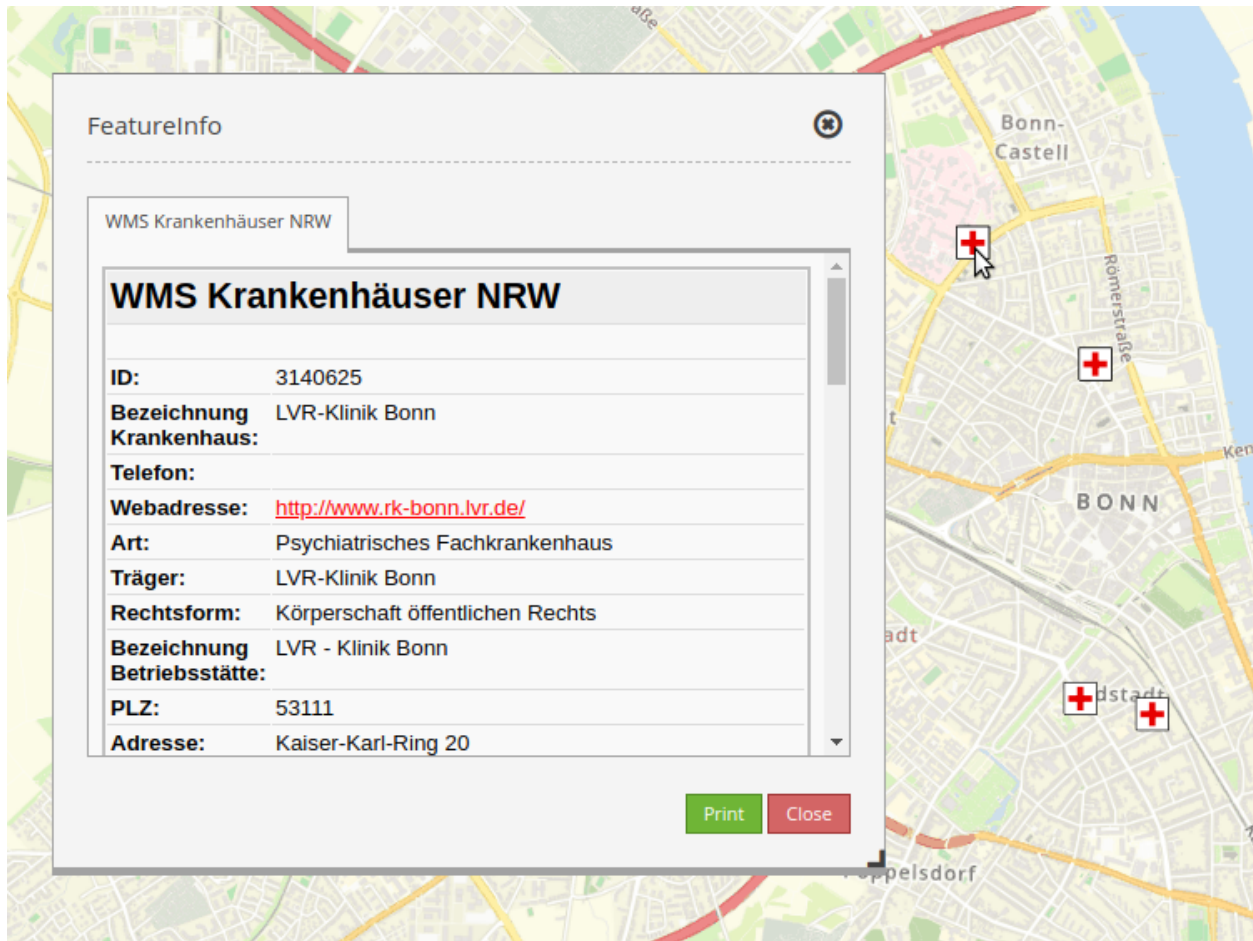
CSS-Styling

The element can be customized with the following CSS-style, for example to increase the width.

```
.mb-element-coordsdisplay {
  width: 500px;
}
```

3.1.4 FeatureInfo

This element provides feature info capabilities to Mapbender. It works with WMS.



The WMS Krankenhäuser NRW (<https://www.wms.nrw.de/wms/krankenhaus?Service=WMS&Version=1.3.0&Request=getCapabilities>) from 'Ministerium für Gesundheit, Emanzipation, Pflege und Alter NRW' serves as example service.

Configuration

Add element

Map area - FeatureInfo

☐ Auto-open ☒ Deactivate on close ☐ Print result ☐ Only valid

Title

FeatureInfo

Display type *

Tabs

▼

Max count

100

⬆ ⬇ ⬇ ⬆

Width *

700

⬆ ⬇ ⬇ ⬆

Height *

500

⬆ ⬇ ⬇ ⬆

Highlighting

☐ Highlighting enabled

Default

On hover

Fill color *

#ffa500

Fill color *

#ff0000

Stroke color

#ff4466

Stroke color

#ff0000

Opacity (%)

40

⬆ ⬇ ⬇ ⬆

Opacity (%)

70

⬆ ⬇ ⬇ ⬆

Stroke width (pixels)

1

⬆ ⬇ ⬇ ⬆

Stroke width (pixels)

3

⬆ ⬇ ⬇ ⬆

Back

Add

Cancel

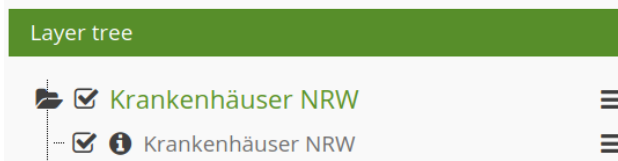
- **Auto-open:** Enable or disable autoopening of the copyright window when starting the application (default: false).

- **Deactivate on close:** true/false to deactivate the functionality after closing the result dialog (default: true).
- **Print Result:** Offer a link to print the result of FeatureInfo (default: false).
- **Only valid:** Display valid WMS (default: false).
- **Title:** Title of the element. It will be indicated next to the button.
- **Display type:** Display of the information, tabs or accordion.
- **Max count:** Maximum number of results that should be displayed in the result dialog.
- **Width/Height:** Width/height of the dialog in px.
- **Highlighting enabled:** Deactivates/activates FeatureInfo Highlighting (default: false).
- **Default fill color** Sets the fill color for selected objects.
- **Default stroke color** Sets the stroke color for selected objects.
- **Opacity (%) of the default color** Sets the opacity of the default color.
- **Stroke width (pixels) of the default color** Sets the stroke width (in pixels) of the default color.
- **Hover fill color** Sets the hover fill color for selected objects.
- **Hover stroke color** Sets the hover stroke color for selected objects.
- **Opacity (%) of the hover color** Sets the opacity of the hover color.
- **Stroke width (pixels) of the hover color** Sets the stroke width (in pixels) of the hover color.

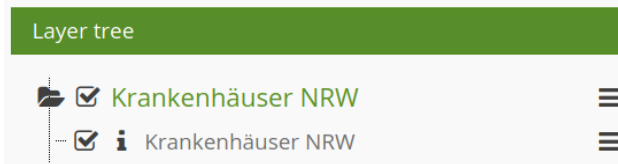
A [Button](#) is also needed for complete frontend integration.

Layer tree settings

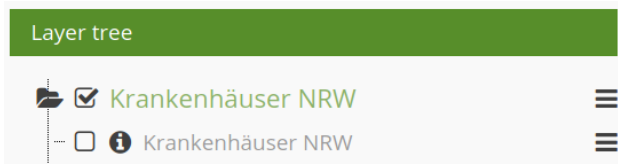
The layer is visible and FeatureInfo request for the layer is activated.



The layer is visible and the FeatureInfo request for the layer is deactivated.



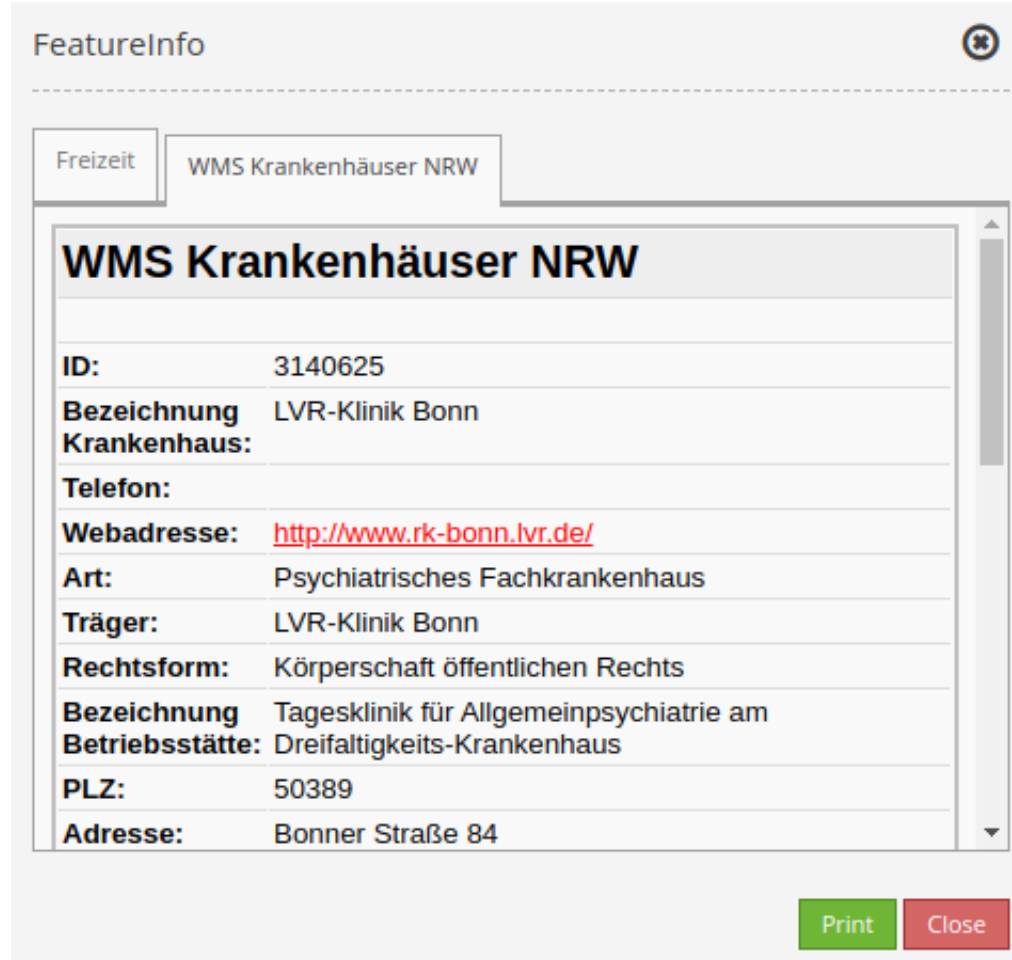
The layer is invisible and there will be no FeatureInfo request (even if FeatureInfo request is activated).



Display as tabs and accordion

With the switch “type”, the responses of multiple services can be displayed either in different tabs or in an accordion.


Example Tabs:



The screenshot shows a web application window titled "FeatureInfo". It contains two tabs: "Freizeit" and "WMS Krankenhäuser NRW". The "WMS Krankenhäuser NRW" tab is active, displaying an accordion view of hospital information. The accordion has a header "WMS Krankenhäuser NRW" and a list of details for a specific hospital. At the bottom right of the window are "Print" and "Close" buttons.

WMS Krankenhäuser NRW	
ID:	3140625
Bezeichnung Krankenhaus:	LVR-Klinik Bonn
Telefon:	
Webadresse:	http://www.rk-bonn.lvr.de/
Art:	Psychiatrisches Fachkrankenhaus
Träger:	LVR-Klinik Bonn
Rechtsform:	Körperschaft öffentlichen Rechts
Bezeichnung Betriebsstätte:	Tagesklinik für Allgemeinpsychiatrie am Dreifaltigkeits-Krankenhaus
PLZ:	50389
Adresse:	Bonner Straße 84

Example Accordion:

FeatureInfo


Freizeit
WMS Krankenhäuser NRW

WMS Krankenhäuser NRW

ID:	3140625
Bezeichnung Krankenhaus:	LVR-Klinik Bonn
Telefon:	
Webadresse:	http://www.rk-bonn.lvr.de/
Art:	Psychiatrisches Fachkrankenhaus
Träger:	LVR-Klinik Bonn
Rechtsform:	Körperschaft öffentlichen Rechts
Bezeichnung Betriebsstätte:	Tagesklinik für Allgemeinpsychiatrie am Dreifaltigkeits-Krankenhaus
PLZ:	50389
Adresse:	Bonner Straße 84

Print
Close

Printing the results

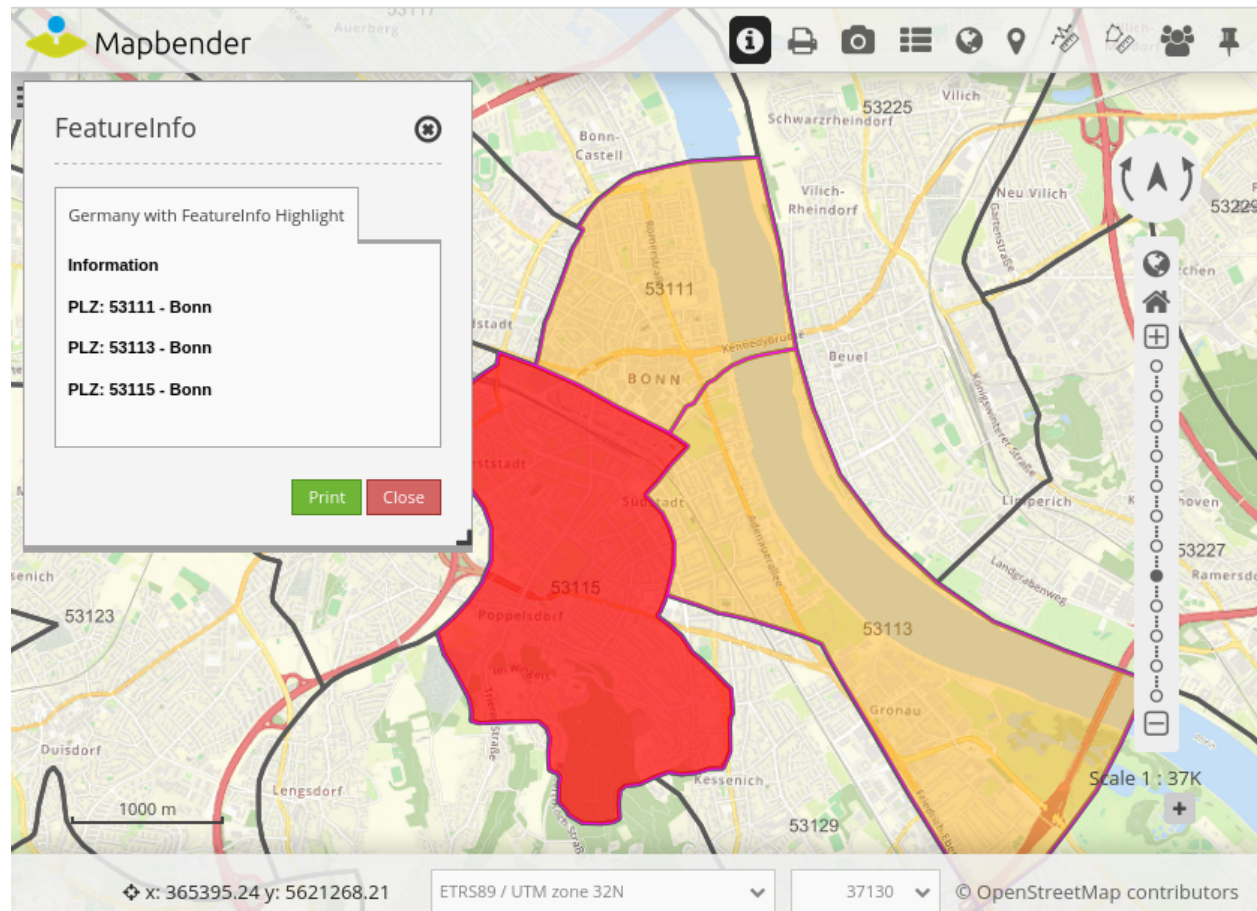
The switch “Print result” allows you to print the output of the FeatureInfo. A “Print” button will appear on the FeatureInfo dialogue. The printing itself is achieved with the printing dialogue of your web browser.

To make sure that all images and background colors are available in your printout, you should check the print settings of your web browser: In Firefox, you can check the option “Print background”. In Chrome-based browsers the option is called “Background graphics”. The used fonts can vary on a printout as PDF and depend on the specific viewer. Furthermore, most web browsers modify the pages a bit before printing to save ink/toner.

FeatureInfo Highlighting

Individual geometries of a WMS can be highlighted with FeatureInfo. This is particularly helpful for the work with comprehensive WMS, because it allows an easier identification of geometries.

A FeatureInfo request with activated highlighting could look like this:



The figure above highlights several geometries in the map (Postcodes ('PLZ'): 53111, 53113 und 53115). The FeatureInfo dialog only displays information belonging to these geometries. The area 53115 is highlighted red due to hovering.

FeatureInfo Highlighting is activated within the Feature Info element. Here, users get the option to choose a default and hover color.

Furthermore, the HTML output of the FeatureInfo request has to be adjusted. In order to do so, the geometry query has to be hidden as a WKT in an HTML div (will not be displayed). In addition, the EPSG code must be transferred and there must be a unique ID in the HTML div (see configuration below). Mapbender evaluates this information and displays the geometries on the map. When you hover over the entries in the info window, the associated geometry is highlighted accordingly. The adaptation looks different depending on which WMS server software you are using. Adjustments can easily be made for MapServer, QGIS Server, GeoServer.

The necessary adjustment is shown here using a configuration for MapServer. In the DATA specification, the geometry is put out as a WKT. In addition, the FeatureInfo template is adapted. If a WMS is now queried via GetFeatureInfo, the corresponding areas are highlighted on the map.

```
DATA "geom from (Select *, ST_AsText(geom) as geom_wkt from plz) as foo USING UNIQUE gid USING SRID 4326
↪"

<div class="geometryElement" id="[gid]" data-geometry="[geom_wkt]" data-srid="EPSG:4326">
<table>
    ...
</table>
</div>
```

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: FeaureInfo           # title of the element
tooltip: Feature Info      # text to use as tooltip
type: dialog               # defines the element type (default: dialog)
target: map                # Id of Map element to query
autoActivate: false       # true/false open when application is started (default: false)
deactivateOnClose: true   # true/false to deactivate the functionality after closing the result
↪ dialog (default: true)
onlyValid: false          # require correct HTML format of response (default: false)
printResult: false        # offer a link to print the result of the featureInfo (default: false)
displayType: tabs         # tabs/accordion (default: tabs)
width: 700                 # width of the dialog in pixels (default: 700)
height: 500                # height of the dialog in pixels (default: 500)
maxCount: 100             # maximum count of hits to be displayed
highlighting: false       # deactivates/activates FeatureInfo Highlighting (default: false)
fillColorDefault: '#ffff00' # fill color to highlight selected objects
strokeColorDefault: '#ff00ff' # stroke color to highlight selected objects
opacityDefault: 25        # opacity for fill color of selected objects
strokeWidthDefault: 3     # line width to highlight selected objects
fillColorHover: '#00ffff'  # fill color to highlight selected objects during hovering
strokeColorHover: '#0000ff' # stroke color to highlight selected objects during hovering
opacityHover: 50          # opacity for fill color of selected objects during hovering
strokeWidthHover: 5       # line width to highlight selected objects during hovering
```

3.1.5 Layertree

Functions

- Display layers
- Display layergroups
- Display thematic layergroups
- Enable and disable layers
- Enable and disable queries for layers
- Change the order of layers
- Zoom to map extent of a layer
- Query the metadata of a layer
- Control the time dimension of a layer

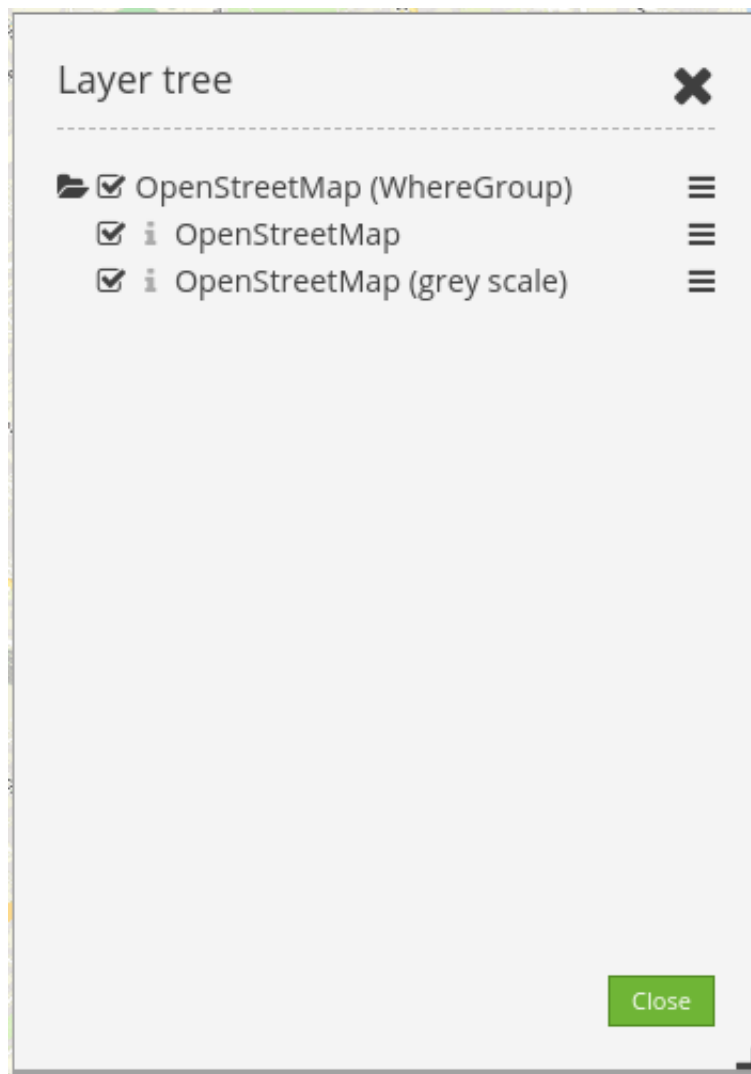


Fig. 1: Layertree as a dialogue popup showing services with layers.

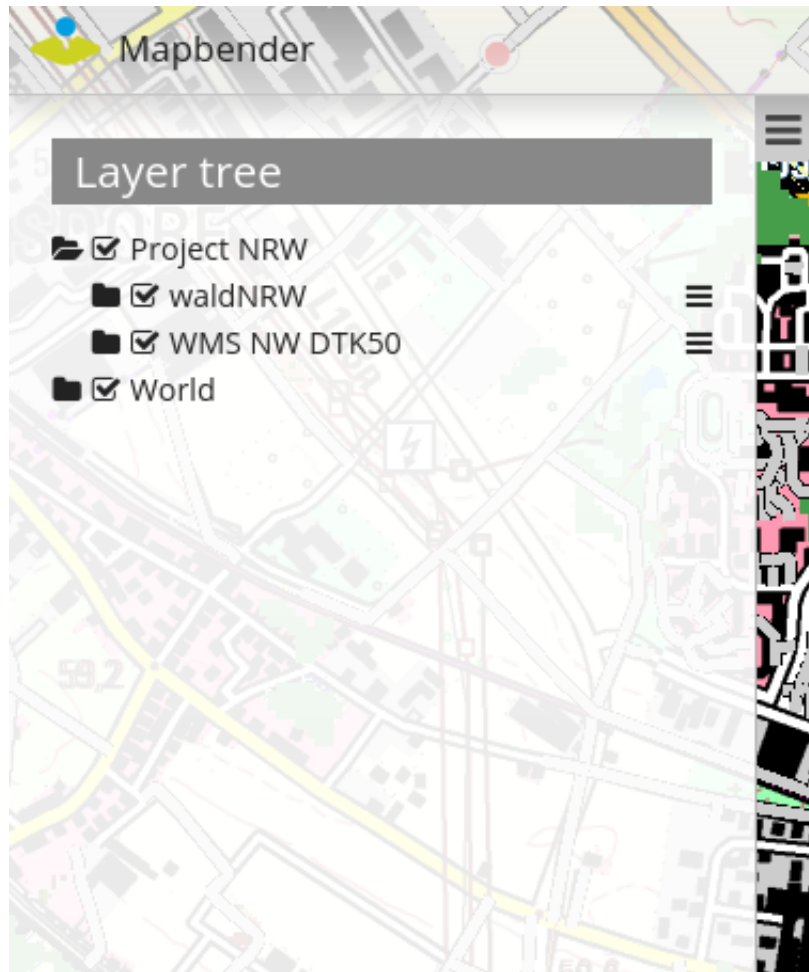


Fig. 2: Thematic Layertree integrated in the sidebar displaying the layerset titles.

Please also configure the elements below that are connected to the Layertree:

- [Layerset](#)
- [Map](#)
- [Sources](#)

Configuration

General Configuration

To use the different Layersets in our Layertree, various adjustments are necessary. These relate to:

1. Configuration of various Layersets
2. Configuration of the map to display the Layersets
3. Configuration of the Layertree itself

Layers are included in the application via the **Layerset** tab in the backend.

The instances are the references to the individual WMS services. With the large + button, new Layersets can be created. New layers can be integrated in the application by adding registered instances into Layersets. In the example, the Layerset “overview” is used for displaying the overview map. For a detailed documentation on how the services can be integrated and registered correctly, please head over to the [Layerset](#) and [Sources](#) documentation..

Demo

Base data
Layouts
Layersets
CSS
Security

View

Filter

+

main

+

⚙

☑

🗑

Id	Title	Source	Type
327	Mapbender Users	#41 Mapbender Users	📍 WMS <div>👁 ⚙ 🗑</div>
332	OSM Demo	#56 OpenStreetMap (OSM) Demo WhereGroup	📍 WMS <div>👁 ⚙ 🗑</div>

overview

+

⚙

☑

🗑

Id	Title	Source	Type
333	OSM Demo	#56 OpenStreetMap (OSM) Demo WhereGroup	📍 WMS <div>👁 ⚙ 🗑</div>

Save Cancel

Fig. 3: Configuration of various Layersets for integration into the Layertree.

In order to let the newly integrated Layersets appear in the application, they must be specified in the [Map element](#). Here, you define which Layersets you want to use in the map by checking the boxes in the list.

For example, the Layerset “overview” is not displayed on the main map. In this step, you define the order in which the Layersets appear in your Layertree and the map. You can move the created Layersets from the list by drag & drop.

Workflow Layertree with basic functions

In the following section, we walk through an exemplary configuration of a Layertree with basic functions in three steps:

1. Configuration of a Layerset
2. Configuration of the map to display the Layerset
3. Configuration of the Layertree itself

Configuration of a Layerset

In the example, we defined one **Layerset** with one instance:

- **Layerset World:**
 - Instance OSM Demo Service <https://osm-demo.wherogroup.com/service?&REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.3.0>

The registered instance of the OSM Demo Service is automatically included in Mapbender’s installation. The WMS only has to be integrated into an existing Layerset. Switch to the tab Layersets. The following example uses the Layerset “World”. In case of difficulties with the integration, [Layerset](#) provides useful information.

Configuration of the map to display the Layerset

In the next step, we configure the [Map element](#) to display the Layersets in the **Map**. To do this, you need to switch to the **Layouts** tab and edit the map element in the map area. It is important that you activate the checkbox next to the “World” Layerset so it appears in the application afterwards.

If you have questions for further configuration, the [Map element](#) documentation can help.

Configuration of the Layertree itself

The last step is the configuration of the **Layertree** itself. The instances specified in the map element can be defined in more detail with the Layertree. If a predefined application has been copied, the Layertree should already work. If the element is newly created, it can already be used with the default settings without necessary further adjustments. For a better understanding of the functions and their usage, we now edit the backend’s Layertree element in the map area.

By enabling **Show base sources**, instances that have been loaded as a base source in the application are displayed in the Layertree.

If you activate **Autoopen**, the Layertree is open by default and does not need to be activated by clicking on a button or the unfolding of a sidepane. The **Title** of the element is displayed in the “Layouts” list and allows you to distinguish between different layertrees.

Via the **Menu** a number of functions can be activated, which are then available in the layertree.

- *Remove layer* (remove layer from the application session)
- *Opacity* (change the opacity of a layer)
- *Zoom to layer* (zoom to full layer extent)
- *Metadata* (show the metadata of the layer)

Edit element

Title	Map			
Layersets *	<input type="checkbox"/> Main <input type="checkbox"/> Overview <input checked="" type="checkbox"/> World			
Tile size	512			
SRS *	EPSG:25832			
Max. extent *	min x *	-500000	min y *	4350000
	max x *	1600000	max y *	6850000
Start extent *	min x *	361000	min y *	5619000
	max x *	371150	max y *	5625000
<input type="checkbox"/> Fixed zoom steps				
Scales (csv) *	5000000,1000000,500000,100000,50000,25000,10000,7500,5000,2500,1000,500			
Other SRS	EPSG:25833,EPSG:31466,EPSG:31467,EPSG:3857,EPSG:4326			

Fig. 4: Configuration of the Layersets in the map element.

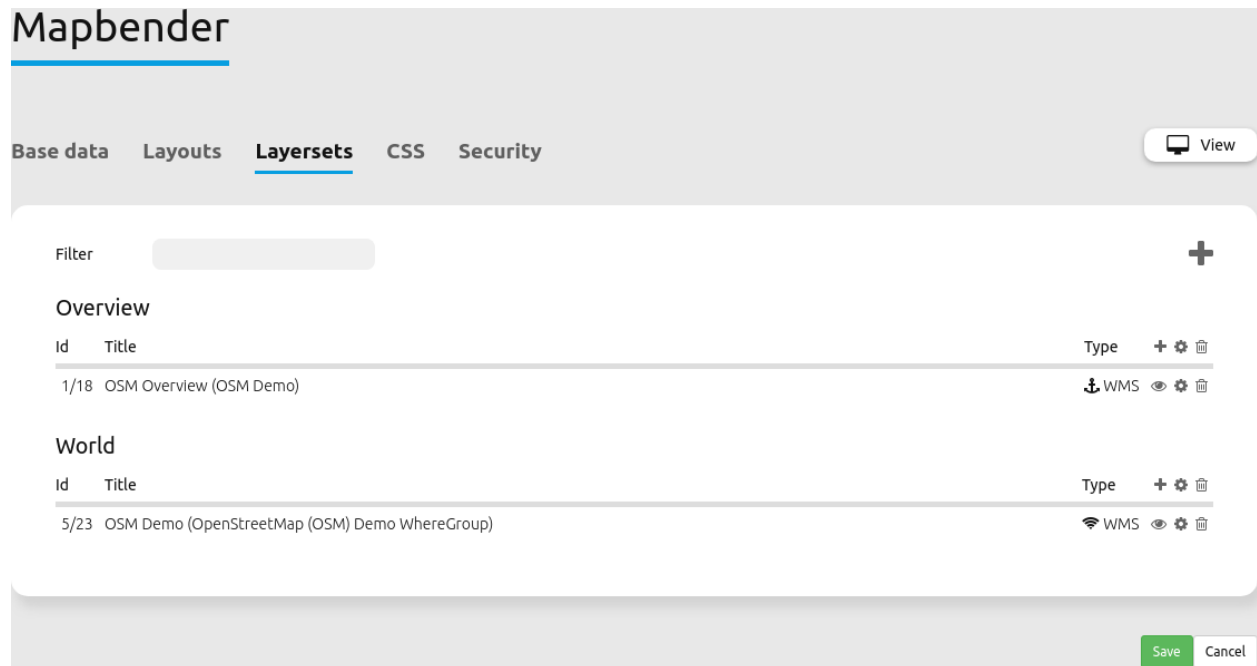


Fig. 5: Configuration of a Layerset for integration into the Layertree.

- *Dimension* (change the dimension, e.g. time or elevation of the Layer - read more about the [Dimensions handler](#))

The individual functions can be activated by clicking on the buttons. All active functions are highlighted. In addition, a symbol of the context menu appears next to each layer in the Layertree. By clicking on the menu, a window pops up and the individual features can be used. The menu can be closed by clicking on the x-button.

The function **Hide visibility by folders** allows to save the configuration of the opacity. When active, the opacity of the individual layer is not displayed and cannot be changed.

The function **Hide info** allows you to disable the "i"-icon. If enabled, the "i"-icon next to a layer name indicates if the FeatureInfo function is active for that layer. The function is inactive if the "i"-icon is grayed out. If you activate the function with a click, the icon will turn dark gray and be framed. If activated, the requested information of the layer appears in a dialogue box.

The checkbox **Thematic layer** activates the thematic division of levels. By clicking on the folder icon of the Layertree to the left of the instance, the embedded layers can be displayed. All layers that have been previously activated in the Layerset now appear in the list.

The checkbox **Allow Reorder at TOC** controls if layers are rearrangeable via drag & drop in the application's layertree (TOC).

The checkbox next to its respective layer name allows to turn a layer on and off. If the checkbox is set, the Layer will appear on the map. However, it will continue to respect the defined rules of the Layerset (such as scale-dependent display).

Edit element

Title	Map			
Layersets *	<input type="checkbox"/> Main <input type="checkbox"/> Overview <input checked="" type="checkbox"/> World			
Tile size	512			
SRS *	EPSG:25832			
Max. extent *	min x *	-500000	min y *	4350000
	max x *	1600000	max y *	6850000
Start extent *	min x *	361000	min y *	5619000
	max x *	371150	max y *	5625000
<input type="checkbox"/> Fixed zoom steps				
Scales (csv) *	5000000,1000000,500000,100000,50000,25000,10000,7500,5000,2500,1000,500			
Other SRS	EPSG:25833,EPSG:31466,EPSG:31467,EPSG:3857,EPSG:4326			

Fig. 6: Configuration of the map element to display the Layerset.

Add element

Sidepane - Layer tree

☒ Show base sources ☐ Autoopen


Title

Menu

- Remove layer
- Opacity
- Zoom to layer
- Metadata
- Dimension

☐ Hide visibility by folders ☐ Hide info

☐ Thematic layer

Themes ☒ 

☒ Allow Reorder at TOC

Fig. 7: Configuration of the simple Layertree in the map area.

Menu

- Remove layer
- Opacity
- Zoom to layer
- Metadata
- Dimension

Fig. 8: Configuration for the context menu.



Fig. 9: Context menu of the layer in the Layertree.

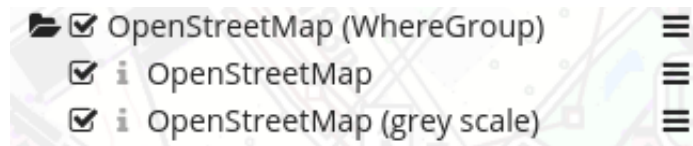


Fig. 10: Layertree Buttons.

Workflow thematic Layertree

In the following section, we walk through an exemplary configuration of a Layertree with advanced features, such as thematic layersets, in three steps:

1. Configuration of multiple Layersets
2. Configuration of the map to display the Layerset
3. Configuration of the thematic Layertree itself

Configuration of multiple Layersets

In the example, we define two layersets with two instances each:

- **Layerset Project NRW:**
 - Instance [DTK50 NRW](#)
 - Instance [Wald NRW](#)
- **Layerset World:**
 - Instance [OSM Demo Service](#) <http://osm-demo.wherogroup.com/service?&REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.3.0>
 - Instance [GEBCO](#)

For the configuration of the layersets, the four services mentioned above were added as instances (For detailed information, see above or in the [Layerset](#) and [Sources](#) documentation).

For this example, the above mentioned steps were performed to add the Layerset “World” [2] with the instance “osm”. Now we add the instance “GEBCO” in this Layerset. To use the thematic grouping, we create a new Layerset named “Project NRW” [3] and load the two instances “DTK50 NRW” and “Forest NRW” into our new Layerset “Project NRW”.

The Layerset should now contain three Layersets. The **Overview** [1] for the overview map, the **Project NRW** Layerset [2] with two regional data sets from Western Germany and the **World**-Layerset [3] with the supranational data.

Configuration of the map to display the Layerset

Mapbender

Base data Layouts **Layersets** CSS Security View

Filter

Overview 1

Id	Title	Type	+	⚙	🗑
1/18	OSM Overview (OSM Demo)	📶 WMS	👁	⚙	🗑

Project NRW 2

Id	Title	Type	+	⚙	🗑
7/25	WMS Wald und Holz NRW (WMS Wald und Holz NRW)	📶 WMS	👁	⚙	🗑
4/22	WMS NW DTK50 (WMS NW DTK50)	📶 WMS	👁	⚙	🗑

World 3

Id	Title	Type	+	⚙	🗑
6/24	WMS for the GEBCO global bathymetric grid (WMS for the GEBCO global bathymetric grid)	📶 WMS	👁	⚙	🗑
5/23	OSM Demo (OpenStreetMap (OSM) Demo WhereGroup)	📶 WMS	👁	⚙	🗑

Save Cancel

Fig. 11: Configuration of Layersets for a thematic Layertree.

Now, we configure the *Map element* to display the Layersets in the map. To do this, we switch to the **Lay-outs** backend tab and edit the feature in the map area. It is now important that you set an active checkbox in the Layerset “World” AND Layerset “Project NRW”, so that they both appear in the application.

If you have questions regarding further configuration of the map, you can view its configuration in the *Map element*.

Configuration of the thematic Layertree itself

The last step is the creation of the Layertree itself. In this example, we add the thematic Layertree to the sidepane.

If the option **Thematic layer** is disabled, the Layertree ignores the configured Layersets and shows the individual instances without thematic structuring in the main level. However, we want to show the layers of our thematic Layersets, so we activate the function **Thematic layer**. Since we inserted both Layersets into the *Map element* of the application, they are now displayed under the **Themes**-area.

Thus, to let the **themes** in the application appear as you want, there are several configuration options:

[1] View theme If this option is set, the Layerset appears as an additional level. If this option is not set, the defined instances are displayed on the main level.

[2] Theme open or closed If this option is set (symbol of the open folder), the theme in the Layertree is automatically opened or closed.

If we keep the default settings in the theme “World” and activate the other options in the theme “Project NRW”, the configuration of the element will look like this:

Now we have the Layersets integrated as thematic groups. By configuring the thematic layer, the Layertree appears as follows:

The Layerset “World” is displayed as a theme, but it is closed and the two buttons are not available. In the Layerset “Project NRW” the theme is shown unfolded upon opening the application. The layers can be activated via a button.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: layertree                                # Title of layertree
target: ~                                       # ID of the Map element to query
type: ~                                        # Type of layertree (element or dialog)
autoOpen: false                               # Opens when application is started (default: false)
showBaseSource: true                          # Shows base layer (default: true)
showHeader: true                              # Shows a headline which counts the number of services
menu: [opacity, zoomto, layer, metadata, remove, layer] # show contextmenu for the layer (like opacity, zoom
↳ to layer, metadata, remove layer), default is menu: []
hideInfo: null
hideSelect: null
allowReorder
themes: { }
```


Edit element

Title

Layersets * ☒ Project NRW
☒ World
☐ Overview

Tile size

SRS *

Max. extent *

min x *	<input type="text" value="-500000"/>	min y *	<input type="text" value="4350000"/>
max x *	<input type="text" value="1600000"/>	max y *	<input type="text" value="6850000"/>

Start extent *

min x *	<input type="text" value="361000"/>	min y *	<input type="text" value="5619000"/>
max x *	<input type="text" value="371150"/>	max y *	<input type="text" value="5625000"/>

☐ Fixed zoom steps

Scales (csv) *

Other SRS

Fig. 12: Configuration of Layersets for a thematic Layertree.

Edit element

☒ Show base sources
 ☐ Autoopen

Title

Menu

Remove layer
 Opacity
 Zoom to layer
 Metadata
 Dimension

☐ Hide visibility by folders
 ☐ Hide info

☒ Thematic layer

Themes

Project NRW	<input checked="" type="checkbox"/>	
World	<input checked="" type="checkbox"/>	

☒ Allow Reorder at TOC

Fig. 13: Configuration of the thematic Layertree in the map area.

☒ Thematic layer
 1 2

Themes

Project NRW	<input checked="" type="checkbox"/>	
World	<input checked="" type="checkbox"/>	

Fig. 14: Backend configuration of the thematic layers.

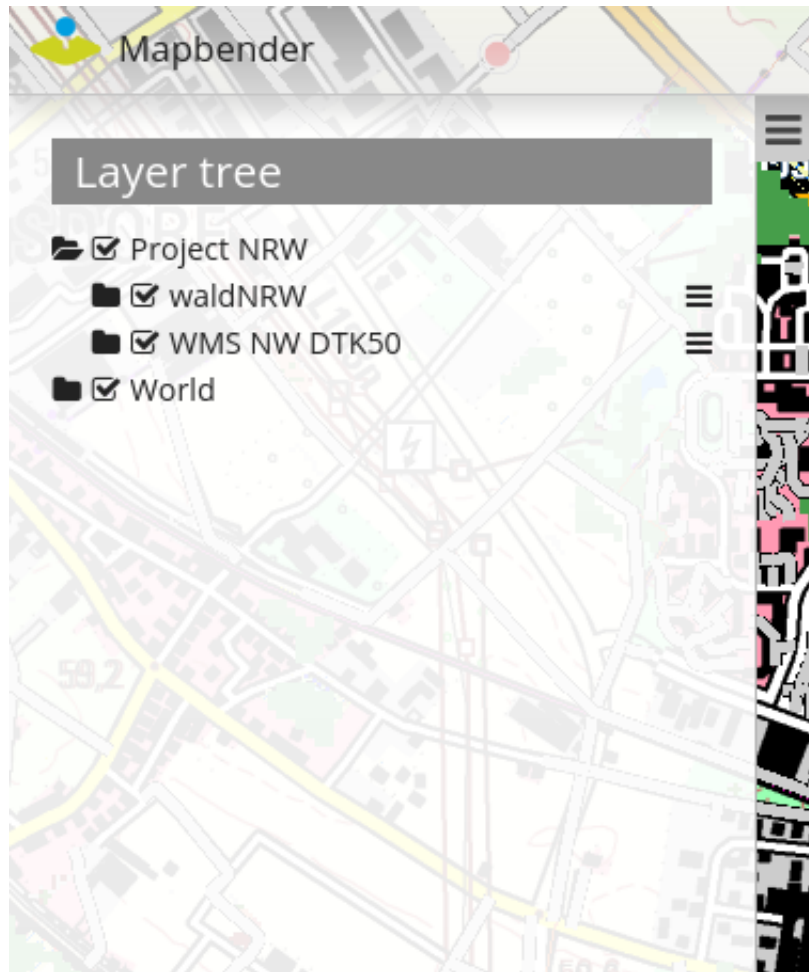
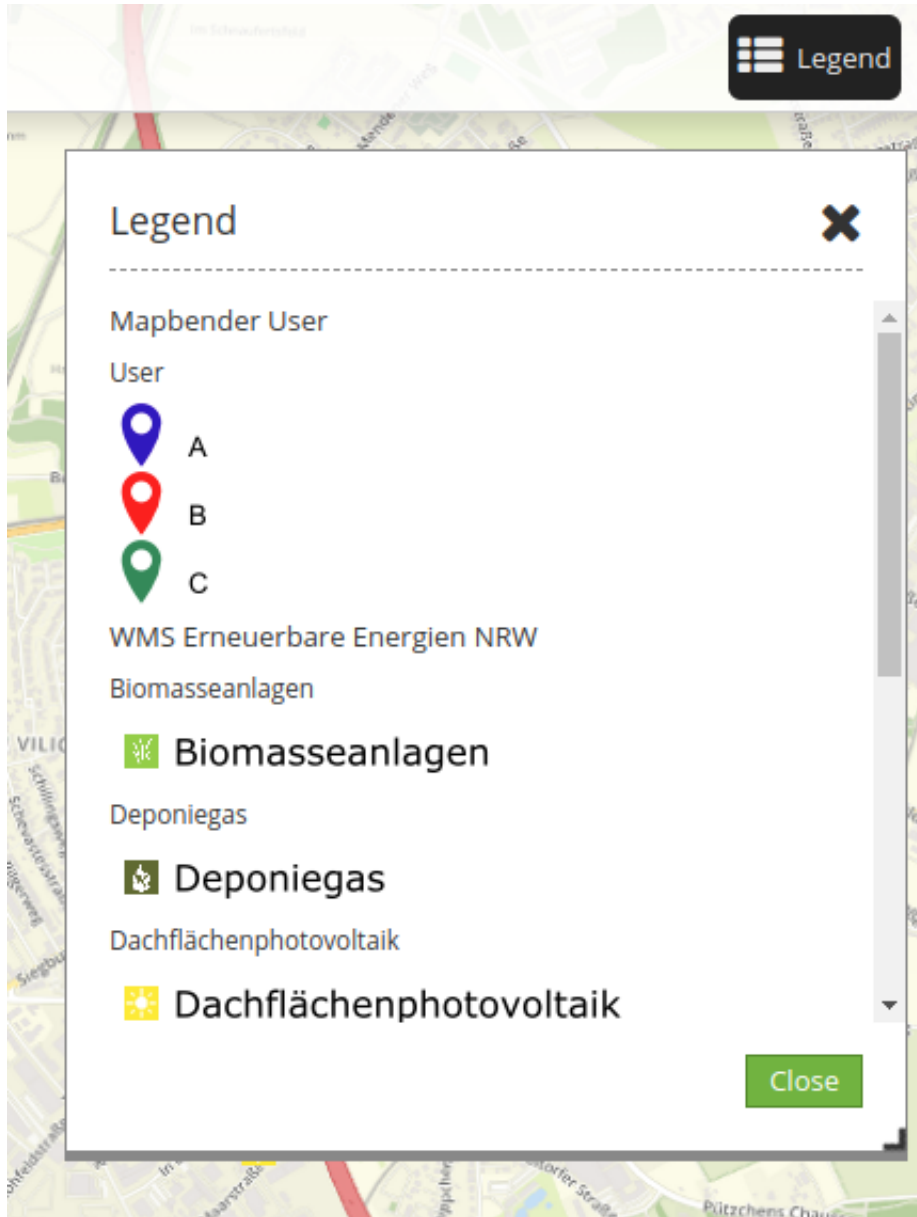


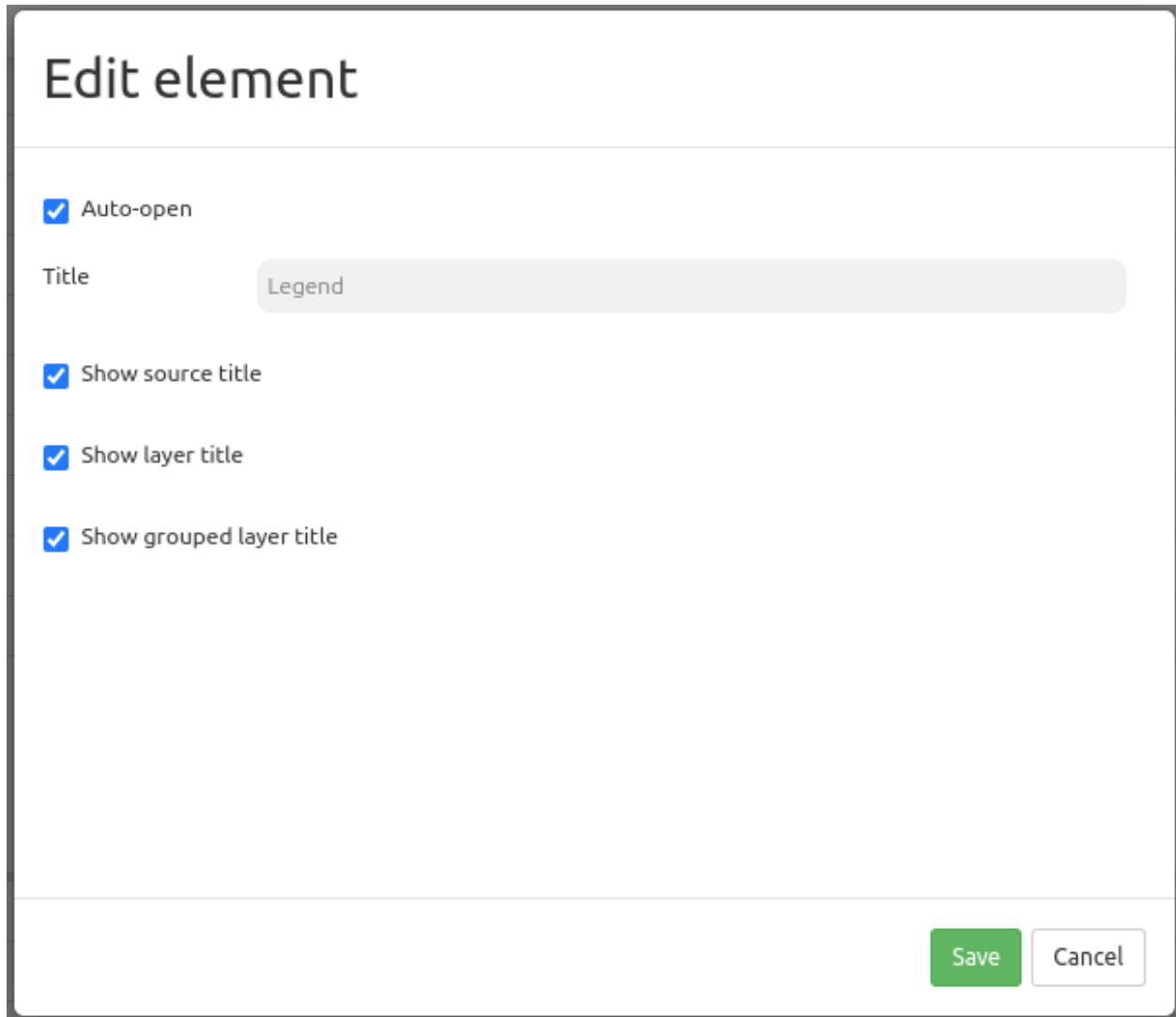
Fig. 15: Configuration of the thematic Layertree in the sidepane.

3.1.6 Legend

The legend object shows a legend of the layers that are displayed in the map. Not every layer provides a legend. In addition to the legend image you can display the service title, layer name and group name (if a group exists).



Configuration



Edit element

☒ Auto-open

Title

☒ Show source title

☒ Show layer title

☒ Show grouped layer title

- **Auto-open:** If activated, the legend opens when the application is started (default: active)
- **Title:** Title of the element. The title will be listed in “Layouts”. Is also shown next to the button, if “Show layer title” is activated.
- **Show source title:** shows WMS/source title (default: true)
- **Show layer title:** shows layer title (default: true)
- **Show grouped layer title:** shows group title for grouped layers (default: true)

The Legend element is integrated via a [Button](#) or in the sidepane.

Configuration Examples

Legend in the Sidepane:

If you want to integrate a legend in the sidepane, click the + -button in the “Layouts”-tab (section “Sidepane”).

Sidepane		⚙
Title	Type	+

Then, choose the element “Legend” in the appearing window. The configurational dialog “Add element – Legend” will open.

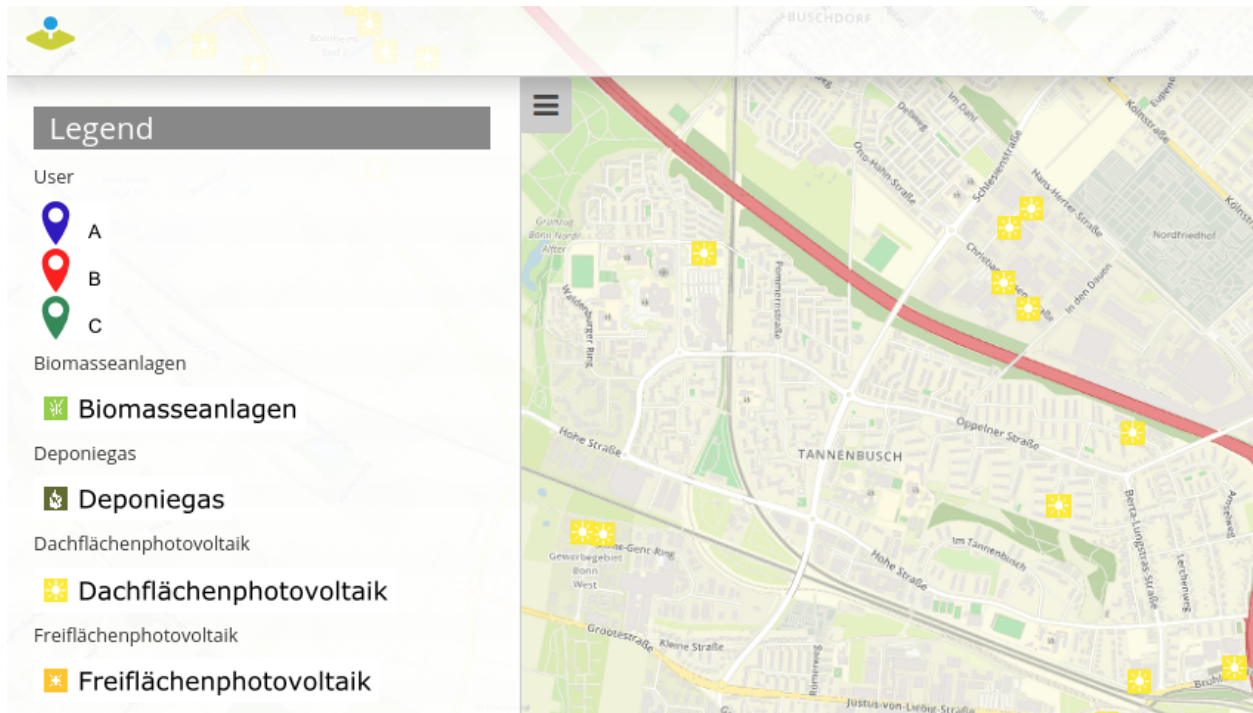
The screenshot shows a dialog box titled "Edit element". Inside, there are several configuration options:

- ☒ Auto-open
- Title: Legend (in a text input field)
- ☐ Show source title
- ☒ Show layer title
- ☒ Show grouped layer title

At the bottom right, there are two buttons: "Save" (green) and "Cancel" (white with a grey border).

Our configured element has the title “Legend”. The legend opens automatically (set checkbox *Auto-open*). Moreover, the layer title and the title of all grouped layers will appear if the checkboxes *Show layer title* and *Show grouped layer title* are set.

Given this configuration, the result looks like this:



In the example the legend was added to the sidepane.

Legend in the toolbar

The legend element can be integrated with a button in the toolbar. First step: Open the application backend and add the legend element into the map area section of the Layout tab.

Map area

Title	Type	
		+

In this example, the following settings are chosen:

Edit element

☒ Auto-open

Title

Legend

☐ Show source title

☒ Show layer title

☒ Show grouped layer title

Save

Cancel

In our example, the checkbox *Auto-open* is dismissed. Therefore, the legend opens only with a click on a button. This *Button* has to be implemented into the toolbar section.

The configuration of a button can look like this:

Add element

Toolbar - Button

☒ Show label

Title

Legend

Tooltip

Show Legend

Icon

Legend (FontAwesome) ▼

Group

Target *

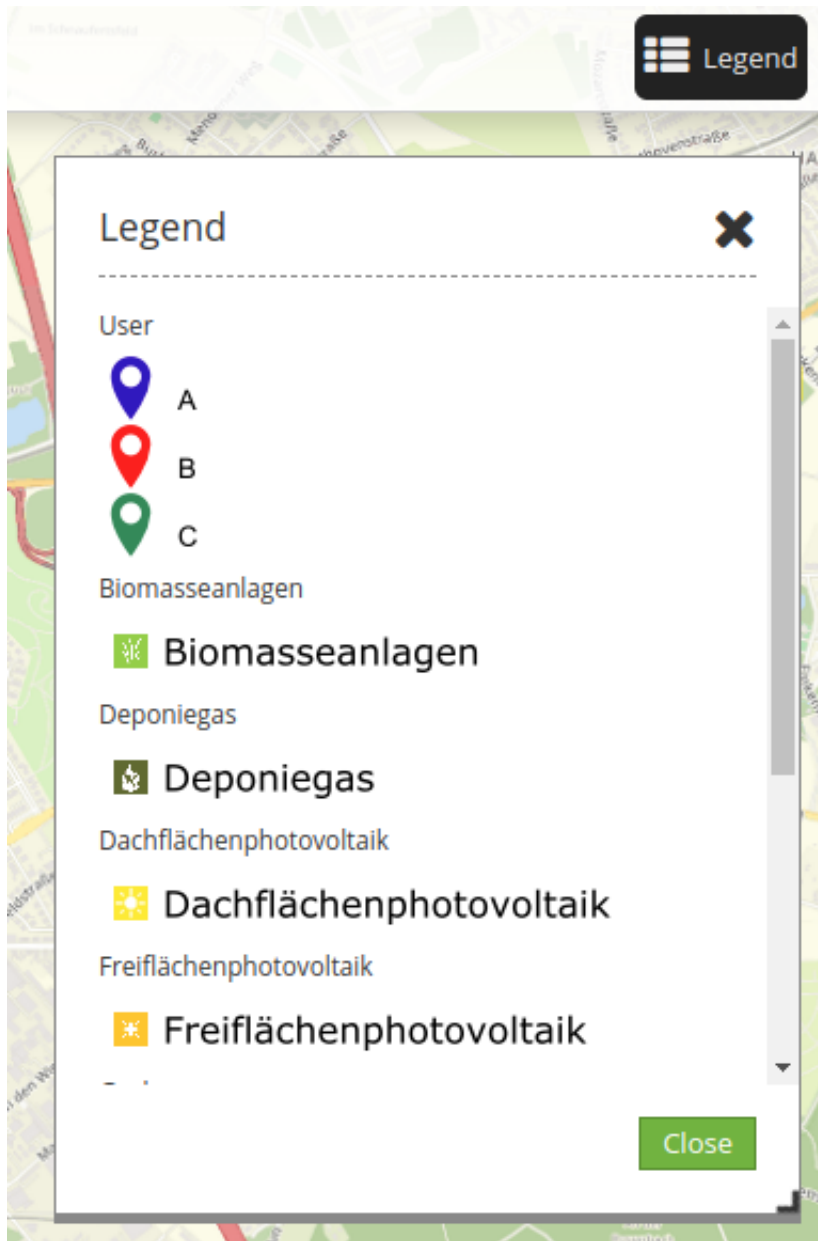
Legend ▼

Back

Add

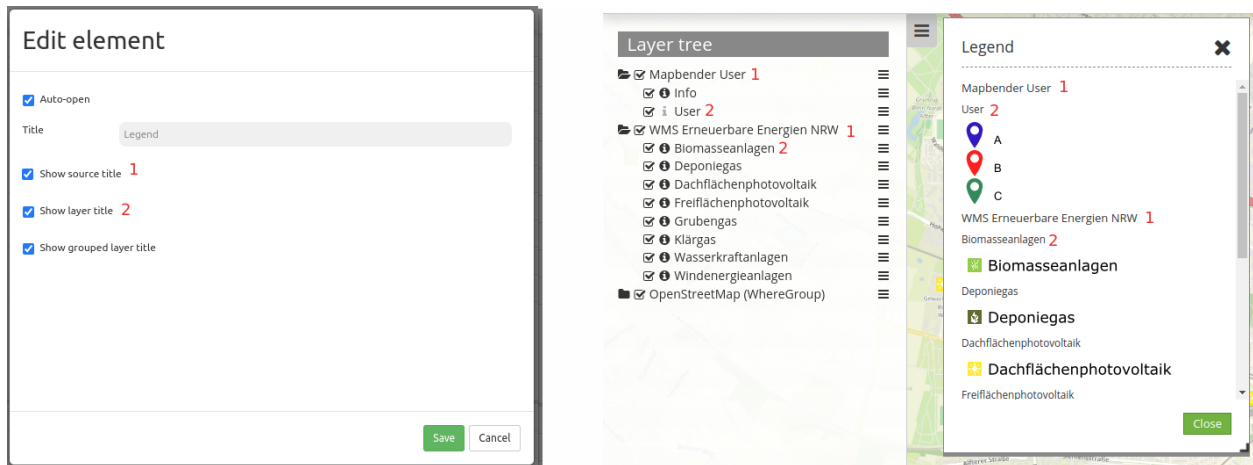
Cancel

Following the above instructions, the result in the application looks like this:



The toolbar shows the button for the legend element. If the button is clicked, the dialog with the generated legend opens.

The activation and deactivation of checkboxes in the configurational settings leads to:



YAML-Definition:

This template can be used to insert the element into a YAML application.

```

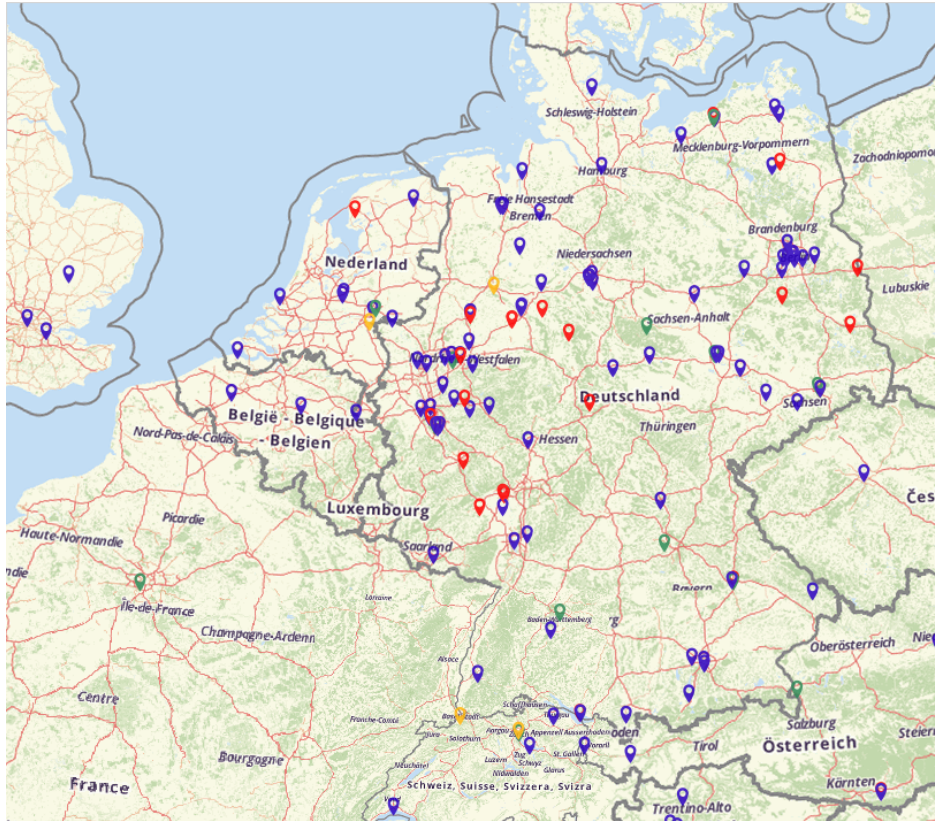
tooltip: 'Legend'           # text to use as tooltip
elementType: dialog        # dialog/blockelement (default: dialog)
autoOpen: true             # true/false open when application is started (default: true)
displayType: list          # accordion/list type of display (default: list)
target: ~                  # Id of Map element to query
hideEmptyLayer: true       # true/false hide when no legend is available (default: true)
showWmsTitle: true         # true/false show WMS title (default: true)
showLayerTitle: true       # true/false show layer title (default: true)
showGroupedLayerTitle: true # true/false show group title for grouped layers (default: true)

```

You can optionally use a button to show this element. See [Button](#) for inherited configuration options.

3.1.7 Map

Map is based on OpenLayers. The element must be integrated into the Map area.



Configuration

Edit element

Title	<input type="text" value="Map"/>			
Layersets *	<input checked="" type="checkbox"/> main <input type="checkbox"/> overview			
Tile size	<input type="text" value="512"/>			
SRS *	<input type="text" value="EPSG:25832"/>			
Max. extent *	min x *	<input type="text" value="-500000"/>	min y *	<input type="text" value="4350000"/> ?
	max x *	<input type="text" value="1600000"/>	max y *	<input type="text" value="6850000"/>
Start extent *	min x *	<input type="text" value="361000"/>	min y *	<input type="text" value="5619000"/> ?
	max x *	<input type="text" value="371150"/>	max y *	<input type="text" value="5625000"/>
Default Resolution [dpi] *	<input type="text" value="96"/>			?
<input type="checkbox"/> Fixed zoom steps				
Scales (csv) *	<input type="text" value="5000000,1000000,500000,100000,50000,25000,10000,7500,5000,2500,1000,500"/>			
Other SRS	<input type="text" value="EPSG:25833,EPSG:31466,EPSG:31467,EPSG:3857,EPSG:4326"/>			

- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Layersets:** Refers to a layerset. Layersets have to be defined first and can then be referred to.
- **Tile size:** Size of the tiles of tiled WMS services.
- **SRS:** Spatial reference system. Two ways of SRS definitions are supported: EPSG:CODE or EPSG:CODE|MY SRS TITLE.
- **Max. Extent:** Maximal map extent, defined by BBOX parameters.
- **Start Extent:** Map extent that is visible at application launch. Defined by BBOX parameters.
- **Default resolution [dpi]:** The default resolution adapts to the screen resolution based on the configured value in dpi. Default: 96 dpi.
- **Fixed zoom steps:** This option activates a zoom behaviour with fixed scales. This is useful to increase

visual quality of services that are cached on very particular resolution steps only. When set true, scale denominator snaps to one of the values given in the *scales* option as defined below (default: false).

- **Scales (csv):** A csv scale list. These scales will be supported in your application if you zoom (e.g. via mouse wheel)
- **Other SRS:** Other spatial reference systems. Two SRS definitions are supported: EPSG:CODE or EPSG:CODE|MY SRS TITLE.

Configuration example

The map element has to be included into the Map area:

Map area	
Title	Type +

The map can entail all instances that are defined in the layerset. The following example distinguishes between a *main* (1) and *overview* (2) layerset.

Demo

Base data
Layouts
Layersets
CSS
Security

View

Filter

main 1.

Id	Title	Type	+	⚙	🗑
2/1	Mapbender Users (Mapbender Users)	WMS	📌	👁	⚙
1/2	OSM Demo (OSM Demo)	WMS	📌	👁	⚙

overview 2.

Id	Title	Type	+	⚙	🗑
1/3	OSM Demo (OSM Demo)	WMS	📌	👁	⚙

In order to display all *Layersets* on the map, they have to be activated. Multiple selections are possible as well. De-selected layersets can function as an *Overview*. In the example, *main* is displayed on the main map and *overview* as an overview map.

The field *SRS* defines the coordinate reference system that is used at application start. In this example, the coordinate reference system EPSG:25832 or ETRS89/UTM Zone 32N was chosen. If the application should support other coordinate systems, simply add those in the *Other SRS* field. In this example, the following codes are used: EPSG:25833 (ETRS89/UTM Zone 33N), EPSG:31466 (DHDN/3-degree Gauss-Kruger Zone 2), EPSG:31467 (DHDN/3-degree Gauss-Kruger Zone 3), EPSG:3857 (WGS 84/Pseudo-Mercator) and EPSG:4326 (WGS 84).

The field *max. Extent* states the maximum zoomable extent of the map application. If there is data outside of the extent, it will not appear in the map. The field *start Extent* refers to the extent of the map that is visible when the application is started in the browser (in this example: the city of Bonn).

The *Default resolution* in dpi defines the resolution of the device being used; the corresponding default value of 96 dpi can be adjusted through this field. If the displayed resolution of the map does not match that of the WMS service, changing this value can help to align the map accordingly.

Note: The scale-dependent resolution currently only works reliably on desktops with regular resolution. Moreover, *Default resolution* is only available from Mapbender 3.3.5 onwards.

Furthermore, the field *scales (csv)* defines the scales that are usable in the application. It is possible to switch between the defined scales with *Scale Selector* or *Navigation Toolbar*. *Fixed zoom steps* were deactivated in the example. That means it is possible to display undefined zoom levels via mouse scrolling.

YAML-Definition:

This template can be used to include the map into a YAML application.

```
layerset: null           # refer to a layerset, define the layerset first and refer to it
srs: "EPSG:4326"        # coordinate reference system. Two ways of srs definitions are supported:
                        # - "EPSG:CODE" or
                        # - "EPSG:CODE|MY SRS TITLE"

extents:
  max: [0, 40, 20, 60]  # maximal map extents
  start: [5, 45, 15, 55] # map extents for the start of the application
scales: "25000000,10000000,5000000,1000000,500000" # a csv scale list
otherSrs:
  - EPSG:31466
  - EPSG:31467
  - EPSG:25832          # other coordinate reference systems. Two srs definitions are supported:
                        # - ["EPSG:CODE", "EPSG:CODE"] or
                        # - ["EPSG:CODE|MY SRS TITLE", "EPSG:CODE|MY SRS TITLE"]
tileSize: 256          # size of tiles
```

Controlling by URL-parameters

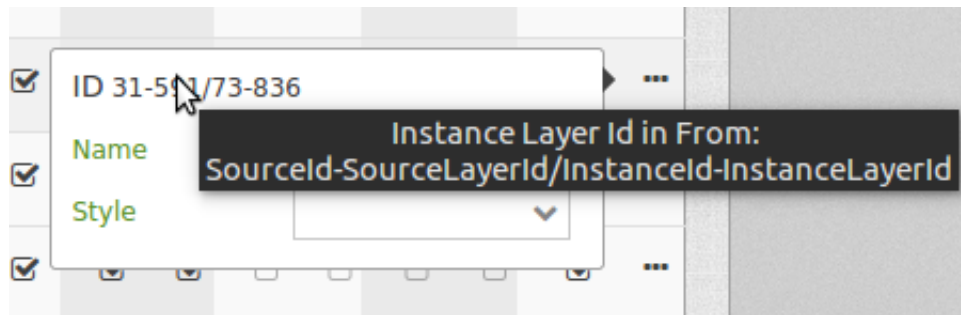
Make Layer visible

If you have a layer with the id <layerid> in a service with the id <serviceid>, you may pass the URL parameter visiblelayers to turn the layer visible:

```
?visiblelayers=<serviceid>/<layerid>
```

You may also pass multiple layers separated by comma.

The layerid and serviceid values are specific to an application. You can get the layerid and serviceid in the specific application, namely in the layerset and there in a layer. Each layer has an icon with three small dots on the right side. Click on the icon and a popup window will appear.



The first value lists the internal SourceID and SourceLayerId (31-591). The second value lists the InstanceID and InstanceLayerId that we want to use now (73-836).

Use these values for the “visibleLayers” parameter in your URL, and separate them by a slash.

For example: <http://localhost/mapbender/application/myapp?visiblelayers=73/836>

If you have two layers that are not visible by default, put the two values of layerid and serviceid into the URL and separate them by a comma.

For example: <http://localhost/mapbender/application/myapp?visiblelayers=73/836,73/840>

Passing POIs

You can pass one or more POIs in the URL. Each POI has the following parameters:

- point: coordinate pair with values separated by comma (mandatory)
- label: Label to display (optional)
- scale: Scale to show POI in (optional, makes only sense with one POI)

If you pass more than one POI, the map will zoom to 150% of the POIs bounding.

To pass a single POI, use the following URL format:

```
?poi[point]=363374,5621936&poi[label]=Label&poi[scale]=5000
```

Passing BBOX

You can pass a BBOX to zoom to by using the following URL query parameter:

```
?bbox=364286,5622263,365979,5622806
```

Passing the scale

You can set the scale from the list of available scales.

```
?scale=1000
```


Passing SRS

You can pass a favorite EPSG code you want to use on start of the application by URL query parameter:

```
?srs=EPSG:4326
```

Passing Center

You can pass a coordinate. The application will open and display the coordinate in the center. In this case, you also have to set the SRS.

```
?center=364286,5622263
```

More Start Parameters

The elements WMS Loader and WMC Loader also provide parameters you can use on start. Have a look at the element descriptions for further information.

3.1.8 Navigation Toolbar

This element simplifies navigation within the map via rotation and zoom. Different scales can be selected from a bar or slider. Users also have the option to zoom to max extent or return to the start configuration. The navigation toolbar itself is draggable.



Configuration

Add element

Map area - Navigation toolbar

☒ Draggable

Title

Components *

Rotation
 zoom to max extent
 Back to start
 Zoom in/out
 Zoom slider

☐ "Back to start" also resets layer settings

Position *

Top left
 ▼

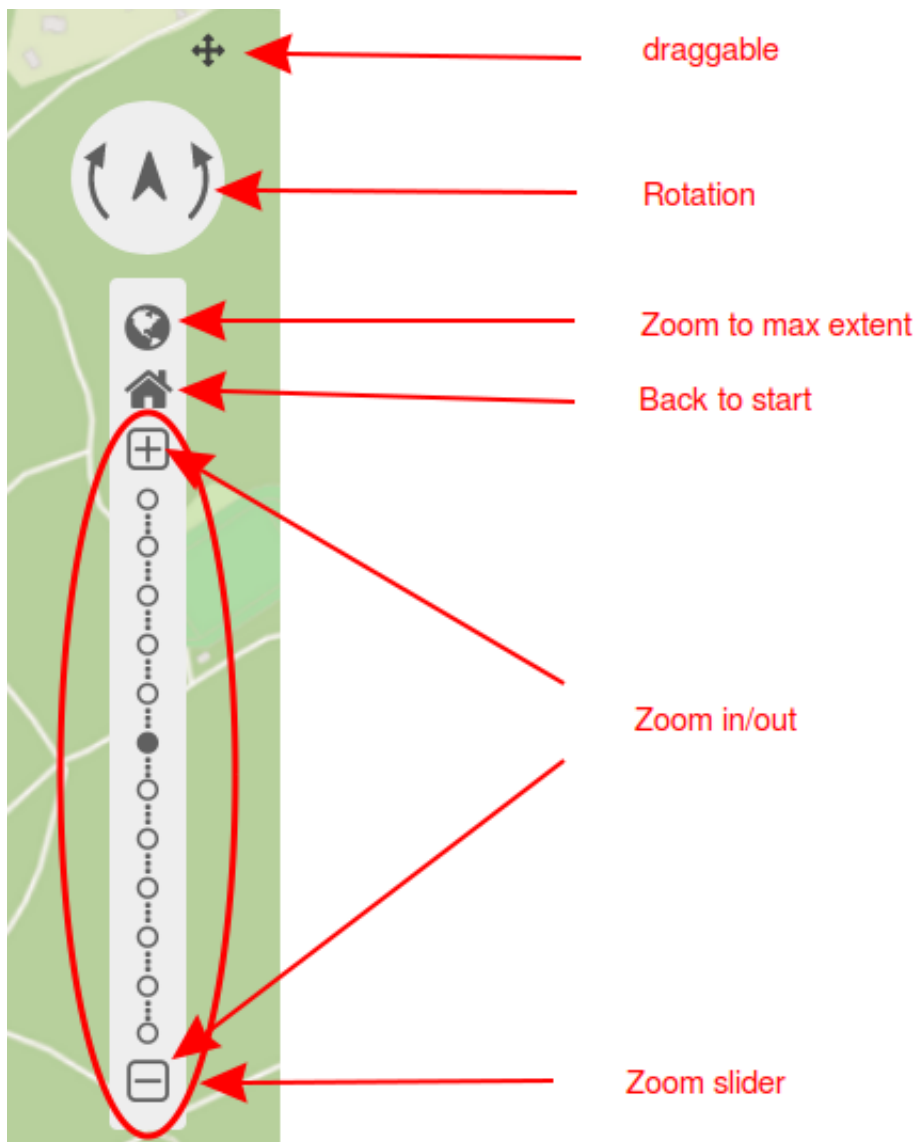
- **Draggable:** Element is draggable or not (default: true)
- **Title:** Title of the element. The title will be listed in "Layouts" and allows to distinguish between different buttons. It will be indicated if "Show label" is activated.
- **Components:** Components of the navigation toolbar (all selected by default). Options: Rotation, zoom to max extent, Back to start, Zoom in/out, Zoom slider
- **Back to start:** Resets layer settings (default: false).
- **Position:** Navigation toolbar alignment, default is 'left-top' (use inline e.g. in sidebar). Options: inline (for using the element in the sidepane), left-top, left-bottom, right-top, right-bottom

Components of the Navigation Toolbar element:

You can integrate various different *components* in the configuration dialog box. These components include:

- Rotation
- Zoom to max extent
- Zoom to start
- Zoom in/out
- Zoom slider

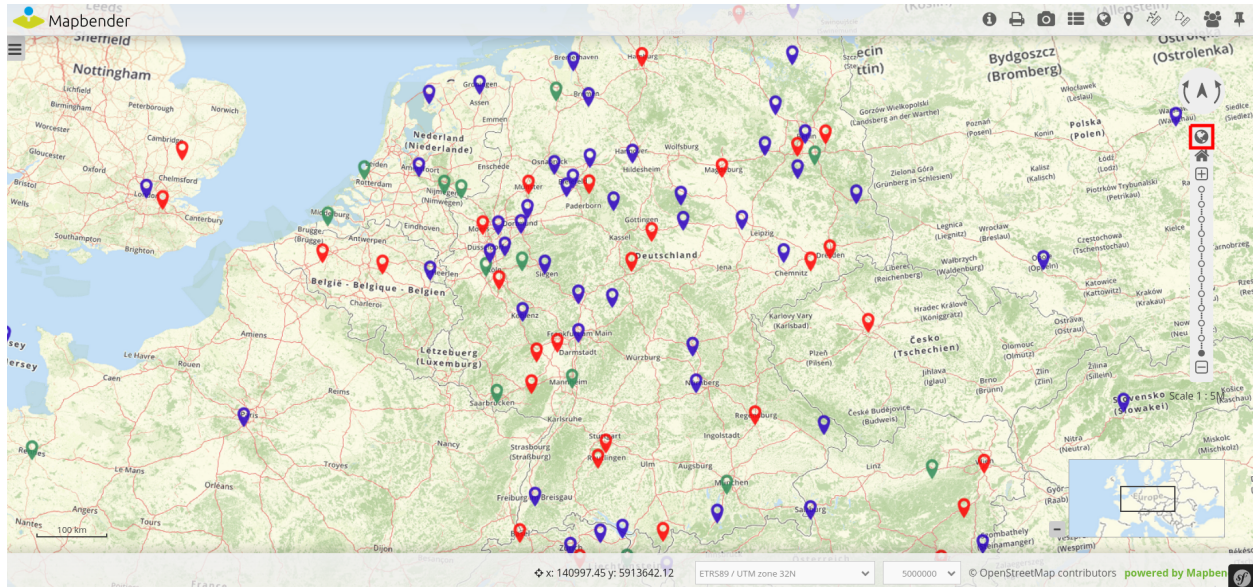
In the application these components look like this:



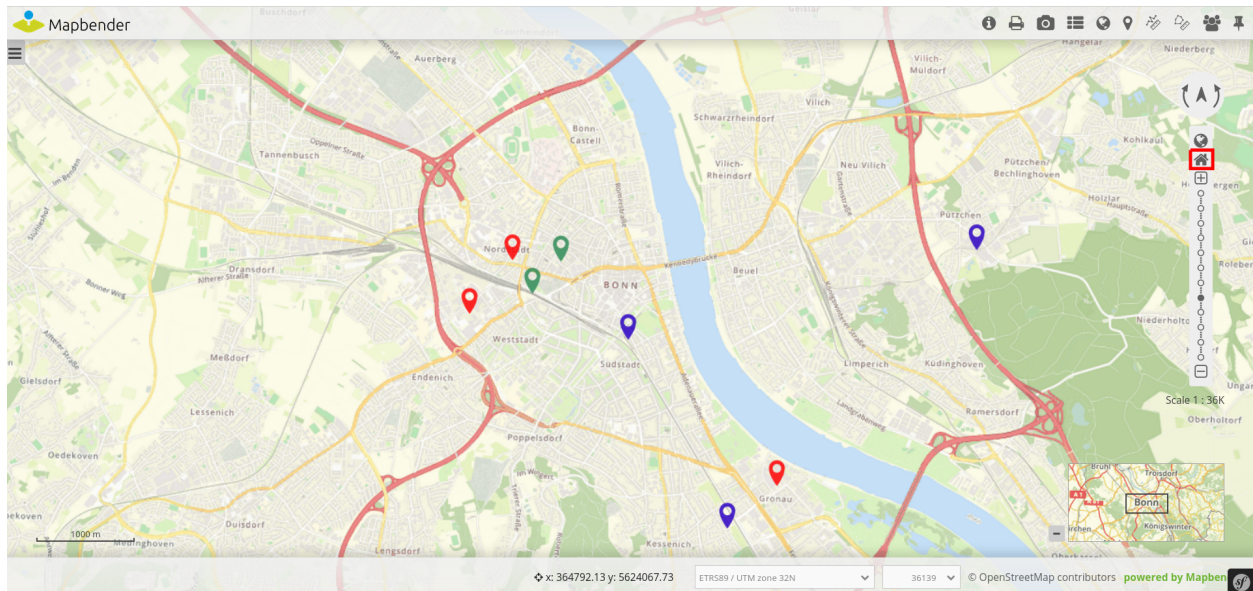
If you set a tick at **Draggable**, then a small cross will appear next to the navigation toolbar that allows the user to move the element around.

Rotation enables changes of the map orientation. A click on the north arrow allows for a restoration of the original rotation.

Users can zoom to their smallest scale by **Zoom to max extent**. This *Max Extent* can be set individually in the *map element*. The element must be integrated into the Map area.



It is possible to restore the initial center, scale, SRS and rotation with **Zoom to start**.



Finally, **Zoom in/out** allows to zoom in (+ -button) and out (- -button) of the map. The element automatically sets to the next higher or lower scale. Zoom steps can be defined in the map element as well. There, users also have the option to deactivate fixed zoom steps. The component **Zoom slider** already includes *Zoom in/out*. Additionally, the user can select different scales from a bar.

YAML-Definition:

This template can be used to insert the element into a YAML application.

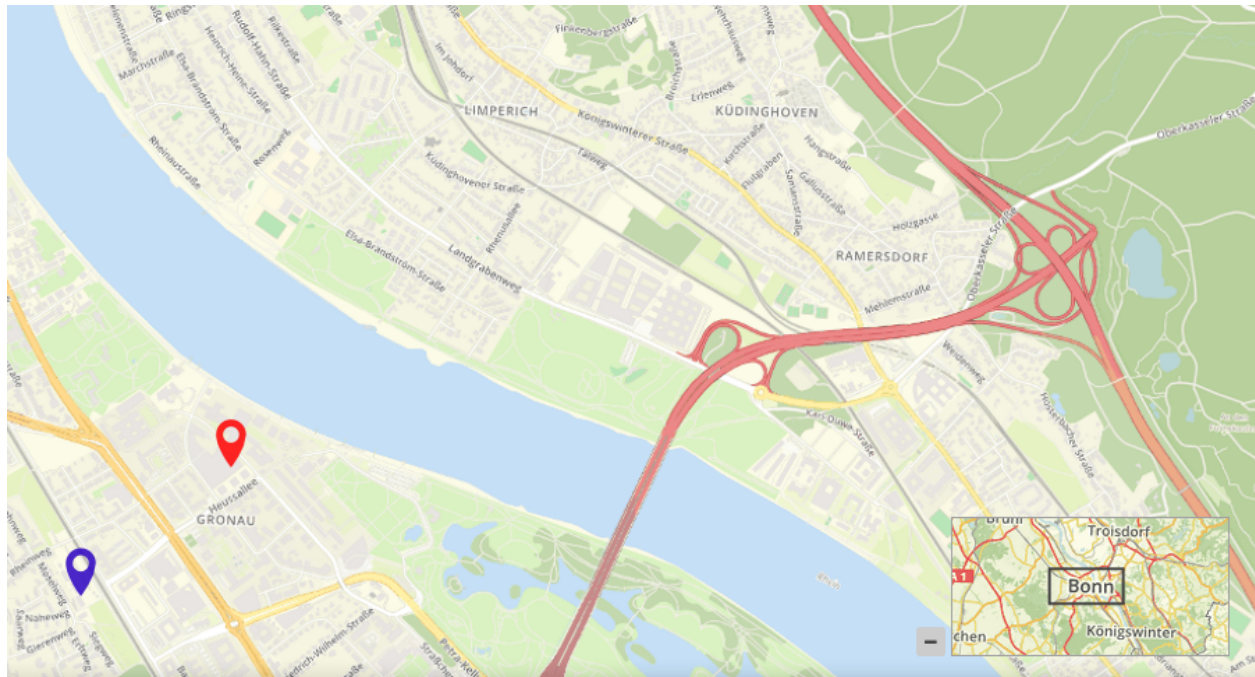
```

tooltip: 'Navigation Toolbar'      # text to use as tooltip
components: ["rotation",        # components of the navigation toolbar (default: all selected)
"zoom_max", "zoom_home",
"zoom_in_out", "zoom_slider"]
target: ~                        # ID of Map element to query
stepsize: 50                     # step value for pan
stepbypixel: false               # step type "by pixel"/"percent", false = percent (default: false)
anchor: left-top                 # navigation toolbar alignment (default: 'left-top')
                                     # use inline e.g. in sidebar
                                     # Options: 'inline', 'left-top', 'left-bottom', 'right-top', 'right-
↳ bottom'
draggable: true                  # element is draggable or not (default: true)
zoomHomeRestoresLayers: false    # resets layer settings (default: false)

```

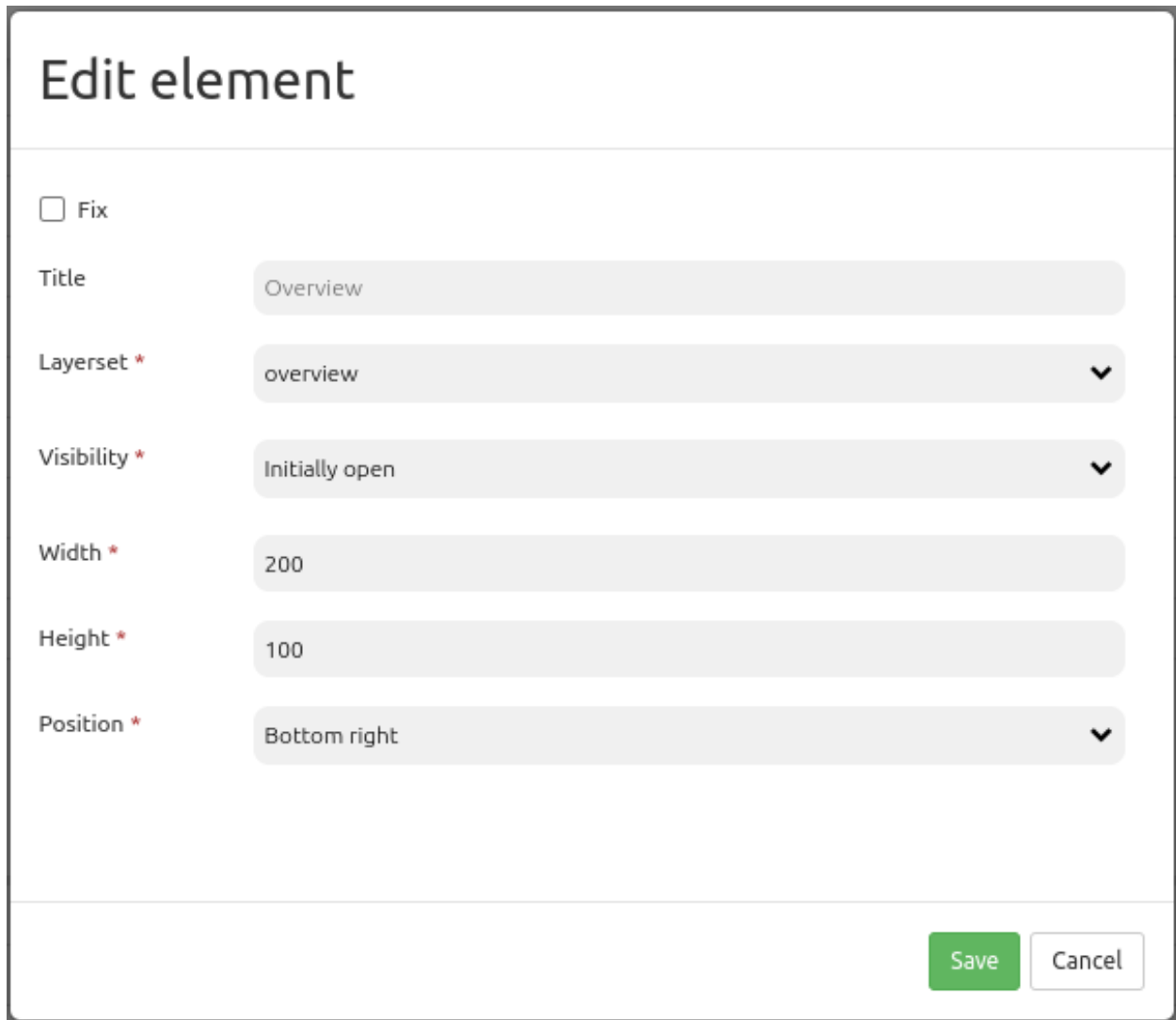
3.1.9 Overview

Mapbender provides an overview map in addition to its main map. This element can be individually adjusted in terms of size, position and zoom behaviour. The overview map refers to a specific layerset and the instance(s) it contains.



Configuration

The configuration dialog offers the following settings:



- **Fix:** Fixes the overview extent (default: false).
- **Title:** Title of the element, will be listed in "Layouts".
- **Layerset:** Refers to a previously defined layerset, e.g. overview.
- **Visibility:** Initially open/Initially closed/Permanently open - Defines whether the overview will be open or closed on start. The first two options will show a button to close the overview map. The third option provides an overview map and does not show a button. (default: Initially open)
- **Width/Height:** Width and height of the element.
- **Position:** Position of the overview map frame in the application; options are: top left, bottom left, top right and bottom right.

Configuration examples

Edit element

☒ Fix

Title

Overview

Layerset *

overview

▼

Visibility *

Initially open

▼

Width *

200

Height *

100

Position *

Bottom right

▼

Save

Cancel

The element provides various configuration options. In the example, the settings *Visibility Initially open* and *Fix* are activated. As a result, the element is displayed (i.e. maximized) when the application is opened and the view of the map is fixed. If the latter function is deactivated, the overview adapts as soon as the map is moved or the scale is changed. The start extent is displayed when the application is opened. In the example, the element has the title "Overview". It is necessary that the overview map is linked to a layerset. The following layersets are available:

Demo

Base data Layouts Layersets CSS Security

View

Filter

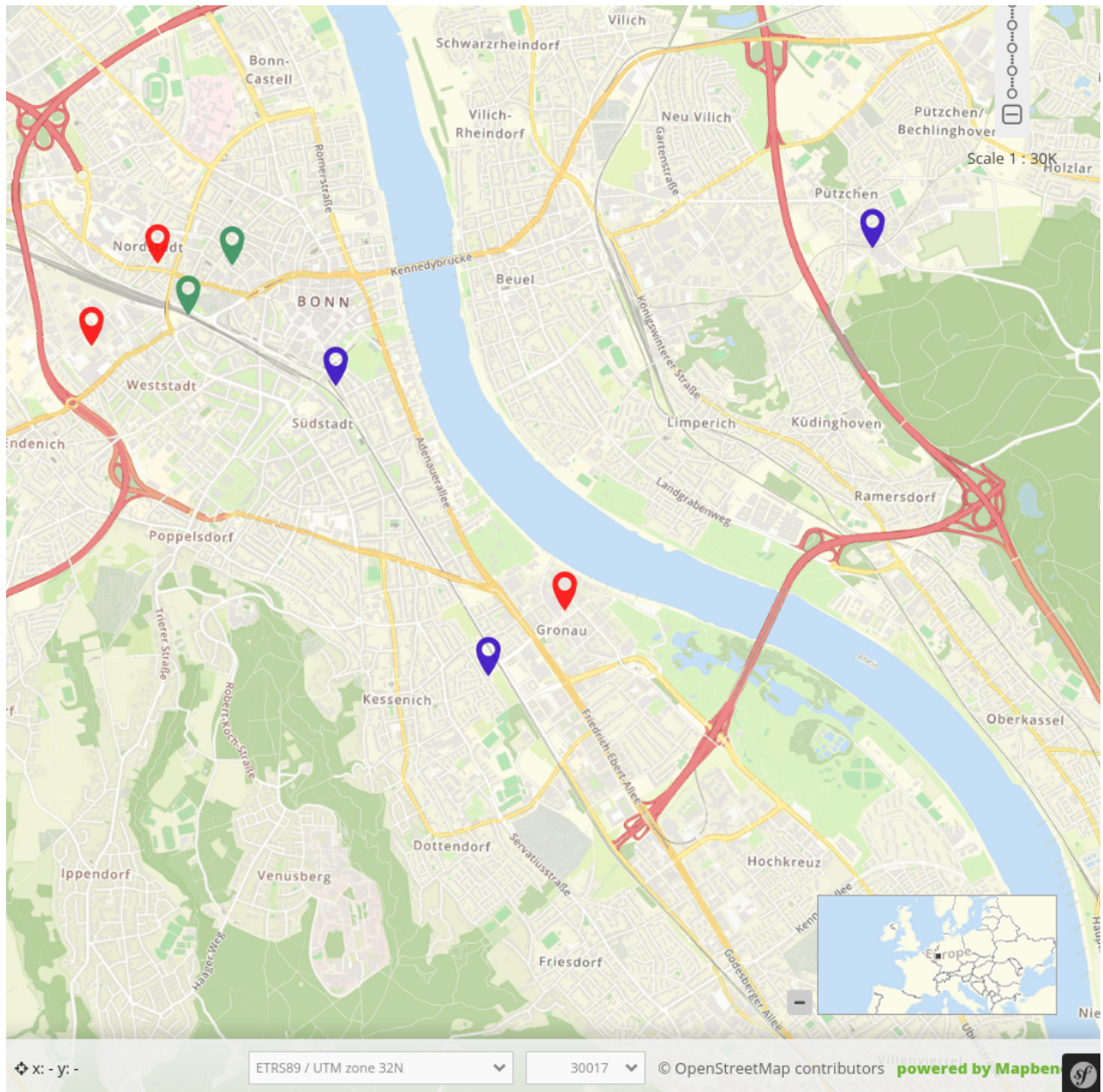
main 1.

Id	Title	Type	+	⚙	🗑
2/1	Mapbender Users (Mapbender Users)	WMS	📍	👁	⚙
1/2	OSM Demo (OSM Demo)	WMS	📍	👁	⚙

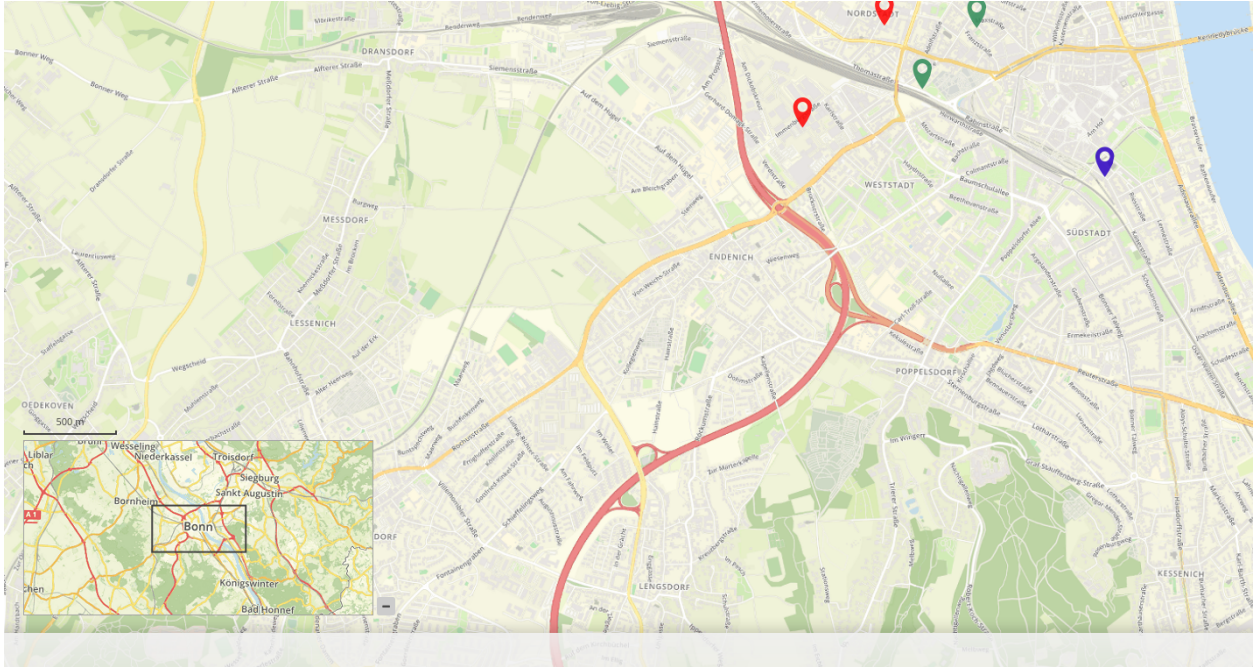
overview 2.

Id	Title	Type	+	⚙	🗑
1/3	OSM Demo (OSM Demo)	WMS	📍	👁	⚙

In this first example, the layerset “Overview” was chosen. Width and height of the element correspond to the default setting. The position is defined as “lower right”. The element looks like this in the application:



In the second example, an adjustment of the parameters (unchecking *Fix*, *Position*: left-bottom, *Width* 400 and *Height* 200) results in an overview like this:



YAML-Definition:

This template can be used to insert the element into a YAML application.

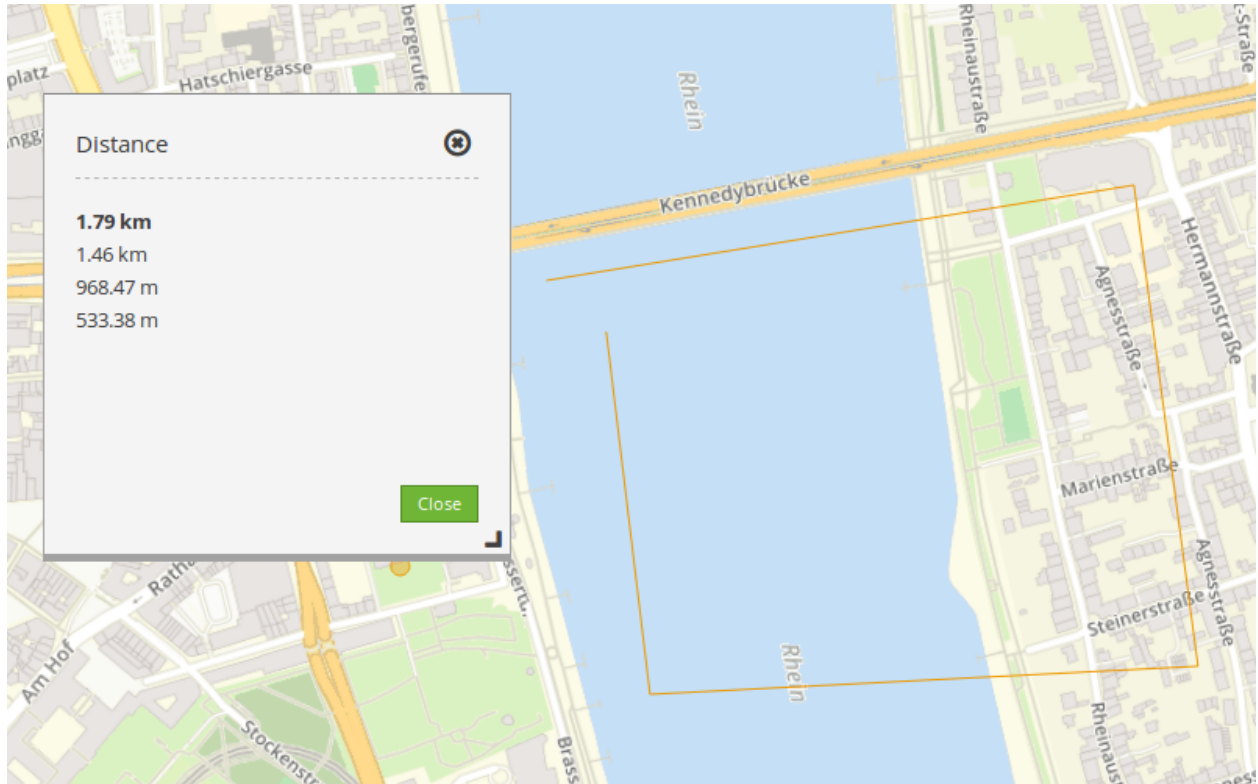
```

tooltip: 'Overview'           # text to use as tooltip
target: ~                    # Id of Map element to query
layerset: ~                  # refer to a layerset, e.g. overview, define the layerset first and refer
↪ to it
width: 200                   # overview width
height: 100                 # overview height
anchor: 'right-top'          # overview alignment (default: 'right-top')
                                # use inline e.g. in sidebar
                                # Options: 'inline', 'left-top', 'right-top', 'left-bottom', 'right-bottom'
visibility: open             # open/closed/open-permanent to open/closed on start will show a button to
↪ close the overview map (default: open), open-permanent will open the overview map and does not show a
↪ button
fixed: true                  # true/false to fix the overview extent (default: true)

```

3.1.10 Line/Area Ruler

The ruler is used to draw a line or area and display its length or area in a dialog. You need a [Button](#) to show this element. Each ruler element can only measure either lines or areas. To use both functions (measuring areas and lines) in an application, you need two buttons that are in the same pre-defined group.



Configuration

Add element

Map area - Line/Area Ruler

Title	Line/Area Ruler
Geometry *	line
Helptext	mb.core.ruler.help ?
Line width while drawing	3
Stroke color	rgba(51,153,204,1)
Stroke width (pixels)	2
Fill color	rgba(255,255,255,0.2) ?
Font color	rgba(0,0,0,1) ?
Font size	12 ?

Back
Add
Cancel

- **Title:** Title of the element. It will be displayed in the measuring window in the application itself.
- **Geometry:** Choose type of element: line or area. Mandatory field.
- **Helptext:** Displays a help text. The default value *mb.core.ruler.help* translates to “Double-click to end drawing” (depending on the browser’s display language).
- **Line width while drawing:** Pixel value that defines the stroke width during drawing.
- **Stroke color:** RGBA value that defines the stroke color. Can be changed using a color picker after clicking on the input field.
- **Stroke width (pixels):** Pixel value that defines the stroke width of the measured geometry.
- **Fill color:** RGBA value that defines the fill color of a measured area. Can be changed using a color

picker after clicking on the input field. When *Line* geometry is selected, this option has no effect.

- **Font color:** RGBA value that defines the font color of the calculated result displayed within the geometry. Can be changed using a color picker after clicking on the input field. When *Line* geometry is selected, this option has no effect.
- **Font size:** Numeric value that defines the font size of the area displayed within the geometry. When *Line* geometry is selected, this option has no effect.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: mb.core.ruler.tag.line           # Choose 'line', 'area' or another title
class: Mapbender\CoreBundle\Element\Ruler # Class of element
target: map                             # ID of Map element to query, e.g. 'map'
type: line                              # Choose type 'line' or 'area'
strokeColor: 'rgba(16, 101, 93, 0.8)'   # Choose rgba value (line and area)
fillColor: rgba(100, 100, 100, 0.5)     # Choose rgba value (area only)
fontColor: 'rgba(0,0,0,1)'              # Choose rgba value (area only)
fontSize: 14                            # Choose numeric value (area only)
strokeWidth: 4                           # Choose pixel value (line and area)
strokeWidthWhileDrawing: 3               # Choose pixel value (line and area)
```

3.1.11 Scale Bar

The Scale bar is a small line which represents the current map scale graphically.



Configuration

Add element

Content - Scale bar

Title	<input type="text" value="Scale bar"/>
Max width *	<input type="text" value="200"/>
Units *	<input type="text" value="kilometer"/> ▼
Position *	<input type="text" value="Bottom right"/> ▼

[Back](#)[Add](#)[Cancel](#)

- **Title:** Title of the element. The title will be listed in “Layouts”.
- **Max width:** The maximum width of the scale bar (default: 200px).
- **Units:** Scale bar units ‘kilometer’ or ‘miles’ (default: kilometer).
- **Position:** Scale bar alignment (default: Bottom right).

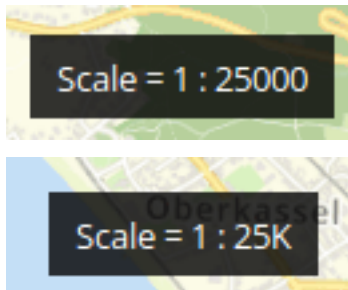
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
tooltip: 'Scale Bar'           # text to use as tooltip
target: ~                     # ID of the Map element to query
anchor: 'inline'/'left-top'/' # scale bar alignment (default: 'right-bottom')
      'left-bottom'/'right-top'/' # use inline, e.g. in sidebar
      'right-bottom'
position: ['10px', '10px']    # scale bar position (default: x=20px, y=20px)
maxWidth: 200                 # the maximum width of the scale bar (default: 200px)
units: ['km']                 # scale bar units 'kilometer' and/or 'miles' (ml) (default: ['km'])
```

3.1.12 Scale Display

The Scale Display shows the current map scale (1:1K or 1:1000).



Configuration

Add element

Footer - Scale display

Title

Scale display

Scale prefix

Scale =

☐ Unit prefix

Back

Add

Cancel

- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons.
- **Scale prefix:** Prefix shown with scale.
- **Unit prefix:** Prefix shown with unit, e.g. 1K as 1000.
- **Position:** Scale Display alignment (default: ‘bottom right’).

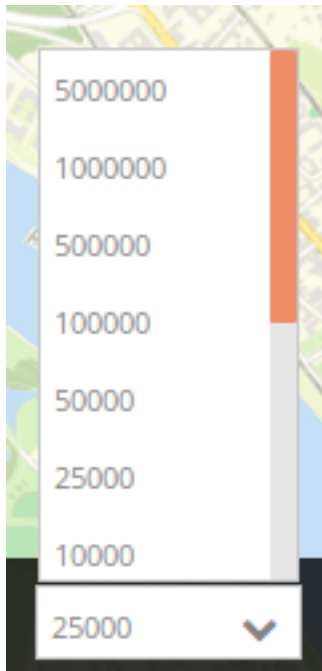
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
tooltip: 'ScaleDisplay'           # text to use as tooltip
target: ~                        # ID of Map element to query
anchor: 'inline'/'left-top'/'left-bottom'/'right-top'/'right-bottom' # Scale Display alignment, default is 'right-bottom'
# use inline, e.g. in sidebar
scalePrefix: Scale               # prefix, shown with scale
unitPrefix: false                # prefix, shown with unit, e.g. 1K as 1000
```

3.1.13 Scale Selector

This element displays a selectbox with scales. The map scale changes when a different value from the selectbox is chosen. The selectbox only offers scales that have been defined in the map element.



Configuration

Add element

Toolbar - Scale selector

☐ Show label

Title

Scale selector

Tooltip

mb.core.scaleselector.class.title

Back

Add

Cancel

- **Show label:** Displays label of the Scale Selector (default: false).
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Tooltip:** Text to use as tooltip.

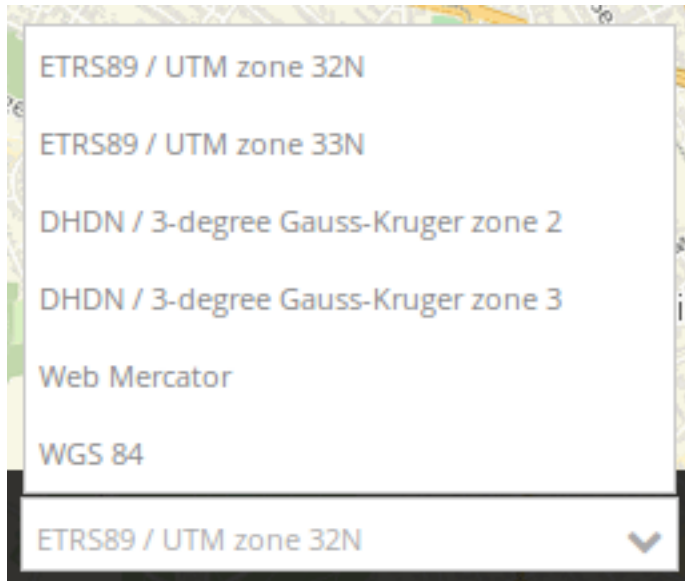
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
tooltip: "Scale selector" # text to use as tooltip
target: ~                # Id of Map element to query
label: false             # false/true to label the scale selector (default: false)
```

3.1.14 SRS Selector

The spatial reference system selector changes the map's spatial reference system. The selectbox offers SRS that are defined within the *Map element*.



Configuration

Add element

Content - SRS selector

☐ Show label

Title

SRS selector

Tooltip

mb.core.srsselector.class.title

Back

Add

Cancel

- **Show label:** Labels the SRS Selector (default: false).
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Tooltip:** Text to use as tooltip.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
tooltip: 'SRS Selector' # text to use as tooltip
label: false           # true/false to label the SRS Selector (default: false)
target: ~              # Id of the Map element to query
```

3.2 Editing

3.2.1 Digitizer

The digitizer element allows the creation and editing of points, lines and areas on a map. In contrast to sketch, this element saves geometries in a table in a database. Currently Mapbender supports PostgreSQL. SpatiaLite and Oracle are available experimentally. Development was carried out in a way that digitizer can be extended to include other data sources such as OGC WFS as well.

In order to use the digitizer, one has to define a specific yaml-definition:

Setting up of Digitizer

You can only use the element in the sidepane.

Element editieren

Title*

Digitizer

Target

Main Map

Schemes

```

1 poi:|
2   label: 'point digitizing'
3   inlineSearch: true
4   maxResults: 500
5   featureType:
6     connection: gisdb
7     table: punkt
8     uniqueId: gid
9     geomType: point
10    geomField: geom
11    srid: 4326
12  openFormAfterEdit: true
13  zoomScaleDenominator: 500
14  allowEditData: true
15  allowDelete: true
16  allowDigitize: true
17  useContextMenu: true
18  toolset:
19    -

```

Aktualisieren Abbrechen

- **Title:** Title of the element. This is displayed in the layouts list and allows you to define several elements.
- **Target:** (ID) of the map.
- **Schemes:** YAML-Definition of the element Digitizer

The Digitizer needs access to a database where the editable tables are. You have to define a new database configuration to be able to connect with the geo database. Read more about this under [YAML Configuration \(Configuration and Application files\)](#).

The definition of the Digitizer is done in YAML syntax in the textarea configuration at schemes. Here you define the database connection, the editable table, the form to display the table, the attribute form and other behavior. If errors occur in the database, fields or form, various error messages appear. Via the normal call and app.php comes a general error message. If you want to reproduce the exact error,

you should call the page via app_dev.php. In this case, detailed error messages about the error behavior appear.

- **debug:** Display error messages, e.g. syntax error in SQL [experimental]

```
poi:
  [...]
  debug: true
  [...]
```

YAML-Definition for the element Digitizer in the textarea schemes

In the following YAML block, the example definition for three Digitizer forms is included. Copy the following block into your Digitizer element to test the capture of points, lines, and polygons. You must first create the database connection and the three demo tables. The SQL commands for creating the tables can be found below. The functionality of the built-in features and additional functions are explained in more detail in this example.

```
poi:
  label: point digitizing
  inlineSearch: true
  featureType:
    connection: search_db
    table: poi
    uniqueId: gid
    geomType: point
    geomField: geom
    srid: 4326
  openFormAfterEdit: true
  zoomScaleDenominator: 500
  allowEditData: true
  allowDelete: true
  allowDigitize: true
  useContextMenu: true
  toolset:
    - type: drawPoint
    - type: moveFeature
  popup:
    title: point test suite
    width: 500px
  searchType: currentExtent
  tableFields:
    gid: {label: Nr. , width: 20%}
    name: {label: Name , width: 80%}
  styles:
    default:
      strokeWidth: 2
      strokeColor: '#0e6a9e'
      fillColor: '#1289CD'
      fillOpacity: 1
      fillWidth: 2
      pointRadius: 10
    select:
      strokeWidth: 3
      strokeColor: '#0e6a9e'
      fillOpacity: 0.7
```

(continues on next page)

(continued from previous page)

```

        pointRadius: 10
formItems:
- type: tabs
  children:
    - type: form
      title: Basic information
      css: {padding: 10px}
      children:
        - type: label
          title: Welcome to the digitize demo. Try the new Mapbender feature!
        - type: input
          title: Name
          mandatory: true
          name: name
          mandatoryText: Please give a name to the poi.
          infoText: "Help: Please give a name to the new object."
        - type: input
          title: Title
          mandatory: false
          name: title
          mandatoryText: Please give a title to the poi.
        - type: textArea
          name: abstract
          title: Abstract
          placeholder: 'please edit this field'
        - type: select
          title: Type
          name: type
          options: {A: A, B: B, C: C, D: D, E: E}
        - type: breakLine
    - type: form
      title: Personal information
      css: {padding: 10px}
      children:
        - type: label
          title: Please give us some information about yourself.
        - type: fieldSet
          children:
            - type: input
              title: Firstname
              name: firstname
              css: {width: 30%}
            - type: input
              title: Lastname
              name: lastname
              css: {width: 30%}
            - type: input
              title: E-Mail
              name: email
              css: {width: 40%}
        - type: select
          multiple: false
          title: Interests
          name: interests
          options: {maps: maps, reading: reading, swimming: swimming, dancing: dancing, beer:
↪beer, flowers: flowers}

```

(continues on next page)

(continued from previous page)

```

        - type: date
          title: favorite Date
          name: date_favorite
          mandatory: true
          css: {width: 25%}
        - type: breakLine
        - type: breakLine
        - type: checkbox
          name: public
          value: true
          title: public (this new object is public)
line:
  label: line digitizing
  inlineSearch: true
  featureType:
    connection: search_db
    table: lines
    uniqueId: gid
    geomType: line
    geomField: geom
    srid: 4326
  openFormAfterEdit: true
  allowDelete: true
  useContextMenu: true
  toolset:
    - type: drawLine
    - type: modifyFeature
    - type: moveFeature
  popup:
    title: line test suite
    width: 500px
  searchType: currentExtent
  tableFields:
    gid: {label: Nr. , width: 20%}
    name: {label: Name , width: 80%}
  styles:
    default:
      strokeWidth: 2
      strokeColor: '#0e6a9e'
      fillColor: '#1289CD'
      fillOpacity: 1
      fillWidth: 2
      pointRadius: 10
    select:
      strokeWidth: 3
      strokeColor: '#0e6a9e'
      fillOpacity: 0.7
      pointRadius: 10
  formItems:
    - type: form
      title: Basic information
      css: {padding: 10px}
      children:
        - type: label
          title: Welcome to the digitize demo. Try the new Mapbender feature!
        - type: input

```

(continues on next page)

(continued from previous page)

```

        title: Name
        name: name
        mandatory: true
        mandatoryText: Please give a name to the new object.
        infoText: "Help: Please give a name to the new object."
    - type: select
      title: Type
      name: type
      options: {A: A, B: B, C: C, D: D, E: E}
polygon:
  label: polygon digitizing
  inlineSearch: true
  featureType:
    connection: search_db
    table: polygons
    uniqueId: gid
    geomType: polygon
    geomField: geom
    srid: 4326
  openFormAfterEdit: true
  allowDelete: false
  useContextMenu: true
  toolset:
    - type: drawPolygon
    - type: drawRectangle
    - type: drawDonut
    - type: drawEllipse
    - type: drawCircle
    - type: modifyFeature
    - type: moveFeature
  popup:
    title: polygon test suite
    width: 500px
  searchType: currentExtent
  tableFields:
    gid: {label: Nr. , width: 20%}
    name: {label: Name , width: 80%}
  styles:
    default:
      strokeWidth: 2
      strokeColor: '#0e6a9e'
      fillColor: '#1289CD'
      fillOpacity: 1
      fillWidth: 2
      pointRadius: 10
    select:
      strokeWidth: 3
      strokeColor: '#0e6a9e'
      fillOpacity: 0.7
      pointRadius: 10
  formItems:
    - type: form
      title: Basic information
      css: {padding: 10px}
      children:
        - type: label

```

(continues on next page)

(continued from previous page)

```

    title: Welcome to the digitize demo. Try the new Mapbender feature!
  - type: input
    title: Name
    mandatory: true
    name: name
    mandatoryText: Please give a name to the new object.
    infoText: "Help: Please give a name to the new object."
  - type: select
    title: Type
    name: type
    options: {A: A, B: B, C: C, D: D, E: E}

```

SQL for the demo tables

The following SQL commands must be executed in your database. You create three demo tables so that the individual functions can be tested using the YAML definition shown above.

```

create table public.poi (
  gid serial PRIMARY KEY,
  name varchar,
  type varchar,
  abstract varchar,
  public boolean,
  date_favorite date,
  title varchar,
  firstname varchar,
  lastname varchar,
  email varchar,
  interests varchar,
  user_name varchar,
  group_name varchar,
  modification_date date,
  my_type varchar,
  file_reference varchar,
  x float,
  y float,
  city varchar,
  geom geometry(point,4326)
);

```

```

create table public.lines (
  gid serial PRIMARY KEY,
  name varchar,
  type varchar,
  abstract varchar,
  public boolean,
  date_favorite date,
  title varchar,
  firstname varchar,
  lastname varchar,
  email varchar,
  interests varchar,
  length float,
  category varchar,

```

(continues on next page)

(continued from previous page)

```
user_name varchar,  
group_name varchar,  
modification_date date,  
my_type varchar,  
file_reference varchar,  
x float,  
y float,  
city varchar,  
geom geometry(linestring,4326)  
);
```

```
create table public.polygons (  
gid serial PRIMARY KEY,  
name varchar,  
type varchar,  
abstract varchar,  
public boolean,  
date_favorite date,  
title varchar,  
firstname varchar,  
lastname varchar,  
email varchar,  
interests varchar,  
area float,  
category varchar,  
user_name varchar,  
group_name varchar,  
modification_date date,  
my_type varchar,  
file_reference varchar,  
x float,  
y float,  
city varchar,  
geom geometry(polygon,4326)  
);
```

Configuration

The following chapters explain the individual components of the Digitizer that build up the base-structure and which can be used in the formular.

Feature basic definition

A basic definition, here for the poi-example, may look like the following snippet:

```
poi:  
  label: point digitizing  
  minScale: 5000  
  featureType:  
    connection: search_db  
    table: poi  
    uniqueId: gid  
    geomType: point
```

(continues on next page)

(continued from previous page)

```

    geomField: geom
    srid: 4326
    filter: interests = 'maps'
  openFormAfterEdit: true
  zoomScaleDenominator: 500
  allowEditData: true
  allowDelete: true
  allowDigitize: true
  [...]
  popup:
    [...]

```

The possible options are:

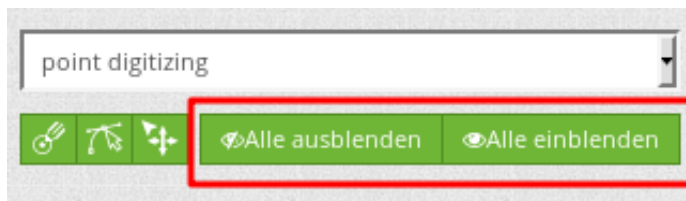
- **label:** Label of the Digitizer popup
- **minScale:** Minimum scale, where the features should be displayed in the map (e.g. minscale: 5000 = show from a scale 'over' 1:5000, when zooming out).
- **featureType: Connection to the database**
 - connection: Name of the database-connection from the parameters/config.yml
 - table: Table-name in which the FeatureTypes are stored
 - uniqueId: Column-name with the unique identifier
 - geomType: Geometry-type
 - geomField: Column-name in which the geometry is stored
 - srid: Coordinate-system in EPSG-code
 - filter: Data filters for values in a defined column, e.g. filter: interests = 'maps'
- **openFormAfterEdit:** After creating a geometry the form popup is opened automatically to insert the attribute data (default: true)
- **zoomScaleDenominator:** Zoom-scales to use for zooming to a feature.
- **allowEditData:** Allow or disable functions to edit or remove data. [true/false]. The Save button is always visible.
- **allowDigitize:** Allow to save data [true/false].
- **allowDelete:** Allow to delete data [true/false]. The Delete button is always visible.
- **allowDigitize:** Allow to create new features [true/false]. if false, no Digitizer buttons will occur (new Point, move, etc.).
- **useContextMenu:** Show the context-menu of a feature. [true/false]
- **allowCancelButton:** Show the Cancel button [true/false]. See *Save, Delete, Cancel*.
- **allowDeleteByCancelNewGeometry:** If true: When you create a new feature, the Cancel button will behave like the Delete button: The feature is removed from the map and the table. This is not the case if you edit an existing feature. [true/false]
- **displayOnInactive:** The current FeatureType will still be displayed on the map, although the Digitizer is deactivated in the Sidepane (Accordion, Tabs) [true/false]. If switched to true, this option is a bit tricky, due to the still activated Digitizer events but will be helpful for experienced users.
- **allowLocate:** Navigation to a feature via the tabs-keyboard-button, simple for operation without mouse. [True / false]. An extra "zoomTo" Button is displayed for each feature.

Nr.	Name			
2	A			
7	as			
8	asdas			
9	afgagaga			

- **allowChangeVisibility:** Allow to change the visibility of one feature in the map (visible / invisible). [true/false]. An Eye Symbol is displayed, which allows to hide or display each feature individually.

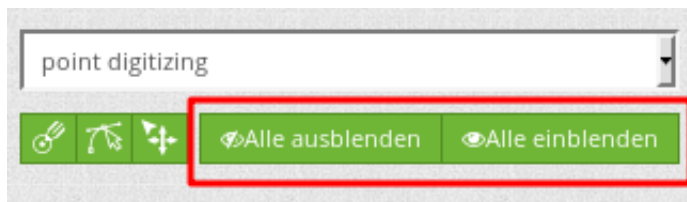
Nr.	Name			
2	A			
7	as			
8	asdas			

- **showVisibilityNavigation:** Allow to change the visibility of all features in the map (visible / invisible). [true/false]

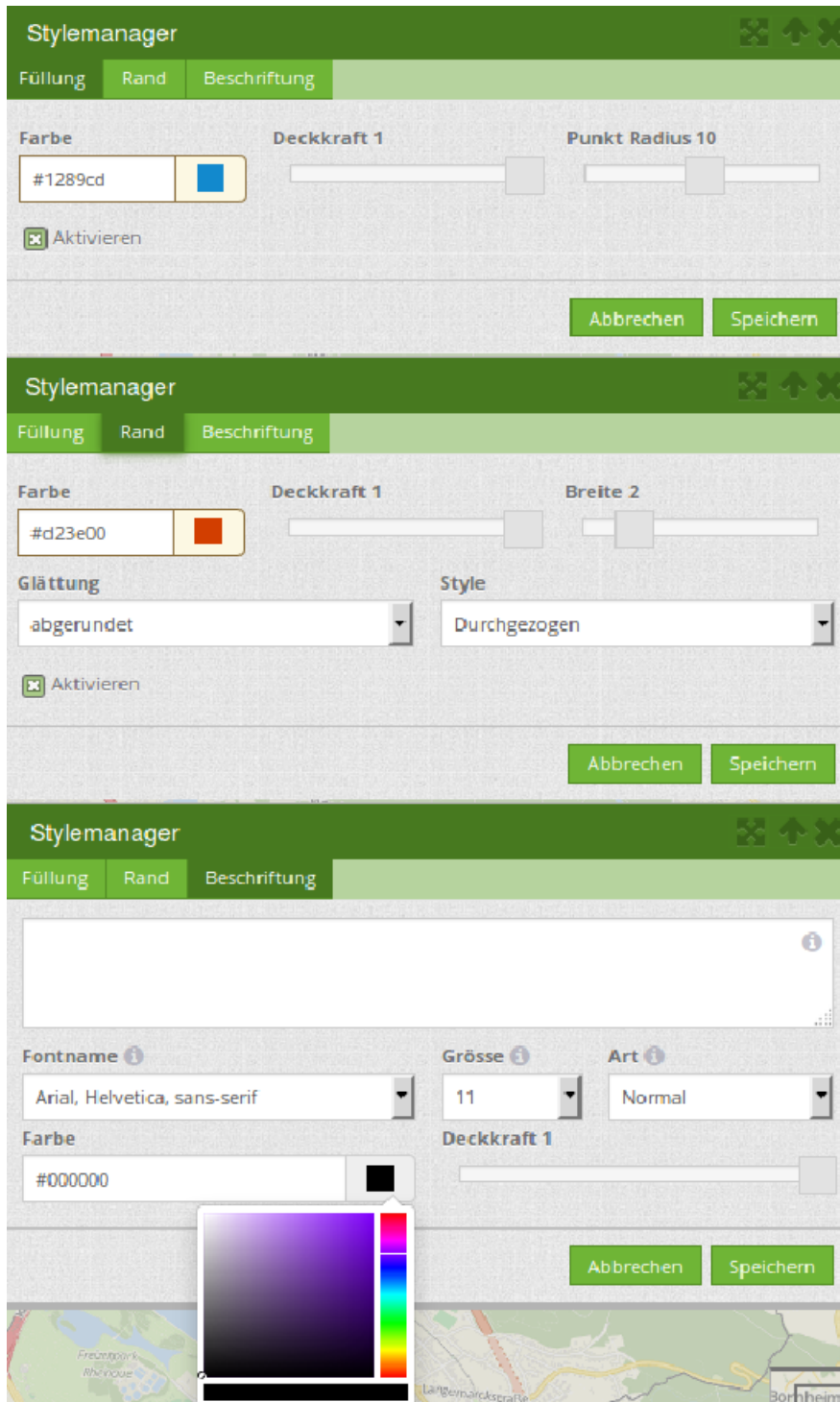


Experimental:

- **allowCustomerStyle:** Allow user-specific styles for features in the map [true/false]. This feature is experimental: For each feature you can set unique styles.

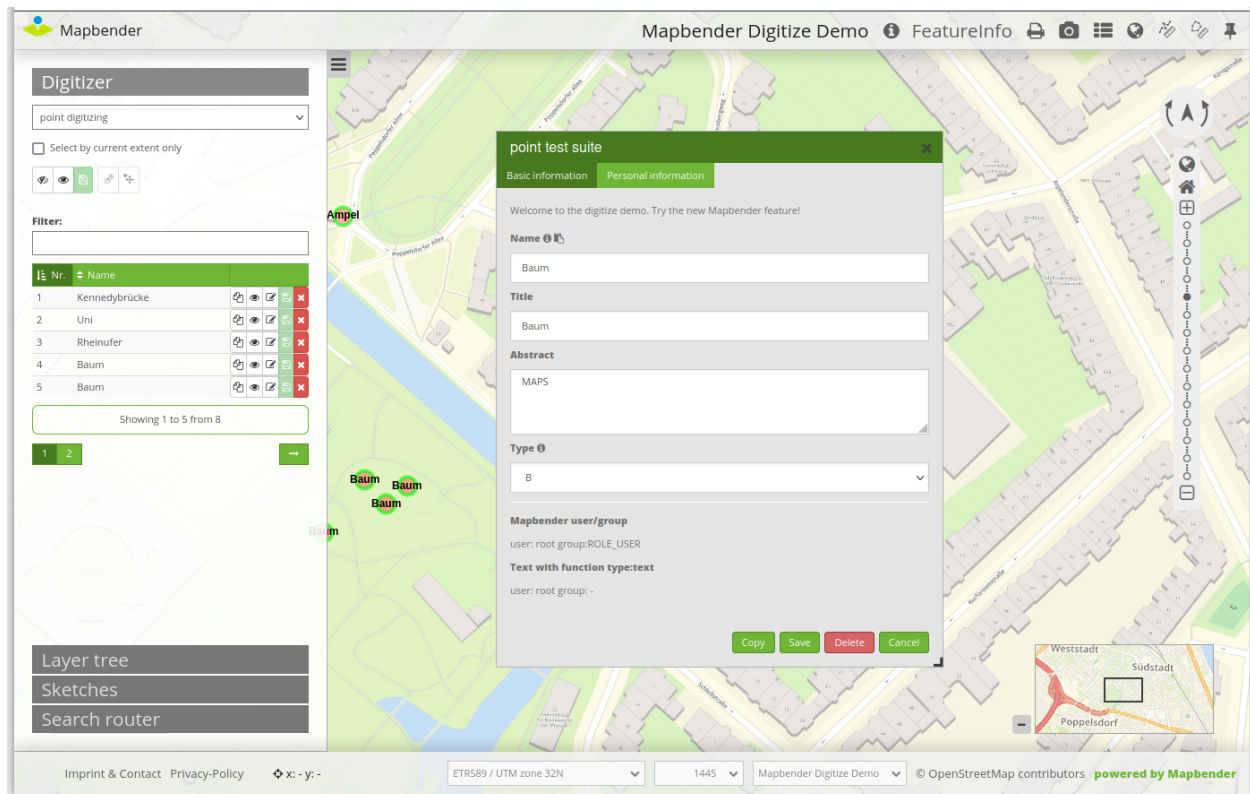


A style-manager is used to let you set the unique styles.



Definition of the popup

In connection with the digitization, very complex forms can be generated for the acquisition of data.



The following option for the construction of the forms are available:

- Define more than one feature type for digitization. You can switch from one feature type to another with a select box.
- It is possible to define a filter to only query a subset of your database table.
- Textfields
- Selectboxes, Multiselectboxes
- Checkboxes and Radiobuttons
- Textareas
- Datepicker
- File upload and Image Display
- Definition of tabs
- Definition breakLines
- Definition of Text
- Mandatory fields, regular expressions to validate the field input
- Definition of help texts
- Refresh after save
- Possibility to copy entered information from a form into the clipboard via a button

point test suite ✕

Basic information Personal information

Welcome to the digitize demo. Try the new Mapbender feature!

Name ⓘ

Baum

Title

Baum

Abstract

MAPS

Type ⓘ

B ▼

Mapbender user/group

user: root group:ROLE_USER

Text with function type:text

user: root group: -

Copy Save Delete Cancel

```

popup:                                     # Define the form as a popup. Further❑
↪experimental adaptations here: http://api.jqueryui.com/dialog/
  title: POI                               # Definition of the popup title
  height: 400                             # height of the popup
  width: 500                               # width of the popup

  #modal: true                               # Everything except the form window is❑
↪grayed out and the position and size of the window is fixed for the duration of the data collection.
  #position: {at: "left+20px", my: "left top-460px"} # Position of the popup in the browser❑
↪area

```

Definition of the feature table

The Digitizer provides an object table. It can be used to navigate to features (zoom on the objects) and open the editing form. The object table can be sorted. The width of the individual columns can optionally be specified in percent or pixels.

- **tableFields: define the columns for the feature table.**
 - definition of a colum: [table column]: {label: [label text], width: [css-definition, like width]}
- **searchType:** search extent in the map, display of all features in the result table or only features displayed in the current extent [all / currentExtent] (default: currentExtent).
- **showExtendSearchSwitch:** Activate or deactivate the display of the searchType selectbox for searching in the curret extent [true/false]
- **view: Settings for the object result table**
 - **type:** Templatenamename [table]
 - **settings:** Settings for the functions of the result table *(Newly added, not fully documented!)*

You can find more detailed information on possible configurations under <https://datatables.net/reference/option/>.

```
searchType: currentExtent
showExtendSearchSwitch: true
tableFields:
  gid: {label: Nr. , width: 20%}
  name: {label: Name , width: 80%}
view:
  type: table
  settings:
    info: true
    processing: false
    ordering: true
    paging: true
    selectable: false
    autoWidth: false
    order: [[1, "asc"]] # 1 | 2 presort columns
```

Tabs (type tabs)

Form elements can be placed unto different Tabs. The formItem type “tabs” is used for this.

```
formItems:
  - type: tabs # Type tabs creates tabs in the popup
    children: # The tabs are defined as sub-objects
    ↪(children) of the form.
      - type: form
        title: Basic information # title of the tabs
        css: {padding: 10px}
        children: # Multiple subobjects in groups can be
        ↪used to arrange data in the form next to each other
          - type: label
            title: Welcome to the digitize demo. Try the new Mapbender feature!
          ...
```

For each input field the CSS-behavior and styling information can be assigned, regardless of the type. This can be used, for example, to highlight important fields or to fill an attribute field when editing another field.

parameters:

- load, focus, blur
- input, change, paste
- click, dblclick, contextmenu
- keydown, keypress, keyup
- dragstart, ondrag, dragover, drop
- mousedown, mouseenter, mouseleave, mousemove, mouseout, mouseover, mouseup
- touchstart, touchmove, touchend, touchcancel

```
formItems:
- type: tabs
  children:
  - type: form
    [...]
    - type: input
      name: firstname
      title: Firstname
      css: {width: 30%}
      input: |
        var inputField = el;
        var form = inputField.closest(".modal-body");
        var datenkennungField = form.find("[name='datenkennung']");
        datenkennungField.val(inputField.val());
      focus: |
        var inputField = el;
        var form = inputField.closest(".modal-body");
        form.css("background-color", "#ffc0c0");
      blur: |
        var inputField = el;
        var form = inputField.closest(".modal-body");
        form.css("background-color", "transparent");
    - type: date
      name: date
      title: Date
      css: {width: 30%}
      # Highlight the year if you edit the date-field and autom. insert the year from the date
      change: |
        var inputField = el;
        var form = inputField.closest(".modal-body");
        var yearField = form.find("[name='year']");
        var year = inputField.val().match(/\d+$/)[0];
        yearField.val(year);
        yearField.css("background-color", "#ffc0c0");
```

Text fields (type input)

```
- type: input                                # element type definition
title: Title for the field                  # labeling (optional)
name: column_name                          # reference to table column (optional)
copyClipboard: false                      # true/false. if true, specify a button that copies
↪ entered information to the clipboard (default: false) (optional)
mandatory: true                           # specify mandatory field (optional)
mandatoryText: You have to provide information. # text occurs if the column is empty on save
cssClass: 'input-css'                     # css class to use as style for the input field
↪ (optional)
value: 'default Text'                     # define a default value (optional)
placeholder: 'please edit this field'     # placeholder appears in the field as information
↪ (optional)
```

Selectbox (selectbox or multiselect [type select])

By defining a selectbox, predefined values can be used in the form. You can choose between a selectbox with a selectable entry (type select) or a multiselectbox with several selectable entries (type multiselect).

(1) select - one selectable entry

```
- type: select                                # element type definition
title: select a type                        # labeling (optional)
name: my_type                              # reference to table column (optional)
copyClipboard: false                      # specify button that copies chosen values to the clipboard
↪ (optional). [true/false] (default: false).
multiple: false                            # define a multiselect (default: false)
options:                                  # definition of the options (key, value)
  1: pub
  2: bar
  3: pool
  4: garden
  5: playground
```

(2) multiselect - several selectable entries

The Multiselect-Box is activated by the attribute “multiple: true”. You can choose multiple entries in the selectbox. The usage and their requirements of the database may vary. In general with the example above, you can switch the “interests” in the POIs to multiselects. The database fields is still a character varying.

```
-
type: select
multiple: true
title: Interests
name: interests
copyClipboard: false
options:
  maps: maps
  reading: reading
  swimming: swimming
  dancing: dancing
  beer: beer
  flowers: flowers
```

Notes: From Digitizer version 1.2 and up, the multiple selection provides an easier mechanism to choose an entry, which also allows a search in the drop-down-list. The navigation through the list is possible via keyboard. Possible entries are highlighted during typing. An already chosen entry can be removed by clicking the small “x” symbol. Check up-to-date information about digitizer versions: <https://github.com/mapbender/mapbender-digitizer>

The SQL (if maps and reading were chosen):

```
gisdb=> select interests from poi where gid=3;
interests
-----
maps,reading
(1 row)
```

The keywords are saved in the database (for example: “dancing: Tanzen” and “flowers: Blumen” stores “dancing,flowers”). It is possible to copy several values to the clipboard with CopyClipboard: true.

```
- type: select           # element type definition
  title: select some types # labeling (optional)
  name: my_type           # reference to table column (optional)
  copyClipboard: true     # Button which copies chosen values to the clipboard (optional)
  multiple: true          # define a multiselect (default: false)
  options:
    a: a                 # definition of the options (key, value)
    b: b
    c: c
```

Get the options for the selectbox via SQL

With a SQL request, the values of the selectbox can be directly pulled from the database. In this case, the key value mapping is not possible and only the indices of the entries can be stored.

```
- type: select                                # element type definition
title: select some types                     # labeling (optional)
name: my_type                               # reference to table column
connection: connectionName                 # Define a connection selectbox via
↪SQL
sql: 'SELECT DISTINCT key, value FROM tableName order by value' # get the options of the
```

Text/Label (type label)

```
- type: label                                # element type definition, label writes a non-editable
↪text to the form window.
text: 'Please give information about the poi.' # define a text
```

Text (type text)

Texts can be defined as a label in the form. In this case, fields of the data source can be accessed by using JavaScript.

```
- type: text                                # Type text for generating dynamic texts from the database
title:      Name                          # Label (optional)
name:      name                          # Name of the field (optional)
css:       {width: 80%}                  # CSS definition (optional)
text: data.gid + ': ' + data.name
# Text definition in JavaScript
# data - data is the object, that gives access to all fields.
# z.B.: data.id + ':' + data.name
```

Textareas (type textarea)

Similar to the text field via type input (see above), text areas can be created that can contain several lines using type textArea.

```
- type: textArea                            # Typ textArea creates a text area
rows: 4                                  # Number of rows for the text area that appears when the form is
↪opened. Field can be expanded by mouse in the form.
name: beschreibung                       # table column
title: Bestandsaufnahme Bemerkung        # Label (optional)
```

Breaklines (type breakLine)

```
- type: breakLine           # element type definition, will draw a line
```

Checkboxes (type checkbox)

```
- type: checkbox           # Type checkbox creates a checkbox. When activated, the specified value (here
↪ 'TRUE') is written to the database.
title: Is this true?      # Label (optional)
name: public             # table column
value: true              # parameter when activating the checkbox is stored in DB (here 'TRUE').
checked: false           # [true/false] behaviour of the checkbox (default: false). Defines whether
↪ checkbox is pre-selected (checked) on load
```

Mandatory fields

The notes for a mandatory field appear above the used fields. In the case of a missing entry in a defined mandatory field, this will be marked in red and (if defined) a speech bubble will appear. The object can not be saved if mandatory data is missing.

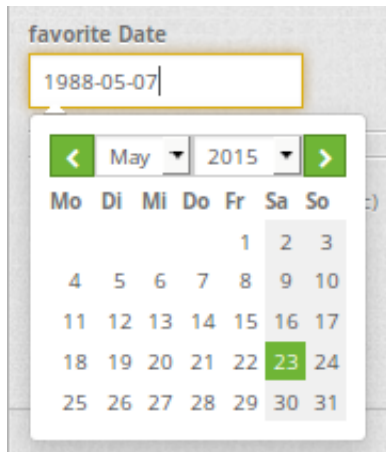
Note: When using multiple tabs in the form, the creator may set an entry incorrectly on a non-visible tab in a mandatory field, so the saving process does not work. No error message appears outside the form. The applicant has to check the information in the form (label: red border/speech bubble with reference) before it can be stored correctly.

```
- type: [Angabe zum Feldtyp]      # Each field can be made mandatory

mandatory: true                 # true - field has to be set. Else you can't save the object.
↪ Regular expressions are possible too - see below.
mandatorytitle: Mandatory info!  # Text that appears in the field when the field is not filled
↪ or filled with an invalid value.
mandatoryText: Please choose a type! # Text that is displayed in a speech bubble above the field
↪ when the field is not filled when it is saved or invalid.
mandatory: /\w+$/gi             # You can define a regular expression to check the input for a
↪ field. You can check e.g. for email or numbers. Read more http://wiki.selfhtml.org/wiki/JavaScript/Objekte/RegExp

# Check if input is a number
mandatory: /^[0-9]+$/
mandatoryText: Only numbers are valid for this field!
```

Date-picker (type date)



- **type:** date *# click in the textfield opens a datepicker*
- title:** favorite Date *# Label (optional)*
- name:** date_favorite *# data table*

Helptexts to the form-elements (attribute infotext)

The infotext can appear over every field, regardless of whether this is a mandatory field or not. If a infotext is specified, an info button appears above the field. Clicking on this button opens the information text.

- **type:** [type name] *# every field, regardless of whether*
- ↪ *this is a mandatory field or not*
- infoText:** Please note - only numbers are valid for this field. *# Notice which will be displayed by i-*
- ↪ *symbol*

Element groups (type: fieldSet)

Elements can be grouped together in one row to provide logical connections or save space. To define a group you have to set type fieldSet and afterwards define the children which shall be grouped.

For each children you can define a width to control the pace for each element.

- **type:** fieldSet *# Grouping of fields, regardless of field type*
- children:** *# Define the group elements by children*
 - **type:** input
 - title:** Firstname
 - name:** firstname
 - css:** {width: 30%} *# Specifies the width of the group element. Together, the elements should*
 - ↪ *be 100%.*
 - **type:** input
 - title:** Lastname
 - name:** lastname
 - css:** {width: 30%}
 - **type:** input

(continues on next page)

(continued from previous page)

```

title: E-Mail
name: email
css: {width: 40%}

```

Coordinate Fields (type coordinates)

For visual verification, manual entry or manual editing of point coordinates, the *coordinates* element can be used.

EPSG angeben	Latitude:	Longitude:
EPSG:3857 (Pseudo Mercator)	6142784.4091751	1553813.9524897

```

- type: coordinates
  title_epsg: 'EPSG angeben'
  title_longitude: 'Longitude'
  title_latitude: 'Latitude'
  coordinatesFieldsOrder: ['eps', 'x', 'y']
  title: ''
  css: {width: 100% }
  epsgCodes: [['EPSG:3857', 'EPSG:3857 (Pseudo Mercator)'], ['EPSG:4326', 'EPSG:4326 (WGS84)']]

```

The possible special properties are:

- **title_epsg**: the caption for the field to select the EPSG code of the coordinates
- **title_longitude**: the title of the longitude field
- **title_latitude**: the title of the latitude field
- **coordinatesFieldsOrder**: a list of strings, describing the order in which the three input elements occur
- **epsgCodes**: a list of tuples, each describing an EPSG code to select and its corresponding label in the form

File upload (type file)

The file upload can be used to link files to a database column in the form. To do this, the uploaded files are stored in Mapbender and the path is noted in the column.

The storage path and the name of the stored files can not yet be changed. The file upload always saves to the same directory and is built up from the parameters:

- tablename
- columnname
- filename

The filesystem path is:

- <mapbender>/web/uploads/featureTypes/[tablename]/[columnname]/[filename].png

The linked URL stored in the database column is:

- <http://localhost/mapbender/uploads/featureTypes/{tablename}/{columnname}/{filename}.png>

```
- type: file                # Typ file for the upload of files
  title: Dateiupload       # Label (optional)
  text: Laden Sie ein Bild hoch. # Informationtext (optional)
  name: file_reference    # table column for the storage path

# Experimental parameters:
#accept: image/*           # Pre-selection of elements in the image format (window for file
↪upload opens with restriction filter)
                             # Other file-formats can be still uploaded
```

Notes: At this time, a “thumbnail” directory is created, which includes a smaller version of an image file. In future development this will be changed.

A possibility to show the uploaded images is the image-element.

Images (type image)



The image-element can be used to view a picture in the form. You can display images by specifying a URL

in a database field or URL using the `src` parameter.

Images, which are marked by the element `file` in a table column, can thus also directly be integrated and displayed.

The image can be specified by specifying the two parameters `src` and `name`.

- **src:** Url-path or file path (can be a relative path)
- **name:** Url-path or file path from the table column (can't be a relative path)
- definition of name and `src` together: The content of the database column from `name` is taken. If the column is empty, the `src` is used.

```
- type: image                                # Feature type field name image.
  name: file_reference                        # Reference to the database column. If defined,
↳ the path or URL in the field can be used and replaces "src" option
  src: "../bundles/mapbendercore/image/logo_mb3.png" # Specify a path or URL to an image. If the path
↳ is relative use relative: true.
  relative: true                            # Optional. If true, the "src" path is determined
↳ from the "/web" directory (default: false).
  enlargeImage: true                        # Image is enlarged to original size/maximum
↳ resolution by clicking on the preview image. It is not scaled to screen size.

# Experimental information about styling
imageCss:
  width: 100%                               # Image CSS Style: Scales the preview image in
↳ the form, different from the original size in percent.
```

Caution: If only `name` and not `name` and `src` are specified, the wrong image appears from the previous data entry, if the column is empty.

Dynamic paths (eg `"bundles/mapbendercore/image/[nr].png"` or `'bundles/mapbendercore/image/' + data.image_reference`) can not be specified.

One way to work around this is to create a trigger that will merge the path and contents of a table field into the database column.

Definition of the available toolsets (Toolset Type)

Toolset types:

- **drawPoint** - Draw point
- **drawLine** - Draw a line
- **drawPolygon** - Draw polygon
- **drawRectangle** - Draw rectangle
- **drawCircle** - Draw circle
- **drawEllipse** - Draw ellipse
- **drawDonut** - Draw a Donut (enclave)
- **modifyFeature** - Move vertices of a geometry
- **moveFeature** - Move geometry
- **selectFeature** - Geometry de-/select (experimental). There is no interaction with the table yet and the available workflows are limited to the Delete operation.

- **removeSelected** - delete selected geometry (experimental). Deletes all objects selected in the map.
- **removeAll** - Caution: remove all geometries from the table

YAML-Definition of toolset types

```
polygon:
  [...]
  toolset:
    - type: drawPolygon
    - type: drawRectangle
    - type: drawDonut
    - type: removeSelected
```

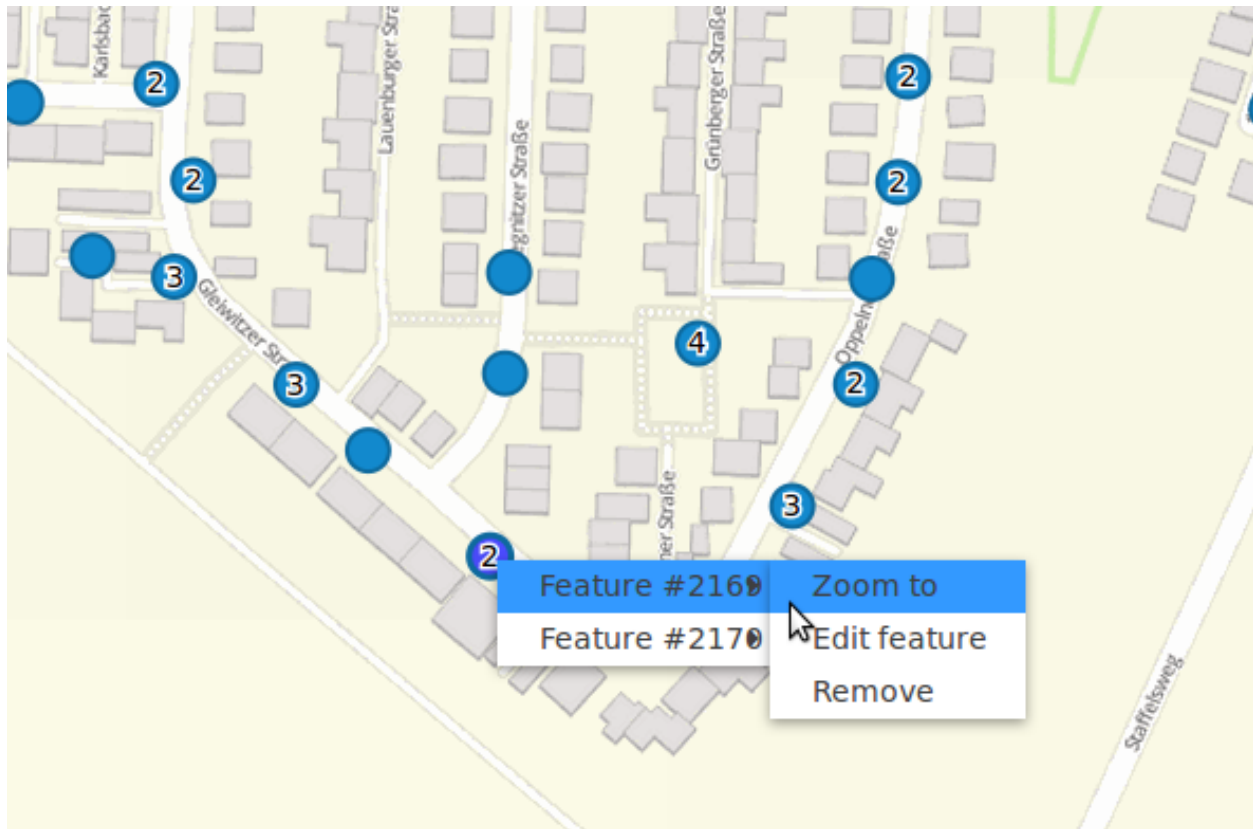
Search in the tables (inline Search)

You can use the inline search to search for an element in the table. The activated element displays a search bar above the table. It shows all the search results for records of the table.

```
poi:
  ...
  inlineSearch: true      # true: allows the search in the table (default: true).
  ...
```

Context Menu

Using the context menu, an object on the map can be considered in more detail. After the activation you can open a context menu via the right mouse click on an object or cluster.



Items of the Context Menu:

- **Zoom to:** Zoom to the map extent of the object
- **Edit features:** Edit the features of the object. Opens the Digitizer dialog.
- **Remove:** Remove the selected object.

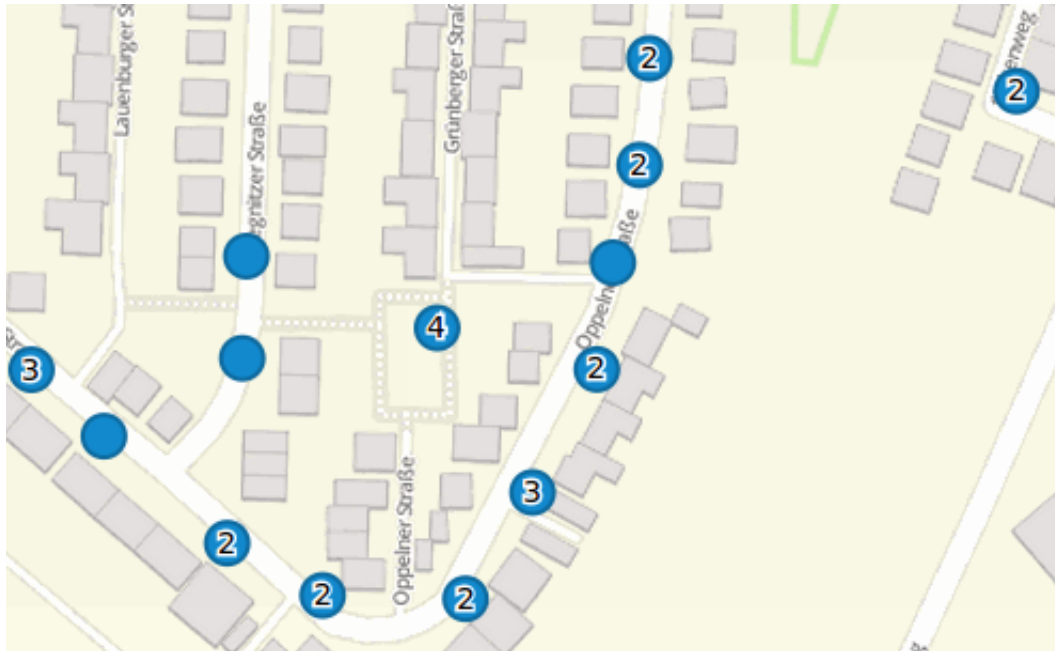
If the corresponding *basic definition* (`allowEditData`, `allowDelete`) not defined, then they are also not available in the Context Menu. In the above example the delete function is not available for the polygons.

```
poi:
...
useContextMenu: true
...
```

Clustering (experimental)

By clustering the objects can be combined on the map. Depending on the defined distance and zoom level different numbers of objects can be clustered.

Due to the complexity of the Clustering, future versions may have changes in functionality and syntax, so we define that still as experimental. Dependencies are to the display of features in the current extent/all areas and the different geometry types.



Definition of the cluster element:

- **scale:** Zoom level.
- **distance:** distance between features in m to activate the clustering.
- **disable:** zoom level to disable the clustering.


```
poi:
...
  clustering:
    -
      scale: 10000      # Zoom level
      distance: 60      # distance between features to cluster
    -
      scale: 2500
      distance: 40
    -
      scale: 1000
      distance: 20
    -
      scale: 500
      distance: 1
      disable: true      # disable clustering at defined zoomlevel
...

```

Map-Refresh after save

After saving an object, the refresh can be activated using the `refreshLayersAfterFeatureSave` option. This parameter is used to reload the defined layer instances in the map-element. This makes changes regarded to WMS services directly visible in the map. This event is thrown only if the "Save" button of the attribute dialog is pressed.

If the YAML application is used in the `/application` folder, it can be specified by a unique name or by the instance-ID. If the applications are edited using the graphical user interface in the backend with the digitizer-element, it can be specified by the instance-ID.



```

YAML - main
  Id      Titel
  -----
  17      Mapbender Users (mapbender_users)
  Typ
  WMS
  
```

```

poi:
  [...]
  allowEditData: true
  refreshLayersAfterFeatureSave: # If no entry is made in this area no map refresh is carried out
  ↪ after saving
    - 17
    - 18
    - osm      # specify by unique name only with applications in app/config/application
  [...]
  
```

Notes: Map-Refresh after save is available from Digitizer version 1.2. Check up-to-date information about digitizer versions: <https://github.com/mapbender/mapbender-digitizer>

Duplicate features

Already captured objects can be duplicated. This is done via a duplicate-button within the popup of the current selected already existing feature, via the context menu and the hit table. In order for the new object to be recognized better in the map, a color highlighting can be defined here.

The Duplicate button can be activated depending on a specific attribute value. This means that only when the corresponding attribute has a specific value (`date > 0`) the duplicate function works.

- **data:** Define default values for attributes.
- **rules:** Rule based duplicating (only if the filter/rule is active the object can be duplicated).
- **style:** Styling of the duplicated feature (more at Design and Styles)
- **on:** Events while duplicating process

```

poi:
  [...]
  copy: # If no specification is made in this area you can't duplicate objects
  enable: true
  data:
    date: 2017
  rules:
    - feature.attributes.id > 10
  style:
  
```

(continues on next page)

(continued from previous page)

```
label: "Dupliziertes Objekt"
fillColor: "#ff0000"
fillOpacity: 1
strokeWidth: 4
strokeColor: "#660033"
on:
  success: widget._openFeatureEditDialog(feature)
  error: console.error(feature)
```

Events

Different events exist that can be associated to a feature to manipulate attributes before or after an action.

- **onBeforeSave:** Event before the storage of a new/modified information
- **onAfterSave:** Event after the storage of a new/modified information
- **onBeforeUpdate:** Event before the update of a modified information
- **onAfterUpdate:** Event after the update of a modified information
- **onBeforeSearch:** Event before the search in the SearchField of the Digitizer
- **onAfterSearch:** Event after the search in the SearchField of the Digitizer
- **onBeforeRemove:** Event before deleting data
- **onAfterRemove:** Event after deleting data

In difference to the save-events, the update-events work only on an update of the data, not on creation.

Note: The events are still in development and should be used with caution. The correct matching of the events and their dependencies are not yet finished and may be changed in future versions.

The following sections show some examples. If you want to set several parameters in an event, these can be listed in sequence, separated by a semicolon, e.g.

```
events:
  onBeforeSave: $feature->setAttribute('interests', 'maps'); $feature->setAttribute('name', 'test');
```

Storage of predefined attribute data in an additional attribute-columns:

The following example shows how data can be written to an additional attribute column. This is done with the column "interests" and the fixed value "maps". When saving, the fixed value is stored in the table and you can use it via a filter for the selected display.

```
events:
  onBeforeSave: $feature->setAttribute('interests', 'maps');
```

Storage of group roles in an additional attribute-columns:

The following example shows how mapbender user data can be written to an additional attribute column. Here, this is done with the column "group" and the storage with the group roles of the user (userRoles).

```
events:
  onBeforeSave: $feature->setAttribute('group', implode(',', $userRoles));
```


Storage of attribute data in an additional attribute-columns:

This example shows how data can be stored in an additional attribute-column after saving. In this case it is done with two geometry-columns "geom" and "geom2". When saving, the data of "geom" should be saved in the field "geom2".

Depending on the use case, the `onBeforeInsert` or the `onBeforeUpdate` event can be used.

At the time of the saving-process the new geometry doesn't yet persist in the database. Therefore it cannot be accessed as a feature but only via the corresponding "item", an internal Digitizer structure. This "item" is based on the formular and the defined attribute fields.

events:

```
onBeforeInsert: $item['geom2'] = $item['geom'];
onBeforeUpdate: $item['geom2'] = $item['geom'];
```

In this event the value of "geom2" is overwritten with the value of "geom".

Storage of different geometry-types:

The above scenario can be extended to a slightly constructed example in which simultaneously different geometry types shall be saved. With the help of PostGIS, lines are interpolated to points. The Digitizer can use an event to fire the according SQL statement.

events:

```
onBeforeInsert: |
  $sql = "SELECT
  ST_Line_Interpolate_Point('".$item['geomline']."'::geometry, 1) as geom";
  $stmt = $this->getConnection()->prepare($sql);
  $stmt->execute();
  $result = $stmt->fetchAll();
  $item['geompoi'] = $result[0]['geom'];
```

The `onBeforeInsert` event is used here. The pipe symbol "|" after the event signals a following multi-line statement. This block contains PHP code, which calls SQL-statement. The SQL-statement calls the `ST_Line_Interpolate_Point` function of PostGIS and commits the digitized line. Because this line is not yet persisted in the database, you have to access it with the "item" (geomline). The next lines build up the SQL-statement and delivers it to the SQL-connection defined in the feature type. The last line writes the resulting point (geompoi) into the point-geometry-field.

Buttons

Further buttons can be defined for the popup forms. The events (by clicking on the buttons) can be freely defined with JavaScript. Thus, for example, mailto data can be generated for the integration of a mail.

poi:

```
...
  popup:
    title: polygon test suite
    width: 500p
    # resizable: true
    buttons:
      - text: message to editor
        click: |
          var body = encodeURIComponent("Sehr geehrter Herr/Frau xx,"+"\\nLink:"+location.href);
          location.href = "mailto:firstname.lastname@mail.com?subject=New edit in webgis&
          ↵body=Mail to editor for further edits.";
```

(continues on next page)

(continued from previous page)

```
- text: message to controller
  click: |
    location.href = "mailto:firstname.lastname@mail.com&subject=webgis&body=really?";
```

Design and Styles

By specifying a style the way the objects are displayed on the map can be defined.

- **default:** defines the normal display of the objects on the map
- **select:** defines the appearance of the objects while mouseover
- **selected:** defines the appearance of the objects after click event

```
poi:
...
  styles:
    default:
      graphic: true
      strokeWidth: 5
      strokeColor: "transparent"
      fillColor: '#c0c0c0'
      fillOpacity: 1
      fillWidth: 2
      # label: ${name} ${type}
      # labelOutlineColor: '#eeeeee'
      pointRadius: 10
    select:
      strokeWidth: 1
      strokeColor: "#0e6a9e"
      fillOpacity: 0.7
      fillColor: "#0e6a9e"
      label: ${name} ${type}
      pointRadius: 10
    selected:
      strokeWidth: 4
      strokeColor: "#648296"
      fillOpacity: 1
      fillColor: "#eeeeee"
      label: ${name} ${type}
      pointRadius: 10
...

```

- **strokeColor:** Color of the border line [color value/transparent]
- **strokeWidth:** Width of the border line [numeric]
- **strokeOpacity:** Transparency of the border line [0-1]
- **fillColor:** Color of the filling [color value/transparent]
- **fillWidth:** Width of the filling [numeric]
- **fillOpacity:** Transparency of the filling [0-1]
- **pointRadius:** Radius around the center [numeric]
- **label:** Labeling the object with fixed values and / or DB fields, e.g. "ID \${nummmer}"

- **labelOutlineColor:** Color of the border from the label [color value/transparent]

YAML-Definition for the element Digitizer in mapbender.yml

This template can be used to insert the element into a YAML application.

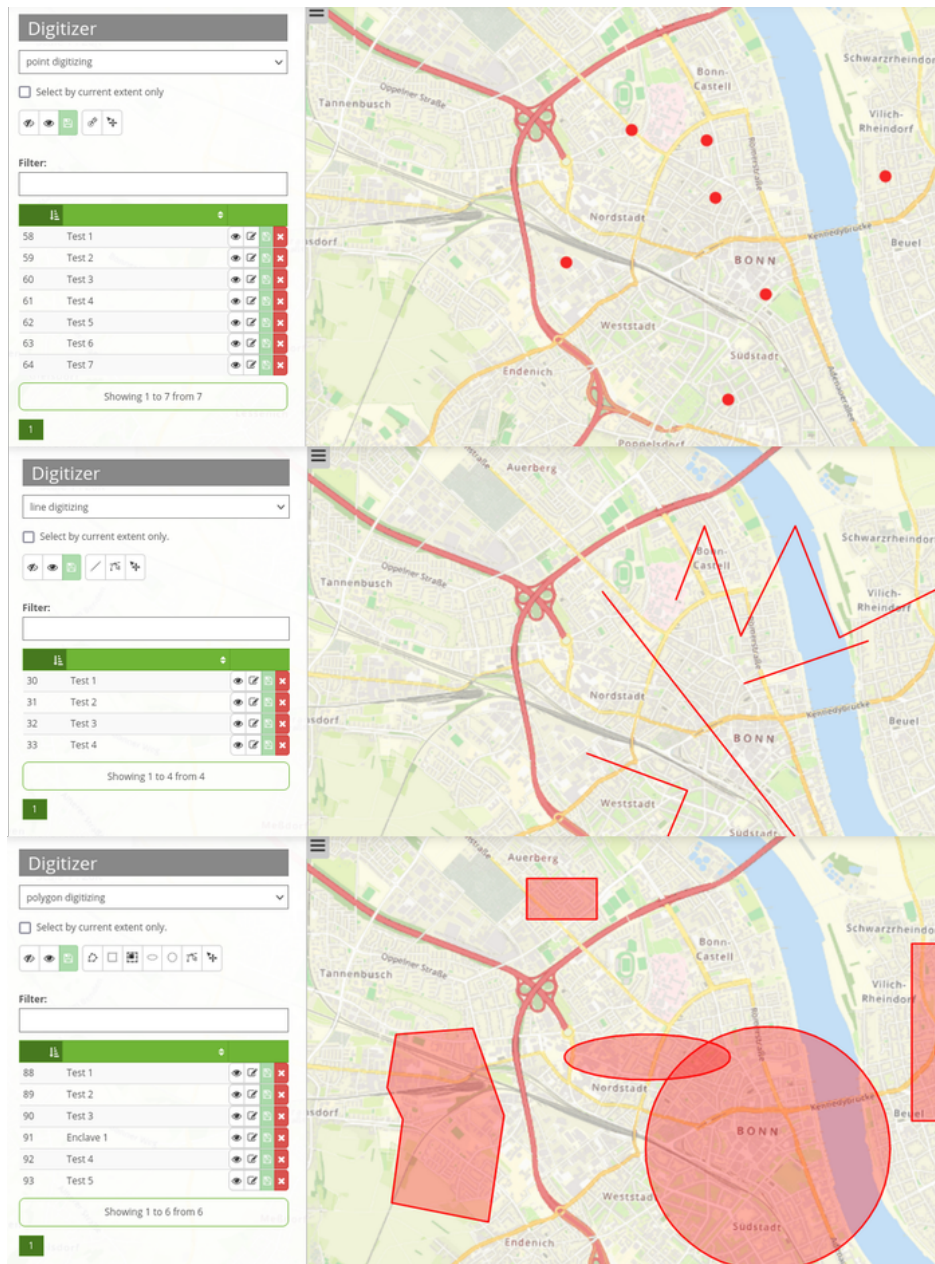
```
sidepane:
  digitizer:
    class: Mapbender\DigitizerBundle\Element\Digitizer
    title: Digitalisation
    target: map
    schemes:
      ...
```

More information about the functionality of the digitizer can be found [here](#):

Functionalities

Digitizer allows the editing of FeatureTypes. These are based on points, lines or polygon geometries and their attribute data. The attribute data is displayed in the configurable form of the Digitizer element. Geometries can be edited directly via the map. Digitizer offers complex editing functionalities:

- Draw points, lines, polygons, rectangles, ellipses or circles
- Move objects
- Add vertices (to lines or polygons)
- Generate areas with enclaves



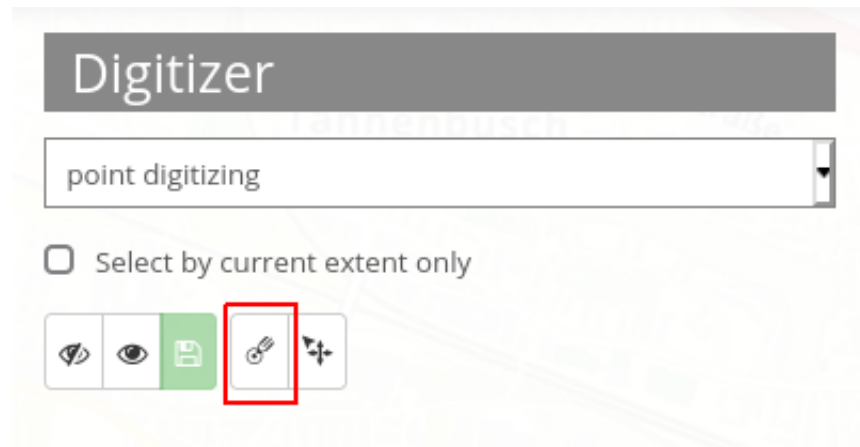
The following sections explain the different functionalities of Digitizer in accordance with its default configuration.

Draw geometries

When drawing geometries, one can choose between the geometry types “point”, “line” or “polygon” via a dropdown menu.

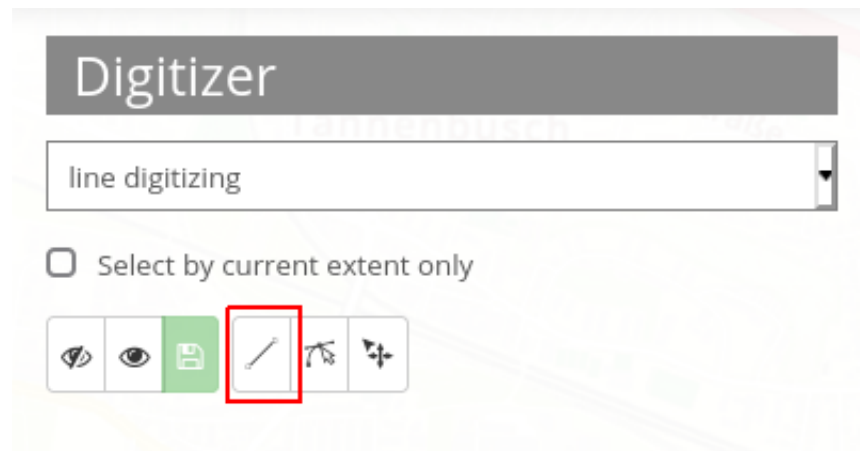
Points

A click on the button “draw point” activates/deactivates the option to draw points.



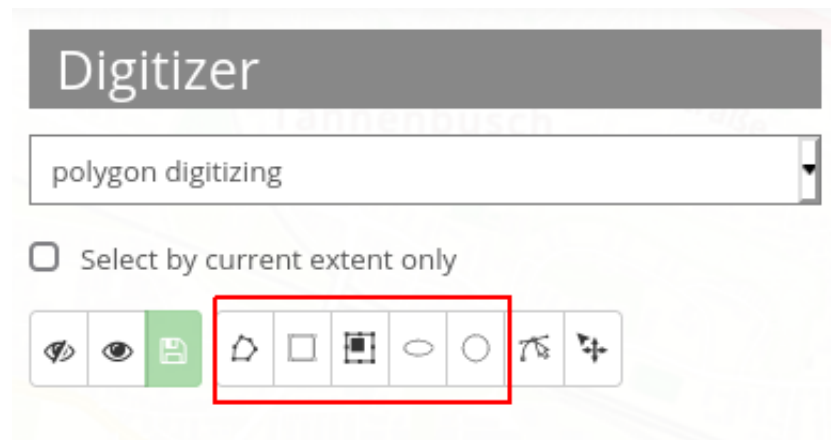
Lines

A click on the button “draw line” activates/deactivates the option to draw lines.



Polygons

Polygons, rectangles, enclaves, ellipses and circles can be drawn after activation of the corresponding button.



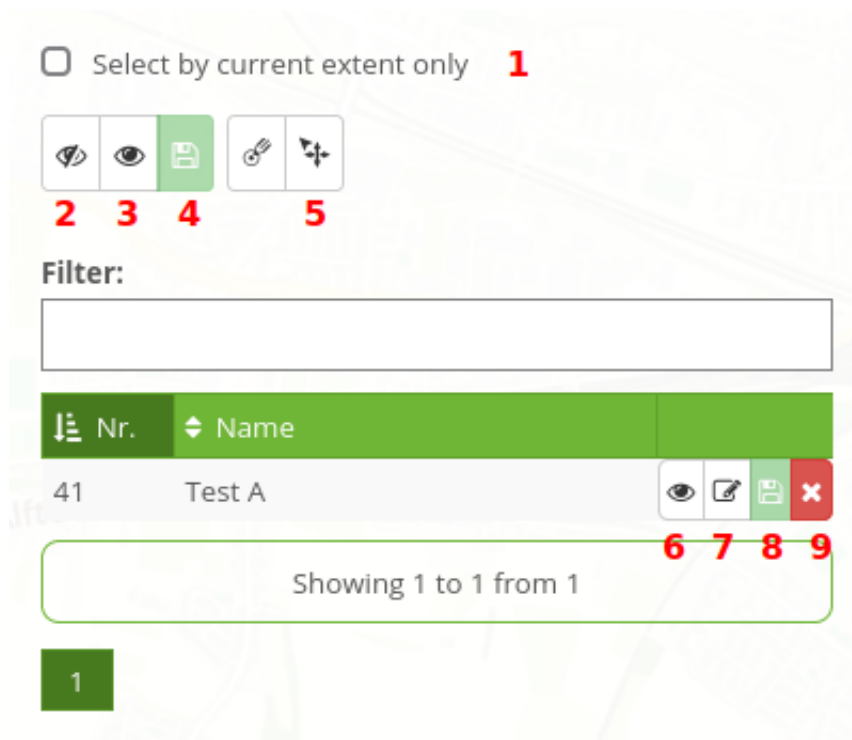
After activation, an object can be created through one or more clicks on the map. Afterwards, a pop-up window will appear, which requests attribute data in correspondence to its configured YAML-configuration.

Edit, save or delete geometries

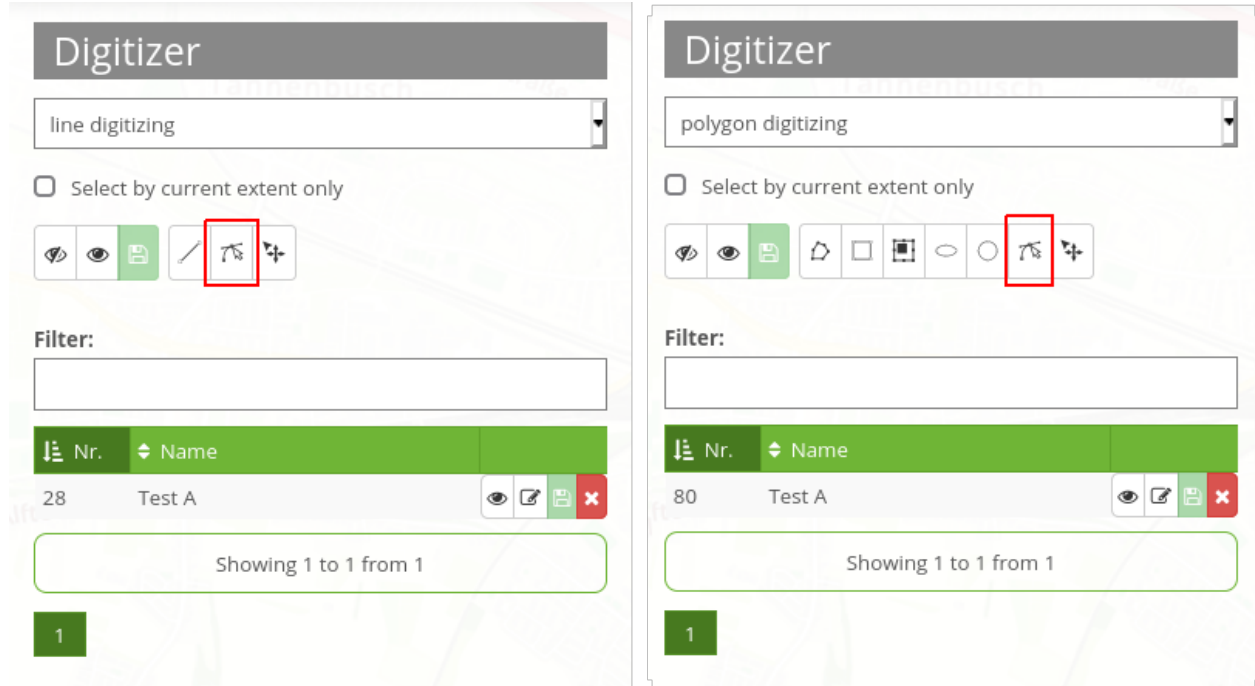
Objects are saved in the defined database table. The objects are displayed on the map and additionally as a table in the sidepane. This simplifies the administration of data. You can define which columns to list via the configuration. In this configuration the number (ID is automatically created after saving) and name of each object are displayed. It is possible to sort the columns or to search in the columns.

The list can also only display geometries of the current extent **1**. It is possible to hide **2** or show **3** all objects via click on the corresponding buttons. Modifications can be saved for all objects **4**. A change would, for instance, be the movement **5** of an object. After the button has been activated, geometries can be moved on the map with the mouse.

Furthermore, individual objects can be hidden **6** and changes saved **8**. You can change the attribute data **7**. Every object can be deleted **9**.



Previously described functions are identical for all geometries. Additionally, intermediate points can be added to lines and the corners of areas modified. This option is activated via click on the Button “Edit”. In order to modify an object, it has to be selected beforehand.



3.2.2 Data Manager

The element Data Manager is similar to the *Digitizer*. However, Data Manager only works with nonspatial data, i. e. you can not create geometries.

Data Manager can be used to maintain reference tables for example a table with contact information.

Setup

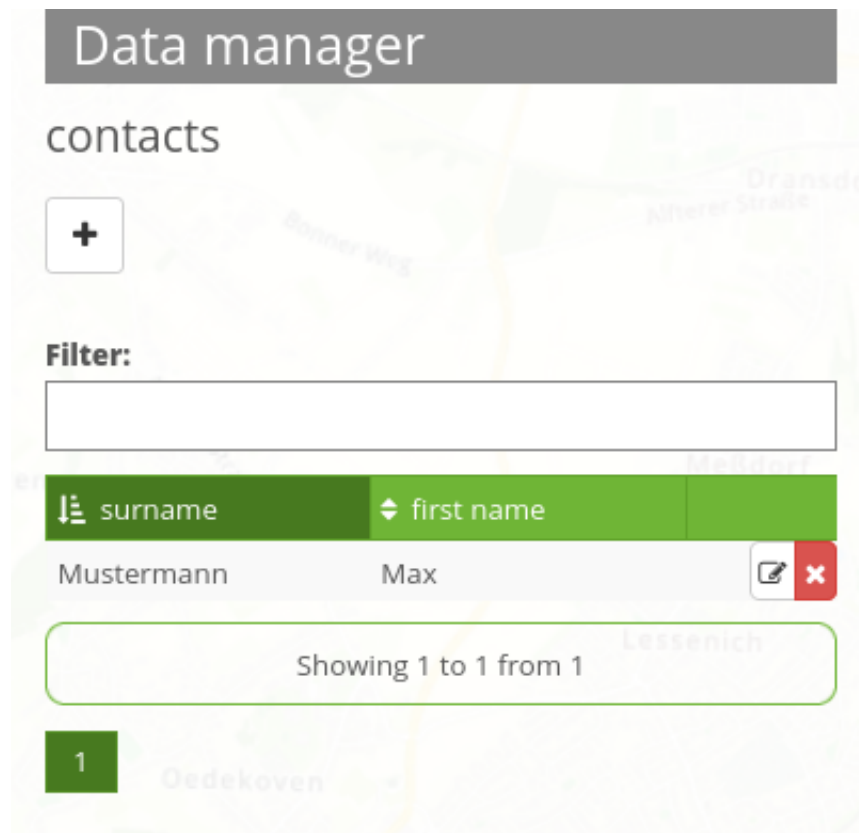
The Data Manager needs access to a database where the editable tables are. You have to define a new database configuration to be able to connect with. Read more about this under *YAML Configuration (Configuration and Application files)*.

Configuration example

The definition of the Data Manager is done in YAML syntax in the textarea configuration at schemes. Here you define the database connection, the editable table, the attribute form.

Hint: If errors occur in the database, fields or form, various error messages appear. The productive call and app.php will give a general error message. If you want to see the exact error message, you should call the page via app_dev.php. In this case, detailed error messages about the error behavior will appear.

Data Manager is a good solution to store simple contact information in Mapbender:



In the example, an input template is opened after clicking the + -button.

contacts ✕

Example for element Data Manager

last name

Oppenheimer

first name

Gonzo

street

Main Road

house number

5a

postcode

12345


location


Nowhere

date_favorite


01 . 02 . 2022 ✕

color favorite

#303cd7 

phone number 

+49 123 456

email 

g.oppenheimer@test.de

notes

Very good person

☐ publish contact

Information

user: root Datum:2022-01-31

Choose one

☐ Option 1

☒ Option 2

☐ Option 3

Select at least one (multiple choice)

Option 1

Option 2 (disabled)

Option 3

Option 4

Save

Delete

Cancel

SQL for the contact table

The following SQL commands must be executed in your database to create the contact table for the example.

```
CREATE TABLE public.contacts (  
    gid serial PRIMARY KEY,  
    firstname varchar,  
    lastname varchar,  
    street varchar,  
    housenumber varchar,  
    postcode integer,  
    location varchar,  
    phone varchar,  
    email varchar,  
    notes varchar,  
    public boolean,  
    date_of_creation date DEFAULT date(now()),  
    user_of_last_edit varchar,  
    date_favorite date,  
    color_favorite varchar  
);
```

Element Konfiguration

The element can be integrated in the sidepane.

Edit element

Title

Schemes

```

1 contacts:
2   datastore:
3     connection: geodata_db
4     table: contacts
5     uniqueId: gid
6     events:
7       onBeforeSave: '$feature->setAttribute(''user_of_last_edit'', $user ); '
8   allowEdit: true
9   allowCreate: true
10  allowDelete: true
11  allowRefresh: false
12  popup:
13    title: contacts
14    width: 550px
15  formItems:
16    -
17      type: form
18      children:
19        -
20          type: label
21          title: 'Example for element Data Manager'
22        -
23          type: fieldSet
24          children:
25            -
26              type: input
27              title: 'last name'
28              attr:
29                placeholder: 'Please enter your last name.'
30                name: lastname
31              css:

```

YAML-Definition

You can find the necessary YAML-Code of the configuration example below:

```

contacts:
  datastore:
    connection: geodata_db
    table: contacts
    uniqueId: gid
    events:
      onBeforeSave: '$feature->setAttribute(''user_of_last_edit'', $user ); '
  allowEdit: true
  allowCreate: true
  allowDelete: true

```

(continues on next page)

(continued from previous page)

```

allowRefresh: false
popup:
  title: contacts
  width: 550px
formItems:
-
  type: form
  children:
  -
    type: label
    title: 'Example for element Data Manager'
  -
    type: fieldSet
    children:
    -
      type: input
      title: last name
      attr:
        placeholder: 'Please enter your last name.'
      name: lastname
      css:
        width: 50%
    -
      type: input
      title: 'first name'
      name: firstname
      css:
        width: 50%
  -
    type: breakline
  -
    type: fieldSet
    children:
    -
      type: input
      title: street
      name: street
      css:
        width: 30%
    -
      type: input
      title: 'house number'
      name: housenumber
      css:
        width: 20%
    -
      type: input
      title: postcode
      name: postcode
      css:
        width: 20%
    -
      type: input
      title: location
      name: location
      css:

```

(continues on next page)

(continued from previous page)

```

        width: 30%

-
  type: fieldSet
  children:
    -
      type: date
      title: date_favorite
      name: date_favorite
      css:
        width: 50%
    -
      type: colorPicker
      title: color favorite
      name: color_favorite
      attr:
        placeholder: '#ff0000'
      css:
        width: 50%
  #-
  # type: breakLine
-
  type: fieldSet
  children:
    -
      type: input
      title: phone number
      name: phone
      mandatory: true
      mandatoryText: 'Please add your phone number.'
      infoText: 'Help: Please add your phone number.'
      copyClipboard: true
      css:
        width: 50%
    -
      type: input
      title: email
      infoText: 'Help: Please add your e-mail.'
      copyClipboard: true
      placeholder: 'Please enter your e-mail.'
      name: email
      css:
        width: 50%
  -
  type: breakLine
-
  type: textArea
  title: notes
  placeholder: 'You can leave notes here.'
  name: notes
  value: 'Guter Tester'
-
  type: checkbox
  name: public
  value: true
  title: 'publish contact'

```

(continues on next page)

(continued from previous page)

```

-
  type: text
  title: 'Information'
  text: "'user: ' + data.user_of_last_edit + ' Datum: ' + data.date_of_creation\n"
- type: radioGroup
  title: Choose one
  name: choice_column_1
  options:
    - label: Option 1
      value: v1
    - label: Option 2
      value: v2
    - label: Option 3
      value: v3
  value: v2 # Pre-select second option by default for new items
- type: select
  title: Select at least one (multiple choice)
  attr:
    required: required
    multiple: multiple
  name: choice_column_2
  options:
    - label: Option 1
      value: v1
    - label: Option 2 (disabled)
      value: v2
      attr:
        disabled: disabled
    - label: Option 3
      value: v3
    - label: Option 4
      value: v4
  value: v1,v3 # use comma-separated values for default multi-select value
table:
  autoWidth: false
  columns:
    -
      data: lastname
      title: last name
    -
      data: firstname
      title: first name
  info: true
  lenghtChange: false
  ordering: true
  pageLength: 10
  paging: true
  processing: true
  searching: true

```

3.2.3 Sketches

The Sketches element allows creating, editing and deleting temporary objects. Sketches are not saved within a database and will be lost after the browser window is refreshed or closed. Sketch geometries will be exported into *Image Export* and *PrintClient*.

The following geometry types can be sketched:

- Point
- Line
- Polygon
- Rectangle
- Circle
- Text

Configuration

Add element

Sidepane - Sketches

☐ Auto activate

☒ Deactivate on close

Title

Sketches

Geometrytypes *

Point

Line

Polygon

Rectangle

Colors

#ff3333,#3333ff,#44ee44

☒ Allow custom color

Back

Add

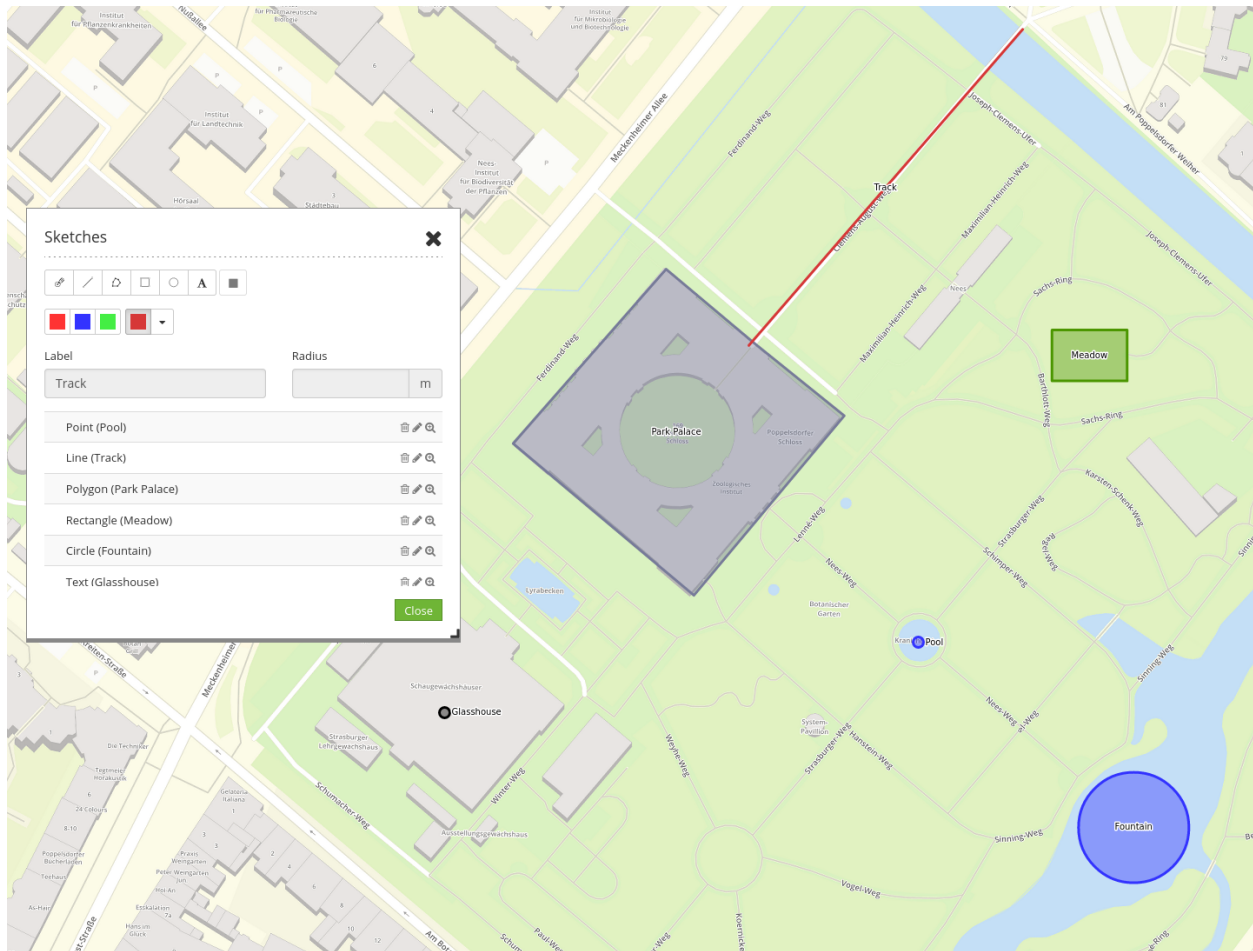
Cancel

- **Auto activate:** Opens the Sketch element on application start.
- **Deactivate on close:** Sketches are deactivated after closure of the element.
- **Title:** Title of the element. This is referenced by the button which opens the element.
- **Geometrytypes:** Lists all geometrytypes (mandatory field).
- **Colors:** Creates buttons for custom colors in the Sketch tool. Color values must be comma separated. Needs at least one input color (mandatory field).
- **Allow custom color:** Creates a custom color picker in the Sketch tool. Disable to show only the defined CSS colors.

Draw geometries

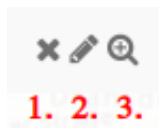
- **Point drawing:** is activated with a click on the “Point” button. Points are set with a simple click on the map and will appear in the Sketches table.
- **Line drawing:** is activated with a click on the “Line” button. Line segments are set via a single click on the map. To finish drawing, double click.
- **Polygon drawing:** is activated with a click on the “Polygon” button. Vertices are set via a single click. To finish drawing, double click anywhere in the map or single click on the last drawn vertex.
- **Rectangle drawing:** is activated with a click on the “Rectangle” button. Rectangles are created by two single clicks.
- **Circle drawing:** is activated with a click on the “Circle” button. For a circle, two single clicks are sufficient: One for the position of the circle center and one for the circumference. After the circle was created you can edit it and set a radius.
- **Text labels:** are activated with a click on the “Text” button. A label has to be defined in the label field before a text is set.

See the image below for sketch examples:



Delete, Edit and Zoom

Drawn sketches can be individually deleted [1] or edited [2]. There is also the possibility to zoom to a particular geometry [3].

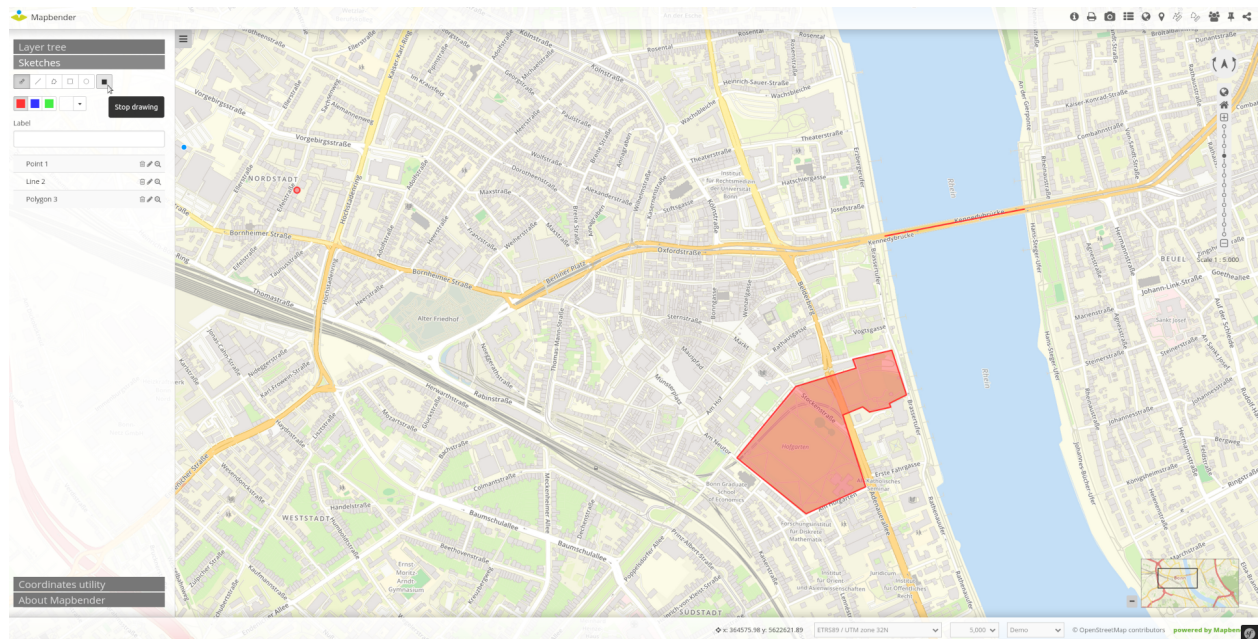


Color picker

It's possible to pick colors after setting up the configuration parameters (see above). If done correctly, buttons offering the colors will appear. Clicking on one of the color buttons determines the color for the next feature drawn, or, if currently editing a feature (click on "Edit geometry" icon in table listing), immediately updates that feature to that color. This will work with the geometry that isn't saved to the table yet, too. The color picker that can be activated with **Allow custom color** in the configuration menu allows a free choice of color at the application level. If the element is configured with only a single color and the frontend color picker is disabled, the color region will not be shown and the default color will be used for all geometries instead.

Stop drawing

To interrupt a drawing, click the stop button: Note that this deletes your current draft.



YAML-Definition:

This template can be used to insert the element into a YAML application.

```

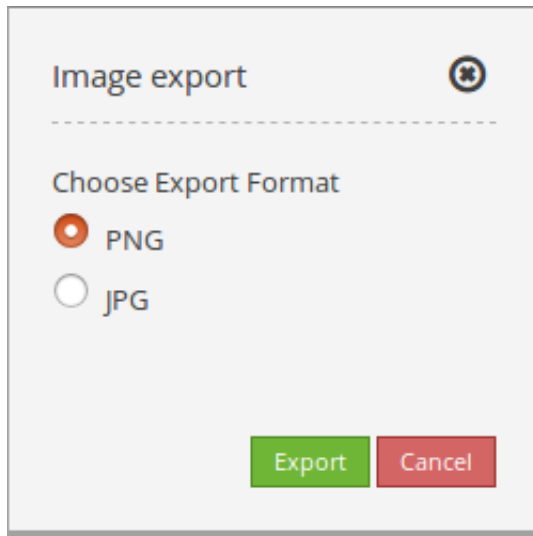
tooltip: 'Sketch'           # text of the tooltip Tooltips
target: ~                 # ID of the element
auto_activate: false      # (default: false)
deactivate_on_close: true  # Drawn sketches will be deactivated if the element is closed
                                ↪ (default: true)
geometrytypes: point,line,polygon,rectangle,circle,text  # list of available geometry types
colors:                  # Mentioned Hex color codes will be implemented into the tool
  - '#ff3333'                # Default color red
  - '#3333ff'                # Default color blue
  - '#44ee44'                # Default color green
allow_custom_color: true  # Implements a custom color picker for user color selection (default:
                                ↪ true)

```

3.3 Export / Print

3.3.1 Image Export

Image Export allows you to export your current map view. You can choose .png or .jpg format.



Configuration

Edit element

Title

Image export

Save

Cancel

- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. The title will be displayed besides the button if “Show label” is activated.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
class: Mapbender\PrintBundle\Element\ImageExport
target: map # Id of Map element to query (e.g. map)
```

You need a button to show this element. See [Button](#) for inherited configuration options.

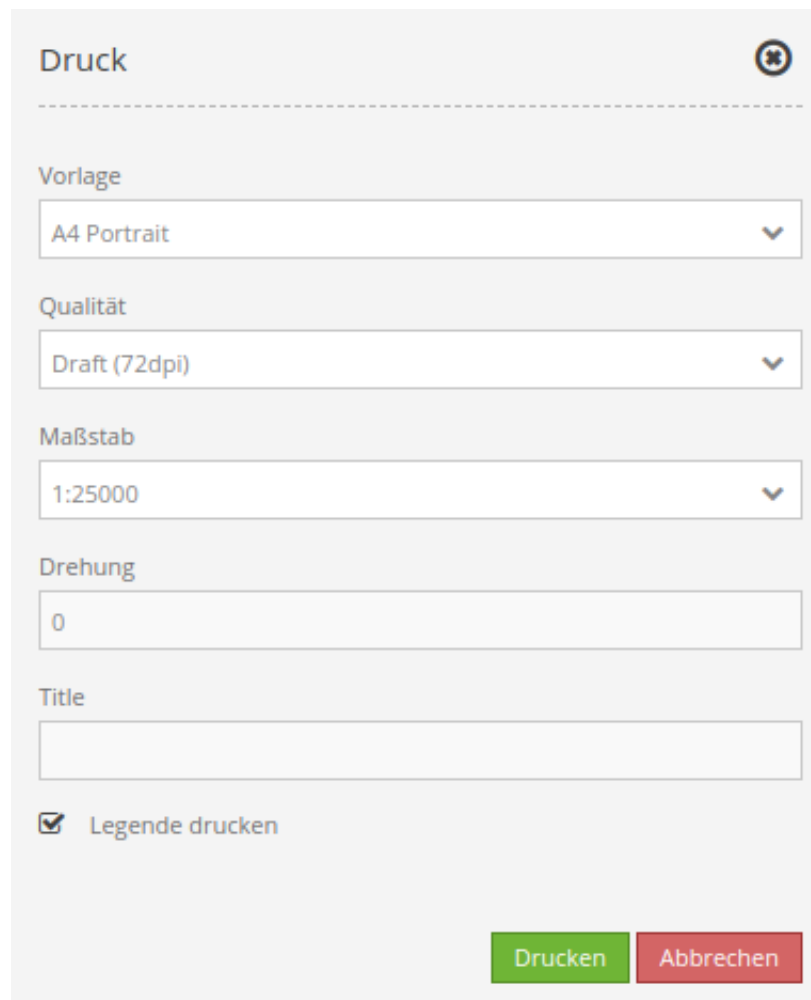
3.3.2 PrintClient

The PrintClient allows for the print of a predefined map area. The following documentation will first give an overview about the general set-up and configuration of the PrintClient. The second part will answer the question how individual print templates can be generated. Lastly, the print process and all its configuration options will be presented.

General

It is possible to define different properties of the PDF you would like to print with the client:

- Print template,
- Quality,
- Scale,
- Frame rotation,
- Activate/deactivate legend
- Optionally, it is possible to define individual input fields (e.g. title, comment, remark), which will then also be printed in the PDF.

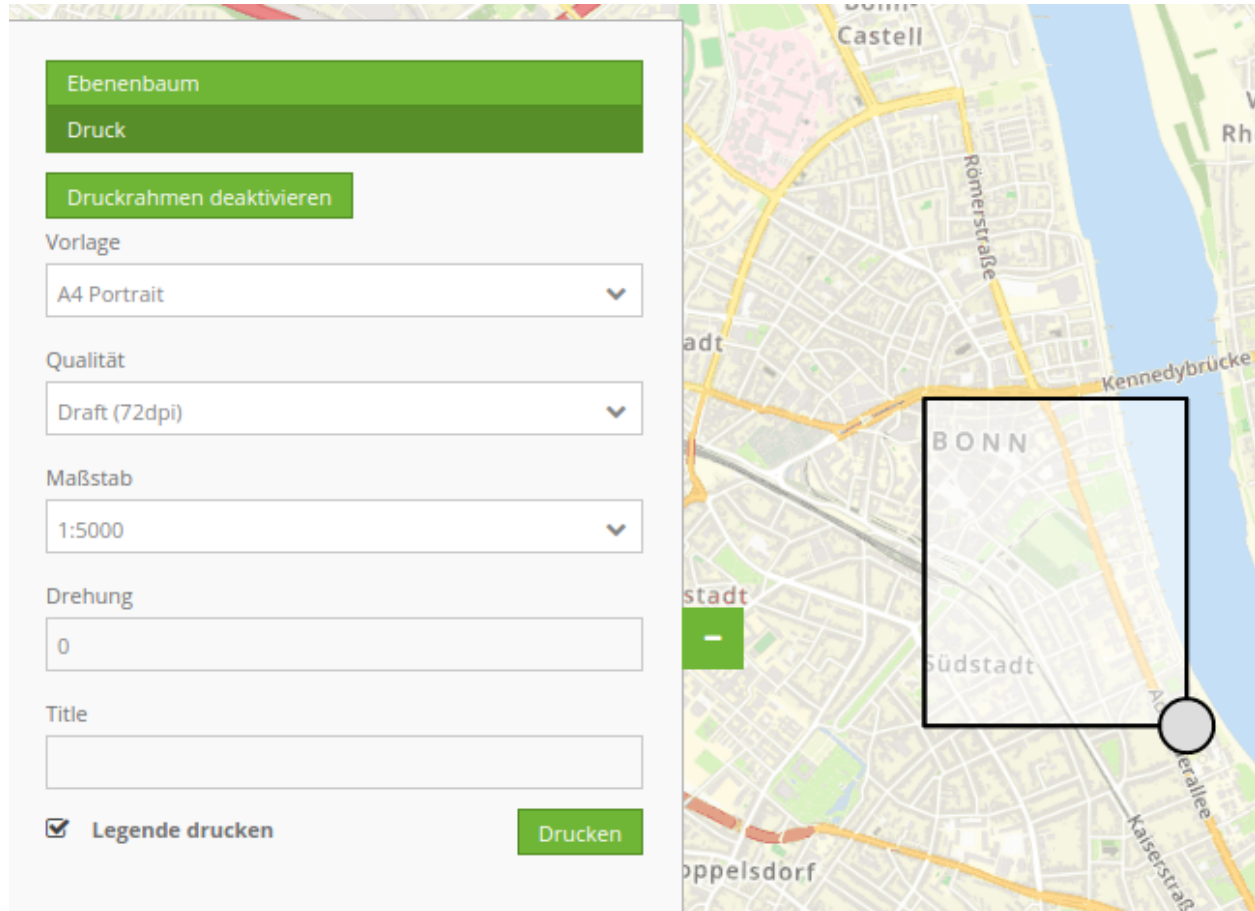


The screenshot shows the 'Druck' (Print) interface of the Mapbender PrintClient. It features a title bar with the word 'Druck' and a close button. Below the title bar, there are several configuration options, each with a label and a corresponding input field or dropdown menu:

- Vorlage** (Template): A dropdown menu showing 'A4 Portrait'.
- Qualität** (Quality): A dropdown menu showing 'Draft (72dpi)'.
- Maßstab** (Scale): A dropdown menu showing '1:25000'.
- Drehung** (Rotation): A text input field showing '0'.
- Title**: A text input field.
- Legende drucken** (Print legend): A checkbox that is checked.

At the bottom of the form, there are two buttons: 'Drucken' (Print) in green and 'Abbrechen' (Cancel) in red.

The PrintClient element can be implemented both as a dialog (via a button) and as element as part of the sidepane. If it is part of the sidepane, you have to activate the print frame first to start the print. The print frame can be moved around freely in the map canvas, it defines the area of the PDF output. When finished, you have to deactivate the print frame to use the map as usual again (when used as a dialog this happens automatically by opening and closing the dialog window).



Configuration



The Printclient can be configured in the backend. It relies on print templates (Format A0 to A4). These LibreOffice Draw files can be individually modified regarding the location of date, scale, north arrow, overview map as well as dynamic images/texts in the PDF.

Edit element

Title	<input type="text" value="Print client"/>		
Scales	<input type="text" value="500,1000,5000,10000,25000"/>		
<input checked="" type="checkbox"/> Rotatable			
<input checked="" type="checkbox"/> Print legend			
<input type="checkbox"/> Legend checkbox active			
File prefix	<input type="text" value="mapbender"/>		
Quality levels			
Dpi	<input type="text" value="72"/>	Label	<input type="text" value="Draft (72dpi)"/>
Dpi	<input type="text" value="288"/>	Label	<input type="text" value="Document (288dpi)"/>

Templates *

+

Template	<input type="text" value="a4portrait"/>	
Label	<input type="text" value="A4 Portrait"/>	
Template	<input type="text" value="a4landscape"/>	
Label	<input type="text" value="A4 Landscape"/>	

- **title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. The title will be displayed next to the button if “Show label” is activated.

- **scales:** define scales to choose from selectbox, or – if empty – a free scale can be defined in a text field.
- **rotatable:** defines if the print extent is rotatable, default is true
- **print legend:** adds a checkbox which allows your print to have a legend
- **legend checkbox checked:** if true, the legend checkbox is checked by default
- **file_prefix:** defines a file prefix for the PDF that is generated when creating a PDF (file_prefix_date.pdf will be created)
- **quality_levels:** defines quality levels in DPI values.
- **templates:** template name. Add template file name without file extensions (Mapbender is looking for two files, e.g. a4portrait.odg & a4portrait.pdf), Template files are located at app/Resources/MapbenderPrintBundle. Deleting or adding templates is possible with the “+”- or “x” buttons.
- **label:** defines a template label for the selectbox in the frontend.
- **optional_fields:**

With the configuration of the following values it is possible to enable optional fields in the print dialog. An example (title, two comment fields, name) is offered in the YAML definition.

- **title:** name of the optional field, default value is null (no optional fields are defined).
- **label:** Label of the optional field.
- **options:** { required: true } : Type of the optional field. Has to be true or false.
- **Display required fields first:** If this checkbox is active, your defined required fields appear utmost.
- **replace_pattern** - You can modify the map request for printing. You can add additional parameters like map_resolution (for MapServer).

Here’s an example for the backend configuration (or look below in the YAML definition for an enhanced and commented example).

Optional fields

title:

label: Title

options:

required: true

☐ Display required fields first

Replace pattern

default:

288: '&map_resolution=288'

pattern: mapbender_user

replacement:

288: mapbender_user 2

Save Cancel

YAML-Definition

This template can be used to insert the element into a YAML application.

```
target: map                # Id of Map element to query
type: dialog               # element or dialog, default is dialog
templates:
  - { template: a4portrait, label: A4 Portrait} # template name, template file name without file
  ↪extension (Mapbender is looking for file a4portrait.odg an a4portrait.pdf), Template files are
  ↪located at app/Resources/MapbenderPrintBundle
  - { template: a4landscape, label: A4 Landscape} # template label in the dialog
scales: [5000, 10000, 25000] # define scales to choose from select-box or if empty free scale can be
  ↪defined in a text-field
quality_levels:            # define quality levels in dpi
  - { dpi: 72, label: Draft (72dpi)} # 72 - dpi value, Draft - label
  - { dpi: 288, label: Document (288dpi)} # 288 - dpi value, Document - label
rotatable: true           # true/false use true for rotation (default: true)
legend: true              # true/false to print a legend (default: false)
legend_default_behaviour: false # true/false, if true the legend checkbox is checked by default
file_prefix: mapbender    # define the file prefix for the PDF that is generated (file_prefix_date.
  ↪pdf will be created)
optional_fields:          # define optional fields (example title-field)
```

(continues on next page)

(continued from previous page)

```

    title:                                     # name of the optional fields, default is null (no optional fields are
↪defined)
    label: Title                             # label of the optional field
    type: text                               # type of the optional field
    options:                                 #
        required: false                     # true/false
    comment1:
        label: Comment 1
        options: { required: false }
    comment2:
        label: Comment 2
        options: { required: false }
    editor:
        label: Editor
        options: { required: true }
replace_pattern:                             # You can modify the map-request for printing
    -                                       # you can add additional parameters like map_resolution (for MapServer)
        default: { 288: '&map_resolution=288' }
    -
        pattern: 'stadtplan.xml'           # or you can request a different service which is optimized
↪for printing
    replacement: { 288: 'stadtplan_4.xml' }

```

Directories

The northarrow

- The “north arrow” image is located at **app/Resources/MapbenderPrintBundle/images/**.
- The “north arrow” image can be replaced to use a different image instead.

The print templates

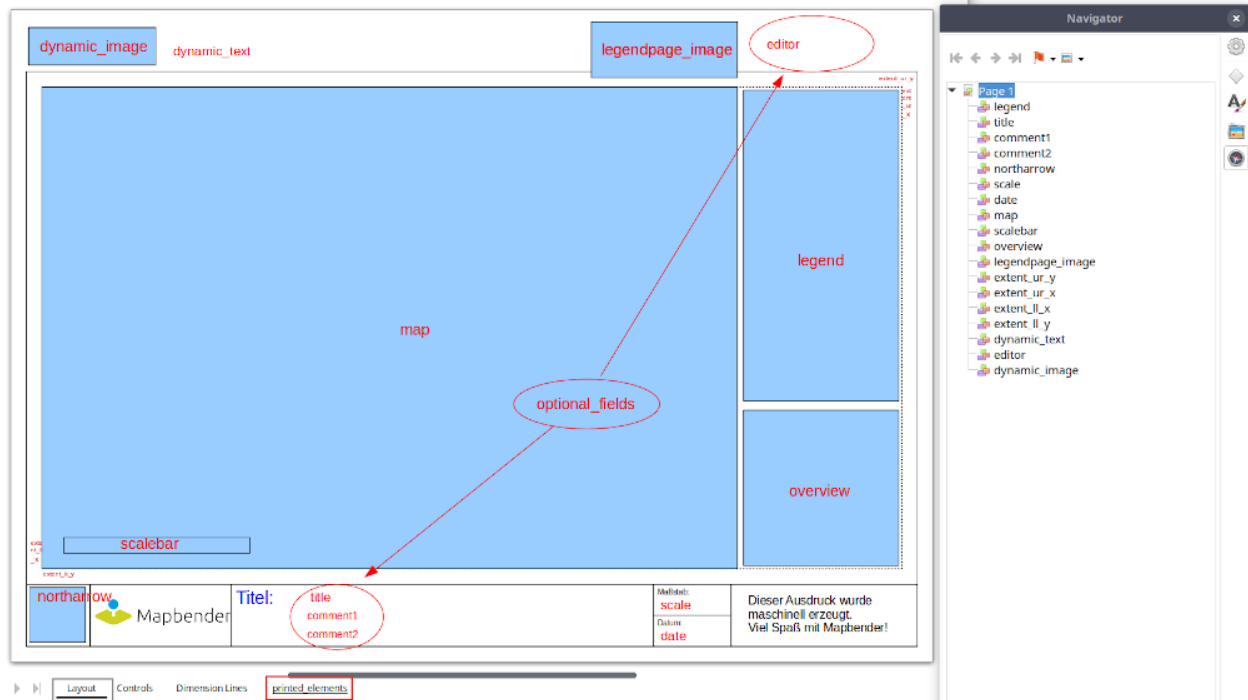
- The print templates can be found under **app/Resources/MapbenderPrintBundle/templates/**.

The print pdf output

- Mapbender saves its generated print files in the browser’s default download folder or shows them directly in your browser (depending on your browser settings).
- If the queued print is used, the files will be saved under the Mapbender directory **web/prints/**.

Create your individual templates

To create an individual print template, already existing print templates (.odg-file) can be copied, or a new LibreOffice Draw file can be used. The template can include fixed objects like a logo, copyright or print information. In addition, a layer for dynamic elements like map, overview, north arrow, scale, date and optional fields can be created. The dynamic layer is an additional non-printable layer in the LibreOffice Draw file. The layer can be added with **Menu: -> Add -> Layer -> define name for the layer and choose checkbox option “not printable”**. An element is assigned to a layer via drag and drop.

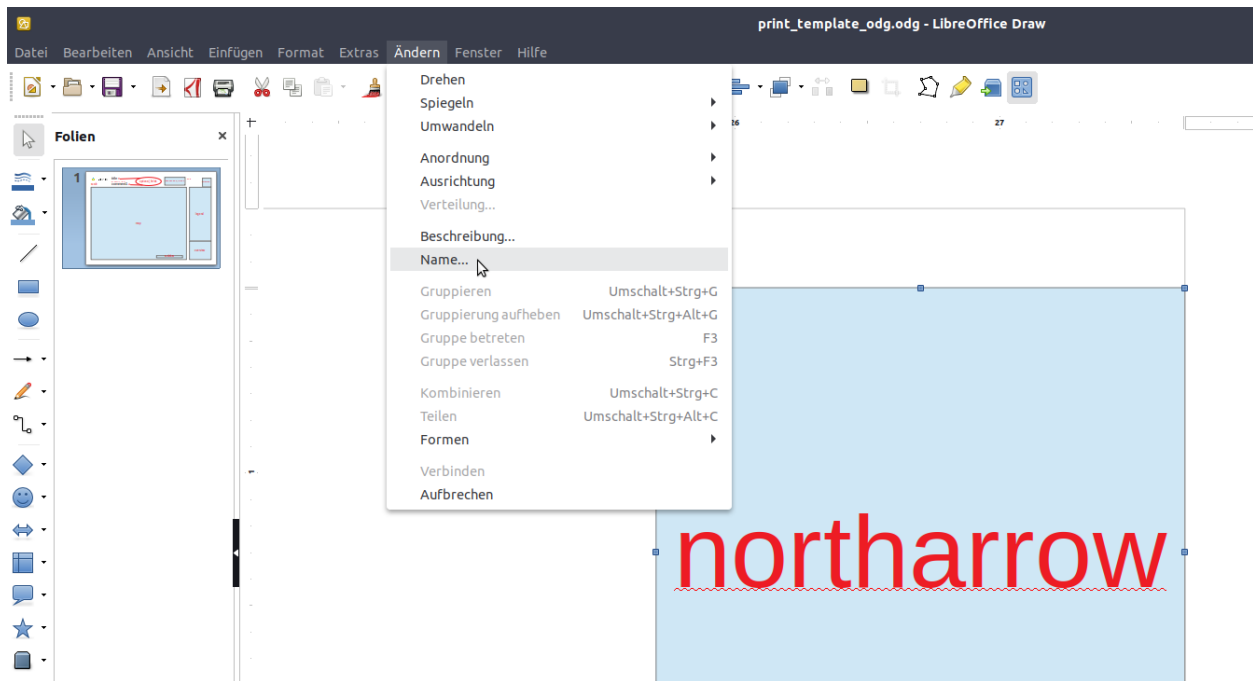


The following objects are available by default in Mapbender:

- map
- overview map
- scale
- scalebar
- date
- northarrow
- extent_ur_y, extent_ur_x, extent_ll_x, extent_ll_y (coordinates of the print extent)
- dynamic_image (connected to group role)
- dynamic_text (connected to group role)

Optional fields in the element definition (e.g. title, comment, editor) can be defined and added to the Open Office Draw file. They need to be inserted into the non-printable area of the file, or else they will appear in the generated template PDF.

When creating dynamic texts, each property must have the appropriate name, for example the field **comment1** must have the name **comment1** in the Open Office Draw file. For naming fields, it is not enough to apply the name to the content of the text field. Instead, the field needs to be selected and its object name edited. Select text field **Menu Edit** **name...** to change the name accordingly.



Export the template to .pdf under the same name as the .odg file. Use the name without its extension in the print yml-definition.

The print script will read the information (position, size, font size, alignment) from the .odg-file and combines those with the fixed objects in the PDF template and the map image in Mapbender to generate your PDF.

Printing elements in front of the map element

In order for the map element to be as large as possible and to avoid white or empty areas, elements can be placed in front of the map image to prevent that space is lost through white areas. This is particularly useful in the case of large printing formats which have a comparatively wide border.

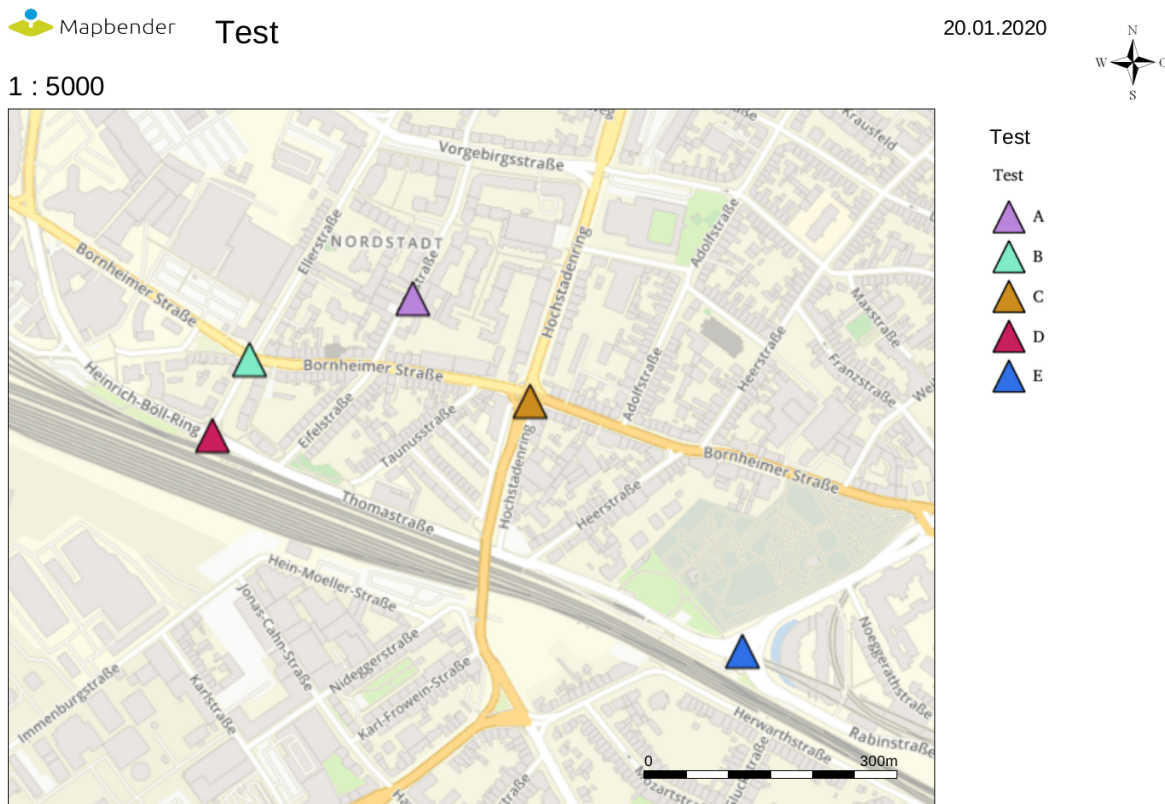
To use this function, the templates have to be adapted and transparent PDF templates have to be created.

Adapt templates:

- **Reorder elements in front of white background**
 - Arrange elements into the foreground (right-click Arrange → To the front)
 - Arrange map element into the background (right-click Arrange → To the back)
- **Select all**
 - Press CTRL + A
- **Print selection as PDF**
 - Export as PDF
 - Selection instead of All

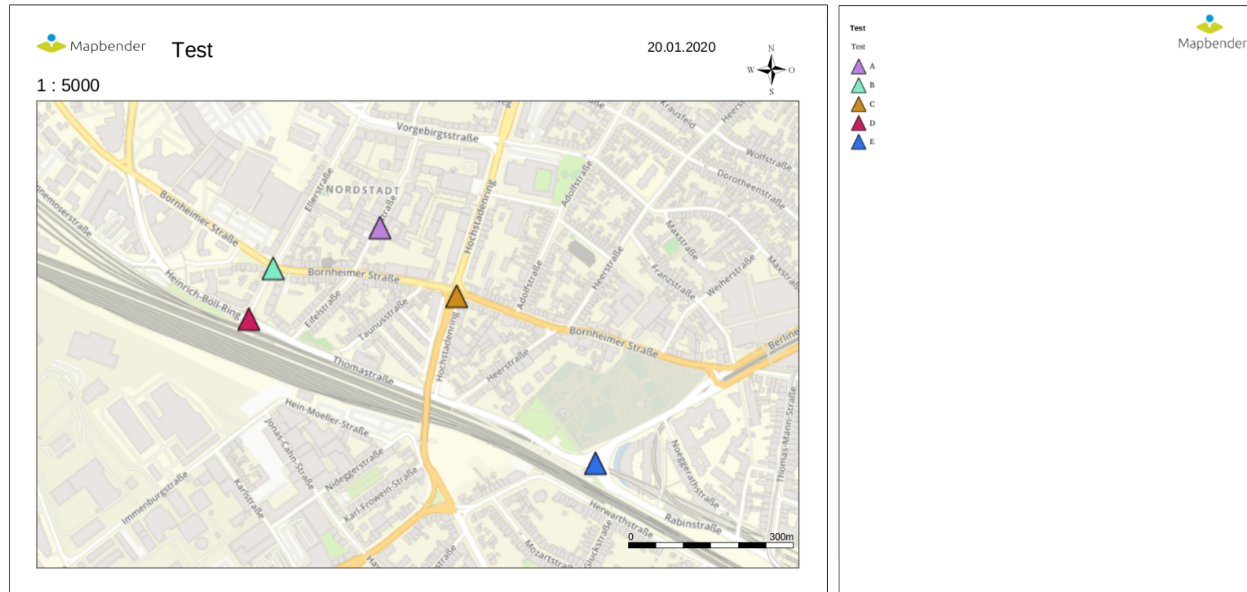
Legend on the first page

The legend can be integrated next to the map on the first page. This field is not included in the print template by default. To insert the legend the OGD print template file needs to be modified. A new dynamic field with the name "legend" on the non printable layer needs to be inserted. **Menu: Modify -> Name...** to change the name of the field to "legend". As final step, the OGD-file has to be exported as PDF-file as described above and saved in the same directory. The result could look like this:



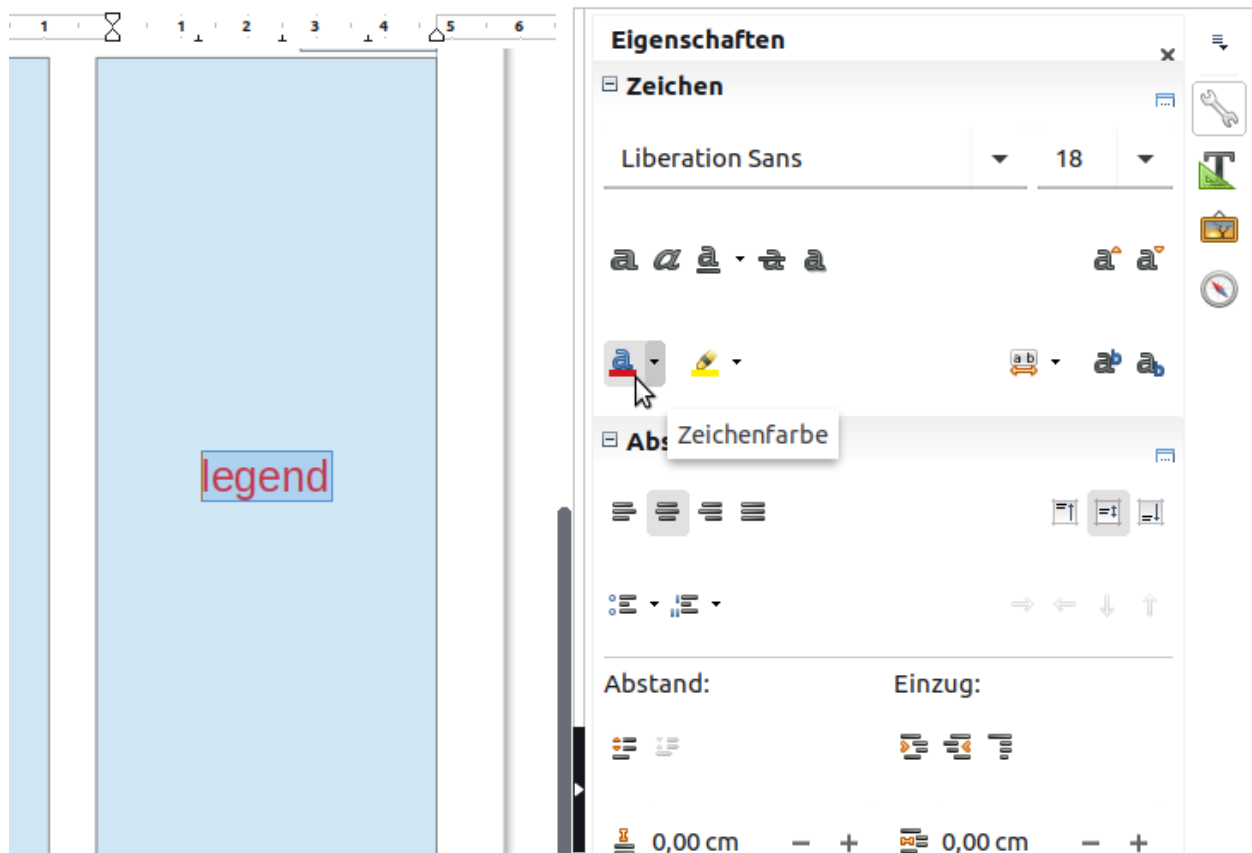
Logo on the legend page

If the legend shall be created on an additional page, the logo can be placed on this page too. This can be achieved with the dynamic element "legendpage_image". A new field on the non-printable layer has to be created and the name changed to "legendpage_image" (**Menu: Modify -> Name...**). The desired logo or image has to be saved in the directory **app/Resources/MapbenderPrintBundle/images/** and its name needs to be changed to "legendpage_image.png".



Coloured texts

The text in the print template can be changed in many ways. Besides the size of the font, one can also change the colour of the text. To do so, a text field via **Menu: Insert -> Text Box** needs to be inserted. To change the colour of the text, select the text in the text field (here: "title"). The colour can be changed in the tab **Properties -> Character**:



Or **right-click the selected text** → dialogue window **Character** → **Font Effects** → **Font color**.



The change of the colour of the dynamic field “title” to blue can look like this:



1 : 5000



Test

Test



The change of the font size works in an analogous manner.

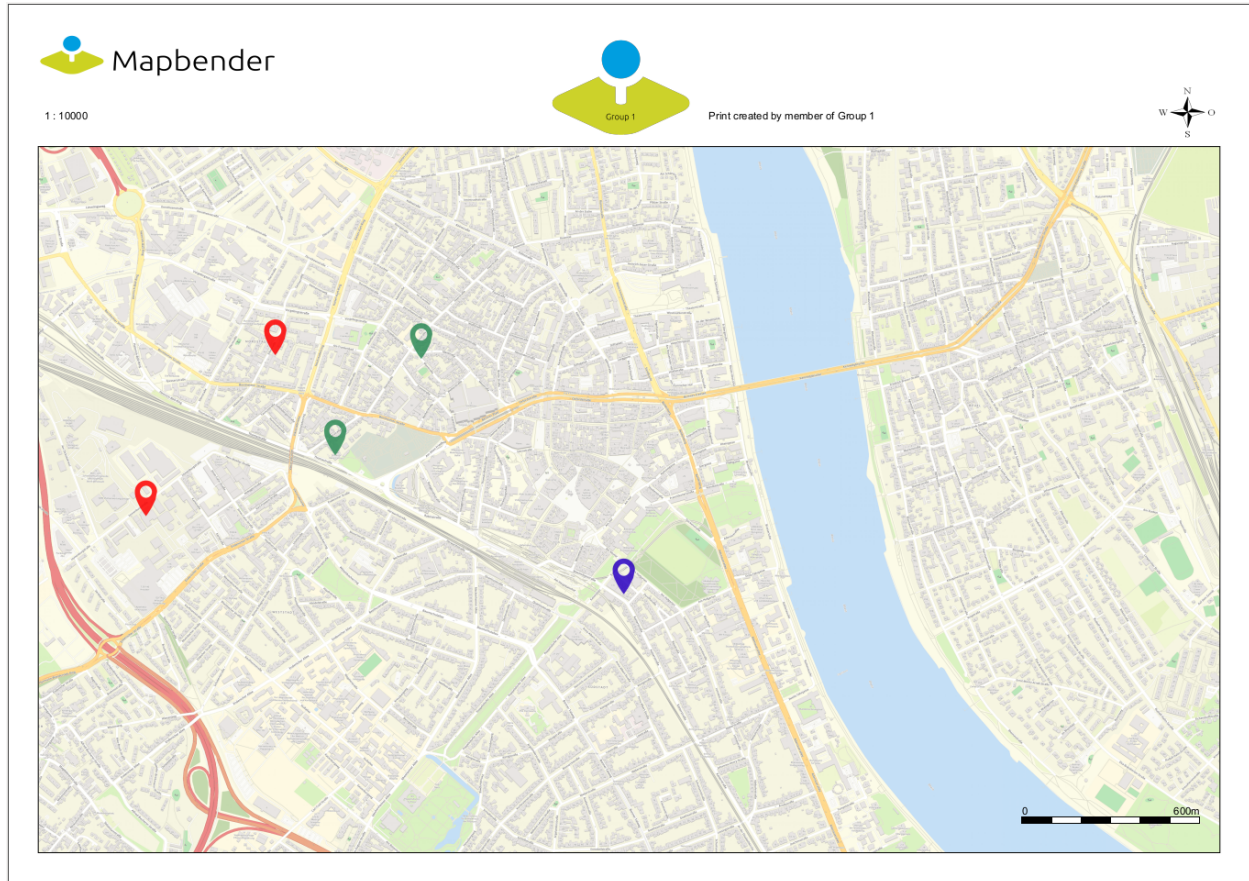
Dynamic images and dynamic texts

Dependent on a group, prints can be created with different logos and texts (e.g. the name of the commune and the individual logo). There are two objects which can handle this: "dynamic_image" and "dynamic_text". If these objects exist in the print layout [Mapbender and the user are members of a group], Mapbender will then search for an image with the name of the group (groupname.png). The picture will be displayed in the print in the object ["dynamic_image"]. The height of the object will be used to scale the image[,] the width will be calculated relative to the height. In the object ["dynamic_text"] the group description will be printed.

Depending on the group, the print can contain different images or descriptions (e.g. logo and name of the commune). This can be achieved through the dynamic elements "dynamic_image" and "dynamic_text". One can insert both elements in the ODG-print template on the non-printable layer and change their names (**Menu: Modify -> Name...**).

Note: Only one dynamic image and dynamic text of a group description will be printed. In other words: Mapbender always takes the first group into account. If a user is member of both groups „intern“ and „Group 1“, then „intern.png“ is taken as dynamic image and the group description of „intern“ will be adopted into the dynamic text field.

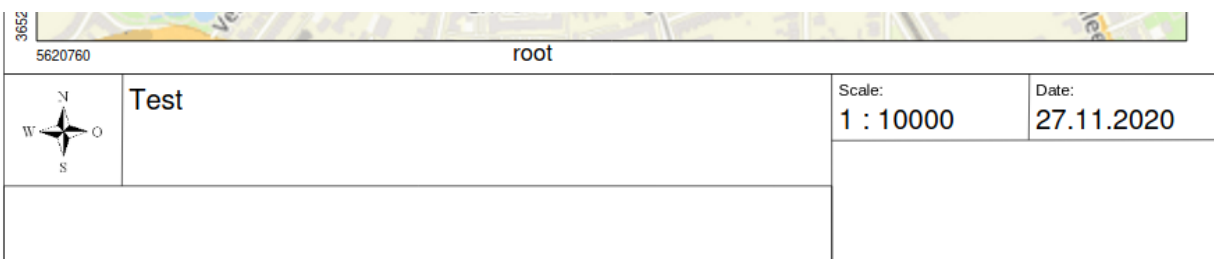
The print with a group named "Group 1" could look like this:



To use this feature, it is required that groups exist. How to create groups and users is described in the Mapbender documentation in the [Mapbender Quickstart](#).

The description of the group will be displayed in the field “dynamic_text” (e.g. copyright message). The element “dynamic_text” looks for a group description that is given in the first assigned group of the print. You can implement the dynamic text independently from the dynamic image.

It is possible to print a user name. In order to do this, one has to insert **user_name** into their print template. The logged-in user’s name is then printed at the defined location. If no user is logged in, nothing is printed.



The printing process

Printing feature information for a selected element

A feature can be selected via digitizer or Feature Info.

The concept is to pass the `feature_type`-name and the selected object ID to the print. Mapbender will then get all the feature data for the selected object and will look for fields in the print template. If a print template field is defined, the data will be printed to this field.

In the next steps it is described how this functionality can be configured. The documentation relies on the POI table that is used in the digitizer example.

You find the configuration and an example print-template in the Workshop/DemoBundle at <https://github.com/mapbender/mapbender-workshop>

There are some steps you have to follow:

1. Create a print template that refers to the feature columns
2. Define a `featureType` and refer to your new print template in your `config.yml`
3. Call feature print from FeatureInfo
4. Or call feature print from digitizer

1. Create a print template that refers to the feature columns

Define text fields in the print template for every information you would like to print for the selected object. The text field name has always the prefix `feature.`, followed with the name of the attribute (column) to export.

```
feature.name for column name of table poi
```

2. Define a `featureType` and refer to your new print template in your `config.yml`

```
parameters:
  featureTypes:
    feature_demo:
      connection: search_db # Name of the database-connection from the config.yml
      table: public.poi # Table-name in which the FeatureTypes are stored
      uniqueId: a_gid # Column-name with the unique identifier
      geomType: point # Geometry-type
      geomField: geom # Column-name in which the geometry is stored
      srid: 4326 # source EPSG-code of data
      print: # print template to offer for feature data print
        templates:
          - template: a4_portrait_official_feature_data_demo
            label: Demo with feature information print (portrait)
          - template: a4_landscape_official_feature_data_demo
            label: Demo with feature information print (landscape)
```

3. Call feature print from FeatureInfo

Note: FeatureInfo is the information output from a OGC WMS service. It offers information for features at a click position.

When you configure a WMS, you can generate a link with the following reference that will trigger the print with feature information.

The following code is an example for a MapServer information template.

```
<table>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<tr>
<td class="th_quer">Print</td>
<td><a href="" onclick="parent.$('.mb-element-map').data('mapQuery').olMap.setCenter([[x],[y]]);parent.
↵.$('.mb-element-printclient:parent').data('mapbenderMbPrintClient').printDigitizerFeature('feature_demo
↵',[gid]);parent.$('.mb-element-featureinfo:parent').data('mapbenderMbFeatureInfo').deactivate();
↵return false">print feature information</a>
</td>
</tr>
</table>
```

The FeatureInfo will open a dialog with a link *print feature information*. When you click on the link, the print dialog opens and offers the print templates that are defined for the feature type.

You can choose the desired region and create a print PDF. The PDF will contain the information for the selected feature.

4. Or call feature print from digitizer

The functionality can also be integrated in the digitizer. It will offer a new button *print* in every feature information dialog.

To activate the functionality, add the following parameter to the digitizer configuration.

```
printable: true
```

With click on the print button the print dialog opens and offers the print templates that are defined for the feature type.

Note: The flexibility to move the print frame won't stop you from choosing a region that does not contain the feature that was selected. In this case, the feature information does not match to the features that are displayed.

Queued Print

The queued print is an experimental print feature for Mapbender which comes with an advanced background print system. Right now, it's still in experimental state due to several potential cache memory regeneration problems on more complex server structures. The queued print is implemented since Mapbender 3.0.8, but deactivated by default. If you choose to activate it, you can use the feature via command line (either manually or as a cronjob). Queued print helps improving resource-intense print jobs, because the queue can manage the print jobs more easily in the background (compared to direct print). In the meantime, you're free to work with Mapbender in other ways.

Queued print: Configuration

To activate the queued print, open the parameters.yml file of your Mapbender installation and insert the following parameter:

```
mapbender.print.queueable: true
```

To deactivate the queued print, either set the parameter to 'false' or delete the whole parameter. As soon as the queued print section is set to true, you can configure a queued print in the Mapbender backend. Open your PrintClient element and adjust the new options "Mode" and "Job queue". To activate the queue, set "Mode" to 'queued'. 'Direct' will implement the default print surface to your Mapbender application. Moreover, the privacy preferences of your print queue are changeable via the "Job queue" field ('global' or 'private'). After adjusting, save the element and open the application.

Mode	<input type="text" value="queued"/>
Job queue	<input type="text" value="global"/>

Queued print: Bash commands

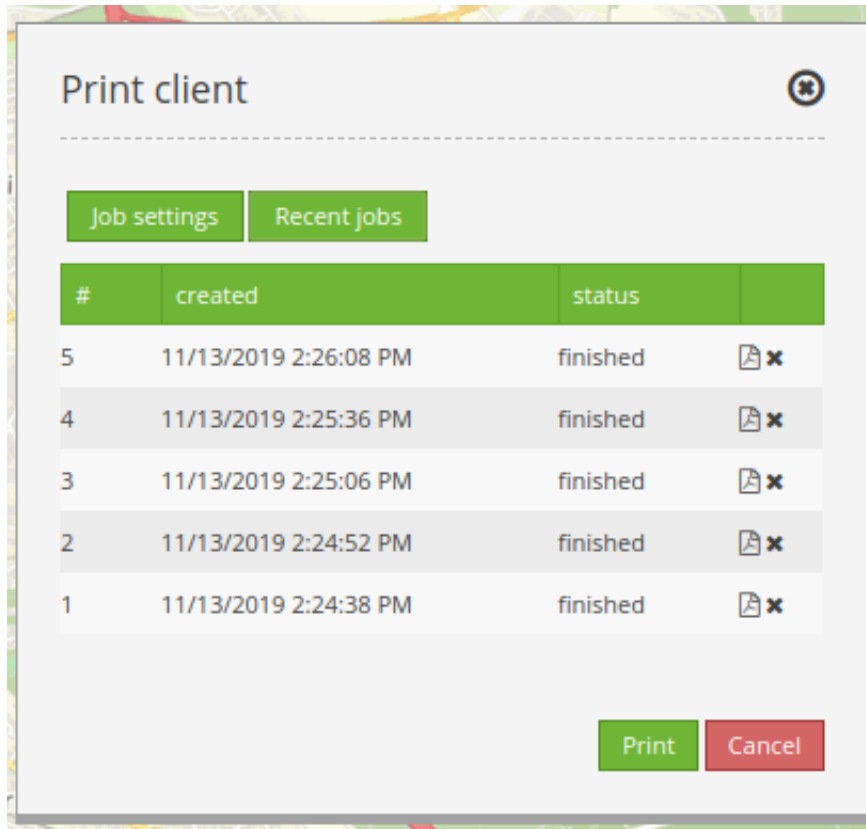
After the setup, the queued print can be controlled with several bash commands, which are as follows:

```
mapbender:print:queue:clean
mapbender:print:queue:dumpjob
mapbender:print:queue:gcfiles
mapbender:print:queue:next
mapbender:print:queue:repair
mapbender:print:queue:rerun
mapbender:print:runJob
```

Note: To run the commands, open a terminal and head to the Mapbender application directory. Find detailed information on the commands under [app/console commands](#).

Queued print: Usage

When using the queued print in the frontend, there are two options: The tab “Job settings” offers the same print settings as the direct print. If the queued print has been set up right, a tab called ‘Recent jobs’ appears next to the ‘Job settings’ tab. If this tab is chosen, a chronological list of your print jobs will be shown. A new job will appear in the list after the “Print” button is clicked.



To start the printing process, type in the bash command

```
app/console mapbender:print:queue:next --max-jobs=0 --max-time=0
```

to execute a print process in the command line. This process starts all the jobs that are added into the print queue list automatically. Alternatively, you can choose to adjust the parameters and create a fitting cronjob. Terminate the process with ‘CTRL + C’. If a print job is finished, it will be listed as “finished” in the status column of the list. Afterwards, the PDF button will open the printable PDF file.

Memory Limits

Queued Print

Print jobs can be resource intensive and may exceed your initially set php.ini memory limit. Therefore it is possible to increase the required memory limit manually. This is an advantage for users who are working with large print templates.

Note: Never reduce the memory limit.

To increase the memory limits for the queued print, adjust *mapbender.print.queue.memory_limit* (string; default is 1G). Caution: This parameter does not allow 'null' as value.

Direct Print

To increase the memory limit of the direct print, adjust *mapbender.print.memory_limit* (string or null; default is null) to your possible memory contingent. If the parameter is set to 'null', Mapbender print will look for your php.ini value. If you set the parameter to a value which is accepted by your php.ini-configuration file, Mapbender print uses this limit instead of the php.ini limit (possible values are e.g. 512M, 2G, 2048M, etc.) Use '-1' for unrestricted memory usage.

Set a WMS Tile Size Limit

If the printing process fails to export a WMS service into the PDF file, the following amendment needs to be made in the parameters.yml file. This is because, under certain circumstances, the requested pixel size for the WMS is too large, resulting in the service no longer delivering images.

```
mapbender.imageexport.renderer.wms.max_getmap_size: 8192
```

These limitations set the maximum possible WIDTH= and HEIGHT= values for the export request. In the GetCapabilities request of the respective service, the maximum resolution is defined under *MaxWidth* and *MaxHeight*, which means that the getCapabilities request already sets the limit - 8192 is the default value, which may need to be further adjusted. The mentioned parameters can also be defined independently of each other: Use *mapbender.imageexport.renderer.wms.max_getmap_size.x* for the **WIDTH=** parameter and *mapbender.imageexport.renderer.wms.max_getmap_size.y* for the **HEIGHT=** parameter.

3.4 Search

3.4.1 Search Router

This element creates a configurable search formular with a configurable result output in which generic SQL search is supported.

Search Router

Search Cities

▼

Name*

Bonn

Search

Reset

Results: 1

ID	Ortsname	Einwohner
2946447	Bonn	313125

Configuration

Note: The SearchRouter needs access to the database where the search tables are. You have to define a new database configuration to be able to connect with the geo database. Read more about this at [YAML Configuration \(Configuration and Application files\)](#).

The element may be integrated into the sidepane or via a button into from the toolbar. To configure a button visit the documentation at [Button](#).

Edit element

Title	<input type="text" value="Search router"/>	
Width *	<input type="text" value="700"/>	⌵
Height *	<input type="text" value="500"/>	⌵
Routes *	+	
Title *	<input type="text" value="Search Cities"/>	🗑
Configuration *	<pre>class: Mapbender\CoreBundle\Component\SQLSearchEngine class_options: connection: geobasis relation: cities1000 attributes:</pre>	

Save

Cancel

- **Title:** Title of the element. The title will be listed in the Layouts section in the backend. It is shown as sidepane tab/button title (usage in sidepane) or dialog title (usage via button).
- **Width:** Width of the dialog (only for dialog, not sidepane)
- **Height:** Height of the dialog (only for dialog, not sidepane)

You can define Searches (Routes) with the + Button. Each Search has a title which will show up in the search form in a selectbox. The definition of the search is done in YAML syntax in the textarea configuration.

- **Routes:** Collection of search routes.
- **Title:** Search Title (appears, when a search is added to Routes by using +)
- **Configuration:** Field to configure the search (appears, when a search is added to Routes by using +)

Route Configuration

In the textarea configuration you define all important information for each search.

- the database connection
- the search tables/views and columns
- the design of the form
- and the design of the result table
- the design of the results on the map

```
class: Mapbender\CoreBundle\Component\SQLSearchEngine
class_options:
  connection: geodata_db      # database alias from config.yml
  relation: polygons          # search table
  attributes:                  # columns that are used in the form and in the result table
    - gid
    - name
    - type
    - city
  geometry_attribute: geom    # reference to the geometry column
form:
  name:
    type: Symfony\Component\Form\Extension\Core\Type\TextType
    options:
      required: true
    compare: exact
  city:
    type: Symfony\Component\Form\Extension\Core\Type\TextType
    options:
      required: false
      label: City/Town
    compare: ilike
results:
  view: table
  count: true
  headers:
    gid: ID
    name: Name
    city: City/Town
  callback:
    event: click
    options:
      buffer: 10
      minScale: null
      maxScale: null
```

Class and Class options

The configuration starts with the definition of the class (always class: MapbenderCoreBundleComponentSQLSearchEngine) followed by the information about the database connection, definition of the search table, the columns that are used and the geometry column.

Form

In the form section you can define the setup of the form and define textfields and selectboxes.

You refer to the table columns and define the type, options and compare.

Type

The form supports two types - text and choice.

You have to define a configuration for each table column you would like to provide in the search form. The configuraton starts with the column name (in the example it is the column called name).

- label - you can define a label parameter (if not defined the capitalized column name will be used).
- required: You can define whether a type should be required (default is false). With required: true a mandatory field is defined. That means that the user has to define a search term here before the search can run. Headings of required fields are marked with a red '*'.

You also can define a compare mode. See section 'comparison mode'.

Type text

Type **text** allows you to provide text fields for your search formular.

Type text supports autocomplete. If you want to add autocomplete to the field you have to add the additional attr-parameters.

Supported autocomplete paramters are:

- data-autocomplete: on - parameter to activate autocomplete
- data-autocomplete-distinct: on - paramter to activate distinct autocomplete
- data-autocomplete-using: column1, column2 - define other column/s that should be also considered on autocomplete

Type text; example with autocomplete:

Cities

Name*

Bon|

- Boniswil
- Bonito
- Bonn**
- Bonnac-la-Côte
- Bonnanaro
- Bonnat
- Bonndorf im Schwarzwald
- Bonnefamille
- Bonnelles
- Bonnemain

```
form:
  name:
    type: Symfony\Component\Form\Extension\Core\Type\TextType
    # input box for text
  options:
    label: Name
    required: true
  attr:
    data-autocomplete: 'on' # activate autocomplete
    data-autocomplete-distinct: 'on' # only show equal results once
    data-autocomplete-using: type, city # autocomplete, list of input fields (with comma
    # seperated), WHERE input
    compare: exact
```

Type choice

Type choice allows you to provide a selectbox in your search formular.

For type choice you can define a placeholder. This is a text that is shown before an option is selected.

You have to define the choices for the selectbox. You define a value and a key.

- key - will be send in the search query
- value - is show as text in selectbox

Note: From Mapbender 3.2 on onwards, you should use the value: key definition and type: Symfony-

ComponentFormExtensionCoreTypeChoiceType

- Type choice; example with different selection options via dropdown:

The screenshot shows a web form with the title "User type". Below the title is a dropdown menu. The dropdown is currently open, showing a list of options. The first option is "Please select..." with a small downward arrow icon to its right. Below this are four more options: "Company", "Administration" (which is highlighted in green), "University", and "User". A mouse cursor is visible over the "Administration" option.

```
type:
  type: Symfony\Component\Form\Extension\Core\Type\ChoiceType
  # box with selection options as dropdown list
  options:
    label: User type
    required: false
    placeholder: 'Please select...'
    choices:
      Company: A
      Administration: B
      University: C
      User: D
      Something else: E
  compare: exact
```

Comparison Mode

For every field a comparison mode can be set. This is used by the engine when the query is sent.

The following comparison modes are supported:

- **exact:** exact comparison (key = val)
- **exact:** comparison ignoring cases (case-insensitive)
- **like:** Standard, both-side 'like'
- **like-left:** left-hand 'like'
- **like-right:** right-hand 'like'

- **ilike:** both-side 'like', (case-insensitive - *searchstring*)
- **ilike-left:** left-side 'like' (case-insensitive - *searchstring*)
- **ilike-right:** right-side 'like' (case-insensitive - searchstring*)

Result

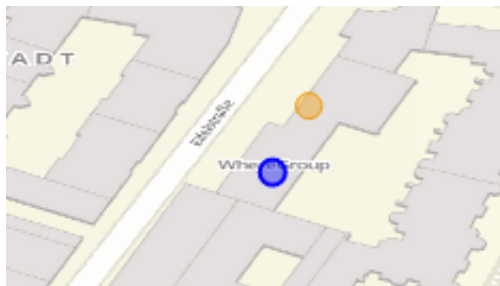
In the section **results** the definition for the result table and styling is none.

```
results:
  view: table
  count: true
  headers:
    gid: ID
    name: Name
    city: City/Town
  callback:
    event: click
    options:
      buffer: 10
      minScale: null
      maxScale: null
```

- **view:** is always set to table. No more options are possible
- **count:** true or false to show the number of results
- **headers:** definition of the columns to display and the alternative labeling
- **callback:** define the action for the click event on a result item
- **event:** only click is supported
- **buffer:** zoom to the result item with a defined buffer
- **minScale and maxScale:** zoom to the result item in a scale between minScale and maxScale

Styling the Results

By default the results are shown in the default-OpenLayers Style.



You can overwrite this by handing over a styleMap-Configuration.

Three different styles are configured:

- **default:** default style
- **select:** style on select

- **temporary:** style on mouse-over

```

results:
  [...]
  styleMap:
    default:
      strokeColor: '#00ff00' # border color
      strokeOpacity: 1       # border opacity (1 - opaque / no transparency)
      strokeWidth: 3         # border width
      fillColor: '#f0f0f0'   # fill color
      fillOpacity: 0         # fill opacity, (0 full transparency)
      pointRadius: 6         # size of the point symbol
    select:
      strokeColor: '#0000ff'
      strokeOpacity: 1
      strokeWidth: 4
      fillColor: '#ff00ff'
      fillOpacity: 0.8
      pointRadius: 10
    temporary:
      strokeColor: '#0000ff'
      fillColor: '#0000ff'
      fillOpacity: 1

```

In the default style the point-symbol interior is transparent (fillOpacity: 0). Only their outlines will be drawn in green.

The selected features will be drawn with a purple fill and an opacity of 0.8. The outline of the symbol is blue.

The temporary symbols on mouse-hover are blue points.



The styleMap settings override the default settings, so you only have to define the sections you want to overwrite. No extra styleMap is set the default style will be used.

Note, that the hexadecimal color values have to be stated in quotation marks, because # would be interpreted as a comment instead.

Configuration Examples

1. Example

In this example a search was configured for the Mapbender user and added into the sidepane, usable under the + in Layouts.

Sidepane		⚙
Title	Type	+

The configuration dialouge for this example looks like this:

Edit element

Title*

Search router

Target

Map

☐ Dialog*

Timeout factor*

3

Width*

700

Height*

500

Routes*

+

Title*

Mapbender User

Configuration*

```

class: Mapbender\CoreBundle\Component
\SQLSearchEngine
class_options:
  connection: demo
  relation: mapbender_user

```

Update

Cancel

The element title (*Title*) is Search. It is again displayed as a title in the sidepane. The checkbox is unchecked, because the element is implemented into the sidepane and not as a button. The *Timeout factor* is set to 2. The fields *Width* and *Height* are filled, but they won't be used in this application, because the element is configured in the sidepane. It is implemented via the + -Symbol and *Routes* into the search, called (*Title*) Mapbender User. The yaml-Configuration of the Element is written in *Configuration*. In Completion it reads:

```

class: Mapbender\CoreBundle\Component\SQLSearchEngine
class_options:
  connection: geodata_db          # database (on which the element has access)

```

(continues on next page)

(continued from previous page)

```

relation: mapbender_user           # table (on which the element has access)
attributes:                       # table columns (which the element addresses)
- gid
- orga
- town
- usertype
geometry_attribute: the_geom      # definition of the geometry column
form:                             # configuration of the form
orga:                             # search field (e.g. search for specific Mapbender User)
  type: Symfony\Component\Form\Extension\Core\Type\TextType
  options:
    required: false                # no mandatory field
    label: 'Mapbender User'        # caption of the search field
    attr:                          # additional definable attributes
      data-autocomplete: 'on'      # auto-completion of search words
      data-autocomplete-distinct: 'on'
    compare: ilike                 # see section 'comparison mode' on this page
town:                             # search field (e.g. search for specific city)
  type: Symfony\Component\Form\Extension\Core\Type\TextType
  options:
    required: false                # no mandatory field
    label: City                    # caption of the search field
    attr:
      data-autocomplete: 'on'
      data-autocomplete-distinct: 'on'
    compare: ilike
usertype:                         # search field (search for specific User type)
  type: Symfony\Component\Form\Extension\Core\Type\ChoiceType
  options:
    placeholder: 'Please select...' # displayed text in field before entering a search
    choices:                       # choices need to have the following format: "entry in the
↳ database column: "displayed name in the drop down list"
      1: Company
      2: Administration
      3: University
      4: User
    required: false                # no mandatory field
    label: User type                # caption of the search field
    compare: exact                 # see section 'comparison mode' on this page
results:                           # configuration of the shown results list
  view: table                       # display results as table
  count: true                       # show number of results
  headers:                         # column title; format: column title in the database: column
↳ title shown in the table
    gid: ID
    orga: 'Mapbender User'
    town: City
callback:
  event: click
  options:
    buffer: 10
    minScale: null
    maxScale: 10000
styleMap:                         # Styling points on the map
  default:                         # Styling of all points on the map
    strokeColor: '#003366'

```

(continues on next page)

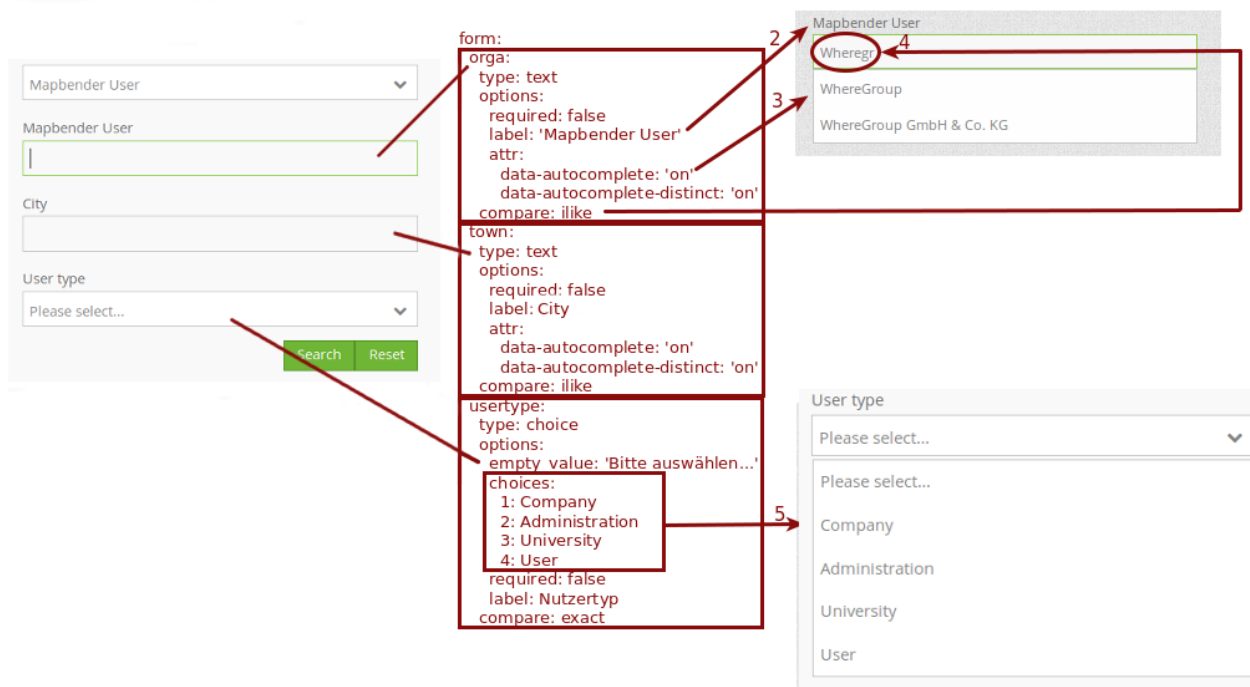
(continued from previous page)

```

strokeOpacity: 1
fillColor: '#3366cc'
fillOpacity: 0.5
select:                                     # Styling of the selected point on the map
strokeColor: '#330000'
strokeOpacity: 1
fillColor: '#800000'
fillOpacity: 0.5
temporary:
strokeColor: '#0000ff'
fillColor: '#0000ff'
fillOpacity: 1

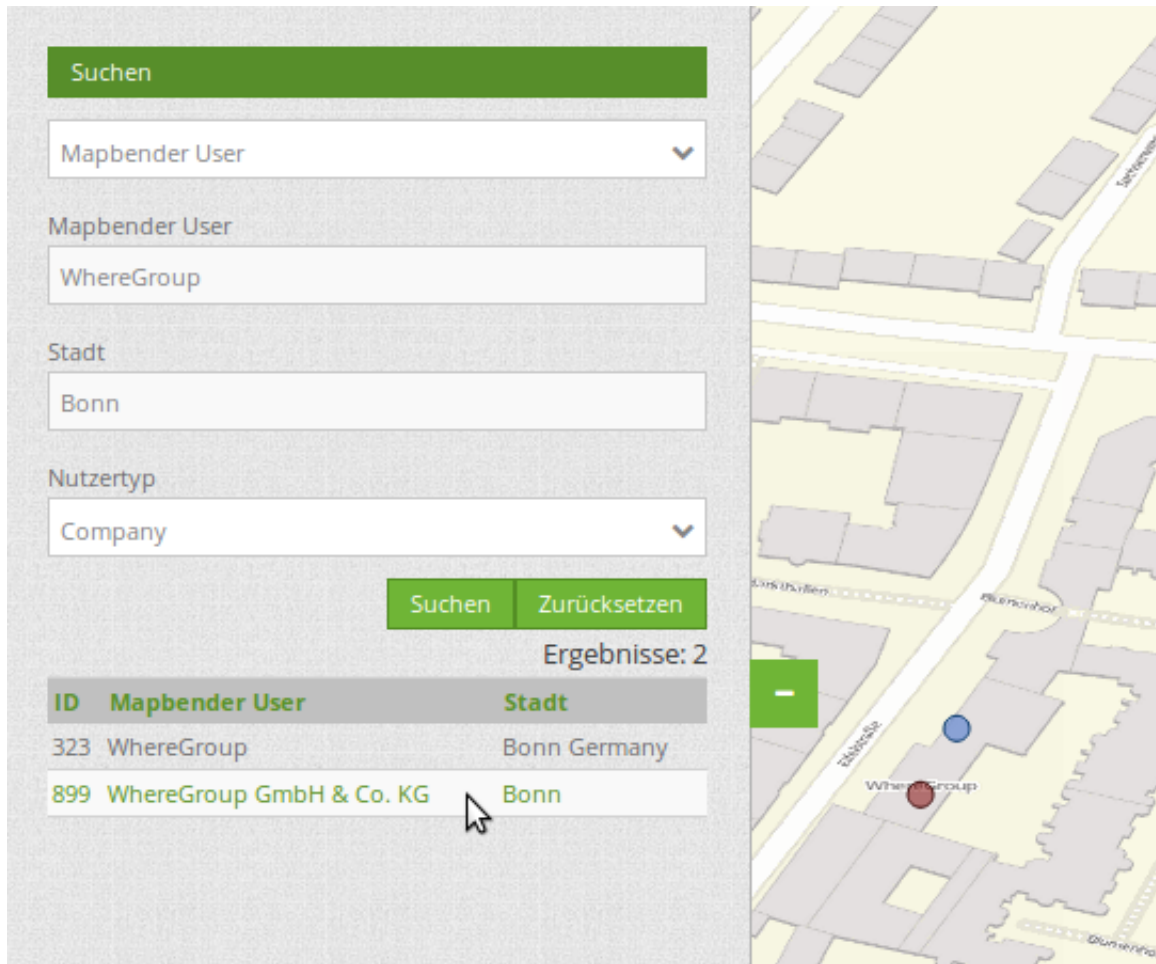
```

This picture illustrates which consequences the configurations in the yaml-definition have for the search formula:

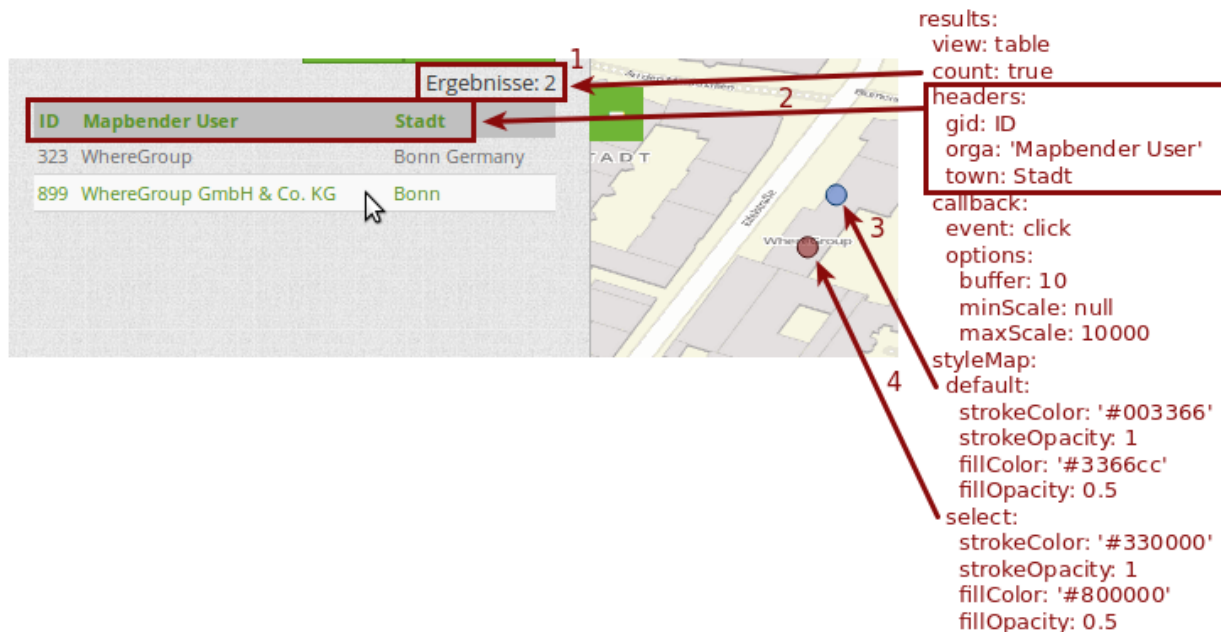


Displayed is the excerpt of the yaml-definition configuring the formula. Columns orga, town and usertype are used in the formula and implemented as the fields Mapbender User, Town and Usertype. Mapbender User and Town are type text, Usertype can be of various types. The text that should be displayed here, if nothing is selected yet, is "Please select..." (Nr. 1 – placeholder: 'Please select...'). The title above these fields is set with a label (Nr. 2). The attribute data-autocomplete: 'on' results in a dropdown menu with recommendations from the database (Nr. 3). Because compare: ilike is enabled it is not necessary to write the exact word. The search will find results that are only similar to the written term (Nr. 4 – Wheregr (the g is lowercase, nevertheless WhereGroup with uppercase G was found)). The fieldtype choice is variable, possibilities are defined in choices (Nr. 5). The table contains the possibilities as numbers (1, 2, 3, 4). In this example every number represents a text, which should be displayed in the dropdown menu.

A complete search for the Mapbender User WhereGroup, in the Town Bonn, of the Usertype Company and the found results will look like this:



This picture illustrates the consequences our configuration of the yaml-definition had on the display of the results.



Here only the configuration of the results is shown. The number of results is shown because `count: true` (Nr. 1) is enabled. The titles of the columns are defined in `headers` (Nr. 2). Here the name of the column is mentioned first, to define which results are referenced. After the colon we set the title which will be displayed in the application. In the block `styleMap` the points are styled. The block `default` (Nr. 3) references all points, and the block `select` (Nr. 4) only selected points.

Because none of these fields are mandatory the search will work with only one field.

2. Example

The following example uses the german geographical names data in 1:250.000 from the [Bundesamt für Kartographie und Geodäsie](#). The data was extracted to `gn250_p` table in the `gisdb` database and can be used for the search. The data has some specific columns:

- `id`: the id of the dataset
- `name`: the name of the dataset
- `kreis`: the administrative county (not for every dataset)
- `oba_wert`: the type of data (e.g. station, museum, etc.)

Example of a route-configuration in the configuration area:

```
class: Mapbender\CoreBundle\Component\SQLSearchEngine
class_options:
  connection: geodata_db
  relation: gn250_p
  attributes:
    - id
    - name
    - kreis
    - oba_wert
  geometry_attribute: geom
form:
  name:
    type: Symfony\Component\Form\Extension\Core\Type\TextType
    options:
      required: true
      compare: ilike
results:
  view: table
  count: true
  headers:
    id: ID
    name: Name
    kreis: Landkreis
    oba_wert: Art
  callback:
    event: click
    options:
      buffer: 10
      minScale: null
      maxScale: null
```

YAML-Definition

This template can be used to insert the element into a YAML application.

```
target: map                                # ID map element
asDialog: true                             # true: results in dialog box
timeoutFactor: 3                           # timeout factor (multiplied by autocomplete
↳(deceleration) to prevent autocorrect after a search has been started
height: 500                                # height of dialog
width: 700                                 # width of dialog
routes:                                     # collection of search routes
  demo_polygon:                             # machine-readable name
    class: Mapbender\CoreBundle\Component\SQLSearchEngine # path to used search engine
    class_options:                             # options passed to the search engine
      connection: geodata_db                  # search_db, DBAL connection name, ~ for
↳default
      relation: polygons
      attributes:                             # list of columns, expressions are possible
        - gid
        - name
        - type
        - city
      geometry_attribute: geom                # name of the geometry column, attention:
↳projection needs to match with the projection of the map element
      form:                                   # declaration of the search form
        name:                                 # field name, column name
        type: Symfony\Component\Form\Extension\Core\Type\TextType # input field, normally
↳text or numbers
        options:                             # declaration of the input field
          required: false                     # HTML5, required attributes
          label: Name                         # custom label, otherwise field name used
          attr:                               # HTML5, required attributes
            data-autocomplete: on             # attribute to activate autocomplete
            data-autocomplete-distinct: on    # attribute to activate distinct autocomplete
            data-autocomplete-using: type     # autocomplete, list of input fields (with
↳comma seperated), WHERE input
          compare: ilike                      # see section 'comparison mode' on this page
        city:
          type: Symfony\Component\Form\Extension\Core\Type\TextType
          options:
            required: false
            label: City/Town
          compare: ilike
        type:
          type: Symfony\Component\Form\Extension\Core\Type\ChoiceType
          options:
            placeholder: Please select a type.
            required: false
            choices:
              A Company: A
              B Administration: B
              C University: C
              D User: D
              E Something else: E
      results:
        view: table                          # display results as table
        count: true                          # show number of results
```

(continues on next page)

(continued from previous page)

```

headers:                                     # column title
  gid: ID                                   # column name -> header
  name: Name
  type: Type
  city: City/Town
callback:                                   # click event
  event: click                             # click or mouseover event
  options:
    buffer: 10                             # buffer (before zoom)
    minScale: ~                             # scaling boundaries for zoom, ~ for no
↪ boundaries
    maxScale: ~
styleMap:
  default:
    strokeColor: '#00ff00'
    strokeOpacity: 1
    fillOpacity: 0
  select:
    strokeColor: '#ff0000'
    fillColor: '#ff0000'
    fillOpacity: 0.4
  temporary:
    strokeColor: '#0000ff'
    fillColor: '#0000ff'
    fillOpacity: 1

```

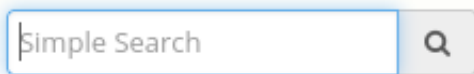
3.4.2 Simple Search

Simple Search offers a single field search or keyword search. The search query is transmitted to a search service. The search servers Apache Solr, Nominatim, photon or an OGC API Features service can be used.

An input field is offered which can be integrated directly into any region. Added to content you have to define the position. Simple Search sends the entered search term to a configurable URL and receives JSON-formatted data, which contain a label and geometry attributes for each entry.

The geometry data can be encoded in WKT or GeoJSON format.

It is configurable which information from the result json should be shown as result



Configuration

Edit element

Title	Simple Search
Placeholder	
Query URL *	https://photon.komoot.io/api/?limit=20&&lat=50.739&lon=7.08428
Query URL key *	q
Query Whitespace replacement pattern *	
Query key format *	%s
Token (JavaScript regex)	[^a-zA-Z0-9äöüÄÖÜß]
Token search (JavaScript regex)	([a-zA-ZäöüÄÖÜß]{3,})
Token replace (JavaScript regex)	\$1*
Collection path	features
Label attribute *	\${properties.name} \${properties.street} \${properties.housenumber} \${property
Geom attribute *	geometry
Geom format *	GeoJSON
Source SRS	EPSG:4326
Delay *	300
Result buffer *	5
Result minscale *	100
Result maxscale *	1000
Result icon url *	/bundles/mapbendercore/image/pin_red.png
Result icon offset *	-6,-32

Save
Cancel

- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Position:** Position (only when used in Map area). Options: ‘left-top’, ‘right-top’, ‘left-bottom’, ‘right-bottom’
- **Placeholder:** Text to display when the search field is empty. If a placeholder is not defined, the content of the title is used instead.
- **Query URL:** Solr URL for the search (e.g. `http://localhost:8080/solr/core0/select?wt=json&indent=true`) or Nominatim URL.
- **Query URL key:** The query parameter key to append (e.g. `q`).
- **Query Whitespace replacement pattern:** Pattern for replacing white spaces.
- **Query key format:** Simple search format (e.g. `%s`).
- **Token search/replace (JavaScript regex):** **Tokenizer split/search/replace regexp.**
 - **Token**, e.g.: `^[a-zA-Z0-9äöüÄÖÜß]`
 - **Token search**, e.g.: `([a-zA-ZäöüÄÖÜß]){3,}`
 - **Token replace**, e.g.: `$1*`
- **Collection path:** Can be a dotted attribute path to extract from the query result (e.g. `response.docs`).
- **Label attribute:** Name of the attribute/s to show as result.
- **Geom attribute:** Name of the geometry data attribute (e.g. `geom`).
- **Geom format:** Geometry data format, can be WKT or GeoJSON (e.g. `WKT`).
- **Source SRS:** EPSG code of the spatial reference system (e.g. `EPSG:25832`)
- **Delay:** Autocomplete delay. Use 0 to disable autocomplete (e.g. `300`).
- **Result buffer:** Buffer result geometry with this (map units) before zooming (e.g. `10`).
- **Result minscale/maxscale:** Scale restrictions for zooming, ~ for none (e.g. `1000` und `5000`).
- **Result icon url:** Icon to display as result marker (e.g. `/bundles/mapbendercore/image/pin_red.png`).
- **Result icon offset:** Offset x and y for the icon (e.g. `0,0`).

Flexible configuration via `label_attribute`

The result JSON provides different information. It is possible to define one or several attributes for the result text.

```
label_attribute: label
```

The definition is relative to the Collection path. It is possible to use additional text to separate the attributes.

```
label_attribute: '${properties.address.city} ${properties.address.road} ${properties.address.house_↵number}'
```

```
label_attribute: 'Town: ${properties.address.city}: ${properties.address.road} - ${properties.address.↵house_number}'
```

YAML-Definition

```

placeholder: 'please search' # Text to display
↳ when the search field is empty. If a placeholder is not defined, the content of the title is used.
anchor: 'right-bottom' # Position (only
↳ when used in Map area). Options: 'left-top', 'right-top', 'left-bottom', 'right-bottom'
query_url: http://example.com/solr/core/0/select?wt=json&indent=true&rows=8 # Example Solr URL
↳ (e.g. ``http://localhost:8080/solr/core/0/select?wt=json&indent=true``) or Nominatim URL.
query_key: q # The query
↳ parameter key to append
query_ws_replace: # Pattern for
↳ replacing white spaces.
query_format: '%s' # Simple search
↳ format.
token_regex: [^a-zA-Z0-9äöüÄÖÜß] # Tokenizer split
↳ regexp.
token_regex_in: ([a-zA-ZäöüÄÖÜß]){3,} # Tokenizer search
↳ regexp.
token_regex_out: '$1*' # Tokenizer replace
↳ regexp.
collection_path: response.docs # Can be a dotted
↳ attribute path to extract from the query result.
label_attribute: label # Name of the
↳ attribute to use for entry labeling
geom_attribute: geom # Name of the
↳ geometry data attribute
geom_format: WKT # geometry data
↳ format, can be WKT or GeoJSON
sourceSrs: 'EPSG:25832' # Projection of the
↳ result data
delay: 300 # Autocomplete delay.
result: # Autocomplete delay.
↳ Use 0 to disable autocomplete.
  buffer: 50 # buffer result
↳ geometry with this (map units) before zooming
  minscale: 1000 # scale restrictions
↳ for zooming, ~ for none
  maxscale: 5000 # scale restrictions
↳ for zooming, ~ for none
  icon_url: /bundles/mapbendercore/image/pin_red.png # icon to display as
↳ result marker
  icon_offset: -6,-38 # Offset x and y for
↳ the icon

```

Set-up of Solr

- **Download:** <https://solr.apache.org/downloads.html>
- **Documentation:** <https://solr.apache.org/guide>
- **Quickstart:** <https://solr.apache.org/guide/solr-tutorial.html>

Set up of Nominatim

- **Download:** <http://nominatim.org/release-docs/latest/admin/Installation/>
- **Documentation:** <http://nominatim.org/release-docs/latest/>

Set up of photon

photon is an open source geocoder built for OpenStreetMap data.

- **Download & Documentation:** <https://github.com/komoot/photon>

Set up of OGC API Features

Several projects support OGC API Features like QGIS, GeoServer, pygeoapi.

HTTP Callbacks

- `/search`: Widget proxy which then queries configured URL. In dev-mode the final query URL will be returned as a `x-mapbender-simplesearch-url` header for easier debugging.

3.5 Share

Share offers a variety of possibilities that simplify the joint work with Mapbender applications: *Share URL* enables quick sharing of self-configured map states via URL, *View Manager* stores custom map states, *Application Switcher* enables cross-application switches and *Persistent Map View* simplifies the application compatibility with the web browser.

Certain view parameters are automatically contained in every application URL. Thus, users can share specific map views.

Contained view parameters include:

- center
- scale
- rotation
- spatial reference system

There is no extra configuration for this functionality. It is always turned on.

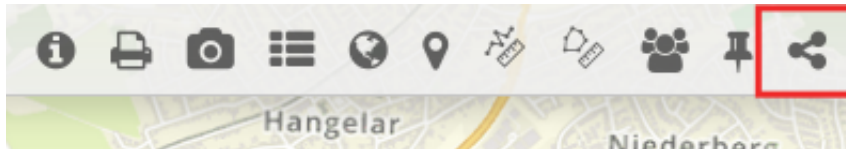
If a URL is opened up in a new browser tab, previously mentioned view configurations will be restored. Made changes can be undone/redone with the browser back/forward buttons.

Users will be sent back to the same part of the map if they refresh the page. They will not be sent back to the configured initial map view. In order to do so, users have to open the application again from the application list or manually delete the hash part of the application URL.

Note: The following information is not saved by the URL: layer selection, sorting, runtime additions, geometry features or source additions via WMS loader.

3.5.1 Share URL

URL sharing can be further simplified by integrating a respective element in the toolbar or footer.



After clicking on the button, the URL is saved to the clipboard. Default browser interactions (e.g. open in new tab) are also possible.

This element stores the following information:

- basic view parameters (center, scale, rotation, SRS)
- layer and layerset settings changes (selected / deselected layersets, sources and layers, layer opacity settings)

Note: The URL does not transfer dynamically added sources (via WmsLoader), dynamically removed layers or sources (via Layertree context menu) or changes to the source or layer order (via Layertree drag&drop)

Configuration

Add element

Toolbar - Share URL

☒ Show label

Title

Share URL

Tooltip

Back

Add

Cancel

YAML-Definition:

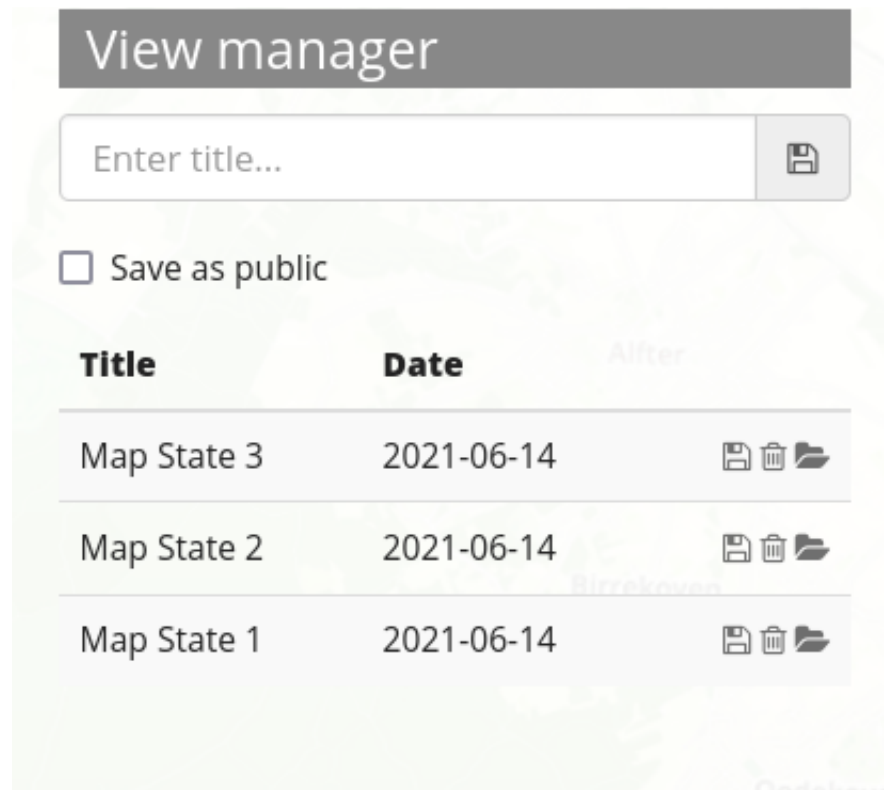
This template can be used to insert the element into a YAML application.

```
title: Share this map view          # Optional custom title, uses default title "Share URL" if omitted
↪(string or empty).
class: Mapbender\CoreBundle\Element\ShareUrl
tooltip: I am displayed on hover    # Optional custom tooltip, same as title if omitted (string or
↪empty).
label: true                        # Enables display of title, set as false will only display icon
↪(default: true).
```

3.5.2 View Manager

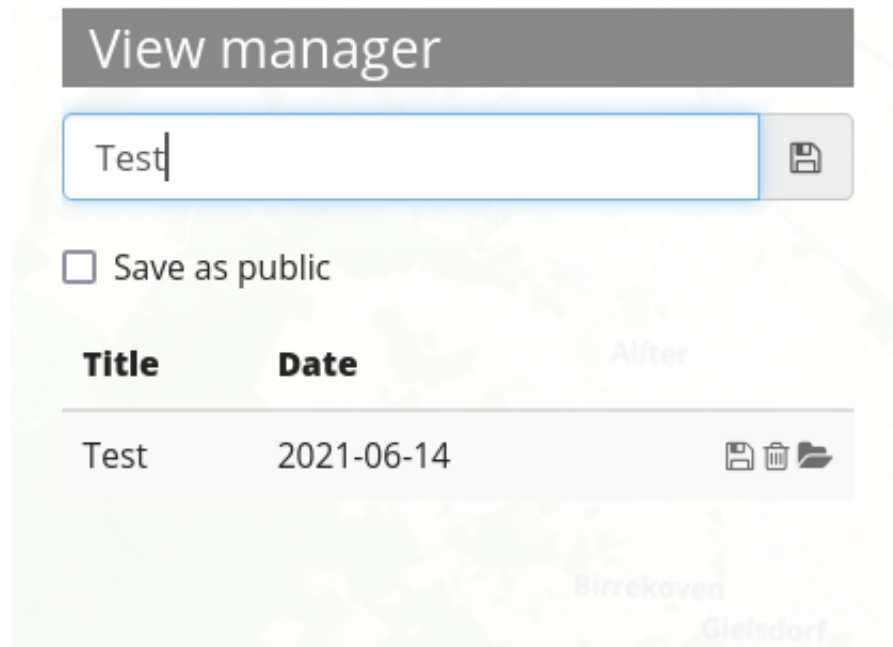
This element View Manager stores and reapplies map states. These contain the following map parameters: center, scale, srs, rotation, layer/source, layerset selection, source opacities. Saved states are always reapplied on top of the current configuration. This means that changed application configurations will remain in effect after reload of the map state.

Note: Thus far the View Manager can only be implemented in the Sidepane.



Basic operations

Each state must be given a title for reidentification. For saving the current map view as a new state, enter a title and click on the adjoined save button.



The most basic interaction (always available) is re-applying the map state stored in the entry. This option is always on: The saved map state will be reapplied as soon as “Apply” is hit on the selected map state in list view. Moreover, entries may offer a “Replace” interaction. This will overwrite the map state stored in the entry, and will also update the title, using the global title input field. Also, entries may offer a “Delete” interaction (with an extra confirmation step).

Note: The View Manager does *not* store or reapply the following configurations:

- any interactively added sources (via WmsLoader)
- any interactively removed layers (via Layertree context menu)
- any values for WMS dimensions
- any dynamically rendered geometries (Digitizer etc.)

Access rights

Each map state is attributed to an application and further separated into a public and user-private list. The rights to save, reapply or delete map states are defined in the element configuration. Furthermore, rights to show private lists and dates as well as the permission for anonymous users to save map states can be set here.

In general, access checks on public entries are suspended for the root user. The administrator can create, update and delete public entries at will.

Anonymous users are excluded from working with private entries and they can never delete public entries. Their ability to create and update public entries is gated through the “Allow saving to anonymous users” option. If this checkbox is deactivated, their access to public entries is downgraded to read-only. If the goal is to exclude anonymous visitors completely, a `ROLE_USER` access restriction has to be set on the entire element.

Configuration

Add element

Sidepane - View manager

Title

View manager

☒ Show private list☐ Show date☐ Allow saving to anonymous users

Public list

Read only



Note: the administrator can always delete public entries.

Back

Add

Cancel

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
publicEntries      # String or empty (falsy value disables public entries entirely); other allowed
↳ values are ro (read only), rw (allow read and write), rwd (allow read and write and deletion)
↳ (default: ro)
privateEntries     # Turns user-private states on, with full usage (save, reapply, delete) (default:
↳ true)
allowAnonymousSave # Extend right to save public entries also to anonymous users (default: false)
showDate:          # Show date of creation or last update in entry listing (default: true)
```

3.5.3 Persistent map view

This feature makes certain view parameters and certain source settings persistent. This enables an application to be closed and opened up again in the same browser without losing certain information.

Persisted and restored settings encompass:

- view parameters (center, scale, rotation, SRS)
- per-layerset selected or deselected states
- per-source and source layer selected or deselected states
- per-source opacity

Note: Persistence is purely based on local browser storage, which means it is private to a user's local browser. It also remains private for multi-user systems. There is no interaction whatsoever with the Mapbender login.

Not-supported map view parameters are:

- Dimension parameter values
- Source additions (via WMS Loader)
- Layer / entire source removals (via Layertree context menu)
- Source / layer reordering operations via Layertree (via drag&drop)
- States of per-layer featureinfo checkboxes

Configuration

Persistent map view is enabled on a per-application basis with a new checkbox under the "Base data" tab.

The screenshot shows the 'Base data' configuration tab in Mapbender. It contains the following fields and options:

- Title ***: Demo
- URL Title ***: mapbender_user.yml
- Thumbnail**: Select file screenshot.png
- Map engine ***: OpenLayers 6
- Description**: Full desktop application showing most of the standard Mapbender elements
- Persistent map state**: ☐ (highlighted with a red box)
- View**: Button in the top right corner

This change introduces a new column in the `mb_core_application` table and therefore requires `app/console doctrine:schema:update --force` to be run.

YAML-Definition:

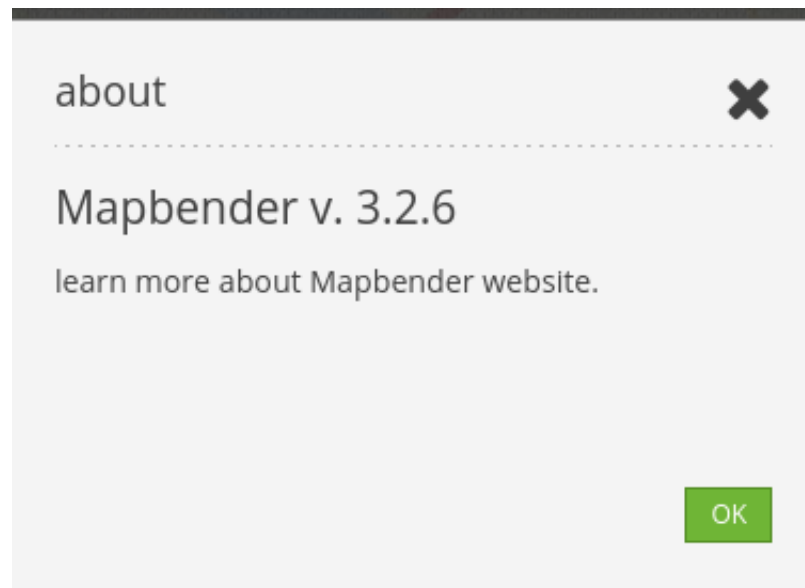
This template can be used to insert the element into a YAML application with a new `persistentView` entry on the top level. Omitting this entry is the same as setting it to false.

```
parameters:
  applications:
    mapbender_user:
      title: Mapbender Demo Map
      screenshot: screenshot.png
      published: true
      persistentView: true      # parameter to activate Persistent map view
      template: Mapbender\CoreBundle\Template\Fullscreen
```

3.6 Other

3.6.1 About Dialog

This element creates a [Button](#) which shows a simple about dialog, listing the version of your Mapbender installation. The button can be placed into the toolbar and the footer region.



Configuration

Add element

Content - About dialog

☒ Show label

Title About dialog

Tooltip About

Back

Add

Cancel

- **Show Label:** Enable/Disable About dialog text next to its icon (default: true).
- **Title:** Text indicated next to the about dialog icon.
- **Tooltip:** Text to use as a tooltip. Appears when hovering over the icon.

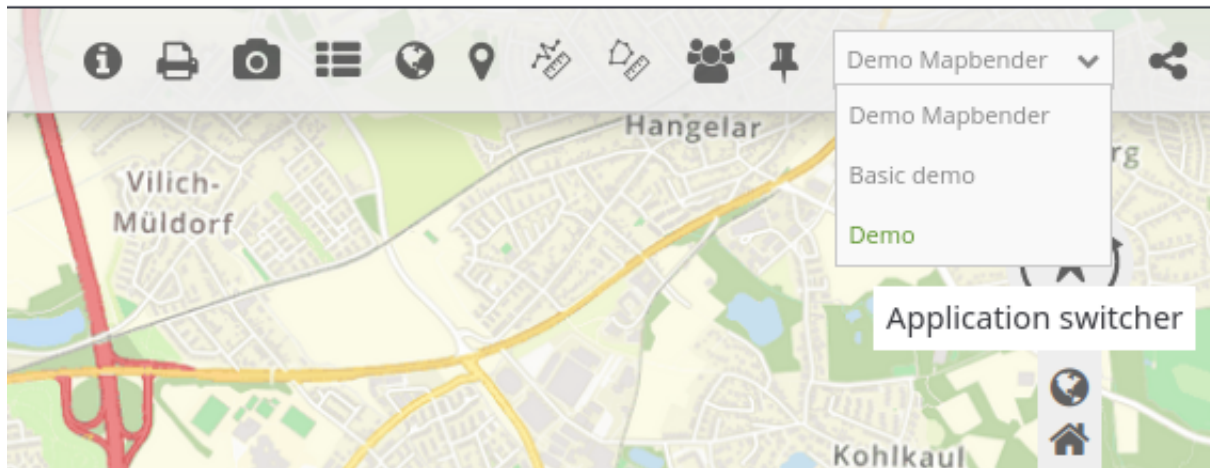
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: 'About Mapbender' # text indicated next to the about dialog icon.
tooltip: 'About Mapbender' # text to use as tooltip
label: true # false/true to label the button, default is true
icon: 'icon-about' # icon to display on button
```

3.6.2 Application Switcher

The Application Switcher provides a selectbox that refers to other Mapbender applications. You can switch from one application to another. The map extent will be preserved.



Configuration

Edit element

Title

Application switcher

Applications *

OL2 Demo
OL2 Mobile demo
OL2 Basic demo
Mobile demo
Basic demo
Demo
Demo Mapbender

☐ Open in new tab

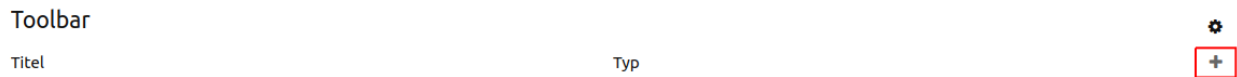
Save

Cancel

- **Title:** Title of the element. The title will be shown as tooltip on mouseover on the selectbox.
- **Applications:** Choose the applications that should be offered in the selectbox.
- **Open in new tab:** Define whether the new application should be opened in the same window (default) or in a new tab.

Application Switcher Configuration

First, you have to select the link element by clicking on the + - symbol in the Toolbar section in the Layouts tab.



After the selection of the ApplicationSwitcher element, the “Add element - ApplicationSwitcher” dialog box opens, where you can configure the element.

You can define a the field *Title*. This title will be displayed as tooltip on mouseover on the selectbox.

With the checkbox *Open in new tab*; you can define whether the new application should be opened in the same window (default) or in a new tab.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: Choose an Application           # Text will be displayed as tooltip
class: Mapbender\CoreBundle\Element\ApplicationSwitcher
applications: ["mapbender_user", "mapbender_user_basic"] # Define the applications for the
↪ApplicationSwitcher as array
open_in_new_tab: true # false/true open application in new tab
```

3.6.3 Button

The button element provides a push button widget. Some elements like [Legend](#), [Layertree](#), [FeatureInfo](#), [Line/Area Ruler](#) and [PrintClient](#) need a button to be displayed/activated if not defined in a frame.

Buttons optionally can be grouped, so that only one button in a group can be active at any given time. This is done by the group paramter. You can define a button that refers to a website or script using the click paramter. You can only choose features as target parameter, which have been added in the Map area or footer before.

Configuration

Add element

Content - Button

☒ Show label

Title	<input type="text" value="Button"/>
Target *	<input type="text" value="Choose one"/> ▼
Group	<input type="text"/>
Tooltip	<input type="text"/>
Icon	<input type="text" value="None"/> ▼

[Back](#)[Add](#)[Cancel](#)

- **Show label:** Enables or disables text (title) next to the button (default: true).
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Target:** Target element (Title(ID)) of the button. The target element will be triggered by clicking on the button (e.g. Print).
- **Group:** Adds the element to a group. Only one button of the group may be activated.
- **Tooltip:** Text, that will be indicated if the mouse hovers over the button for a longer time.
- **Icon:** Symbol of the button. Based on a CSS class.

Icons

For some symbols you can choose between two different types of icons:

- A symbol based on a graphic (e.g. "About"),
- A symbol based on a font (e.g. "About (Font Awesome)").

The latter are based on a `IconSet`, which is delivered with Mapbender as a module. We recommend to use the symbols from this library.

More information on that topic:

- <https://github.com/mapbender/icons>
- <http://rawgit.com/mapbender/icons/master/demo.html>

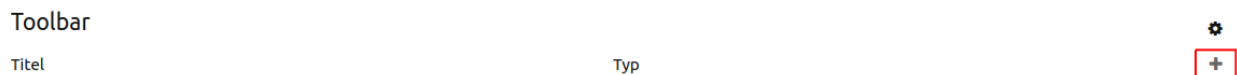
Configuration examples:

Depending on the purpose of the application, different buttons are required which provide different features. These can be integrated as needed. You can add buttons for features which are integrated in the Map area. For example, the legend or the line and/or area ruler can be activated or opened with buttons:

Button for the legend element

The legend is very helpful because it provides information about the Map area. In this user example, the legend is integrated in the Map area. You can find a description of how to configure the legend element in this documentation under [Legend](#). You can add a button for the legend by following these steps:

First, you have to select the button element by clicking on the + - symbol in the Toolbar section in the Layouts tab.



After the selection of the button element, the "Add element - Button" dialog box opens, where you can configure the button.

You can set the name of the legend button in the field *Title*. Here the title is "Legend". There will be no text displayed during hovering over the button, because no text was defined in the field *Tooltip*. You can choose from a variety of icons to set the icon for your button. In this example, the icon "Legend" was chosen.

Element hinzufügen

Toolbar - Button

☒ Beschriftung anzeigen

Title

Target * ▼

Group

Tooltip

Icon ▼

Zurück

Hinzufügen

Abbrechen

In the *Target* field you can choose the predefined feature which should be activated or opened by clicking on the button. The dropdown list contains all features which have been added to the Map area, the sidepane or the footer before. In this example, the option "Legend" was chosen.

Element hinzufügen

Toolbar - Button

☒ Beschriftung anzeigen

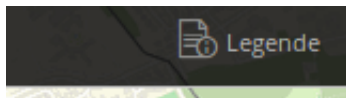
Title	Legende
Target *	Auswahl erforderlich
Group	FeatureInfo
Tooltip	Image export
Icon	Legend
	POI
	Print client
	WMS loader

Zurück

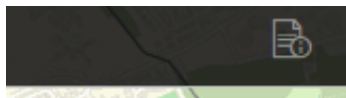
Hinzufügen

Abbrechen

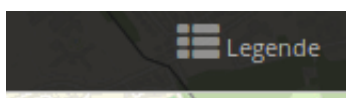
In this example, *Group* remains empty. In the Mapbender application, the button will look like this:



The title of the element is displayed next to the icon, because in the configuration dialog box the option “Show label” was checked. If the checkmark is not set, the button looks like this:



If you choose the *Icon* “Legend (Font Awesome)” instead of “Legend” and check the option “Show label”, the button would look like this:



Now, by clicking on the button, the legend will open in a dialog box.

Button for Line and Area Ruler

The line and area ruler also can be integrated in the application via buttons. These elements must have been added to the Map area, sidepane or footer in order to do so.

This example demonstrates how to create groups: Both buttons should be in one group, so that only one line or area ruler can be active and not both at the same time.

You can add this button, like the legend button, by clicking on the + -symbol in the application in the Layouts tab. The dialog box "Add element - Button" appears. For the line ruler feature it can look like this:

Add element

Sidepane - Button

☒ Beschriftung anzeigen

Title	Linienmessung
Target *	line ▼
Group	messen
Tooltip	Linien messen
Icon	Line ruler ▼

Back

Add

Cancel

In this example, the title (*Title*) of the button is "Line ruler". The element references to a *Target* called "line". This element was created beforehand with the feature Line/Area Ruler. You can find a description on how to create this feature under [Line/Area Ruler](#).

To group this button and the button for the area ruler, you have to put a group name in the field *Group*. In this example, the name of the group is "measure". You also have to add this group name to the button for the area ruler.

When you hover your mouse over the button, the text "measure line" will appear (*Tooltip*). The option "Line ruler" was chosen for *Icon*.

The button for the area ruler can be integrated the same way as the button for the line ruler. The dialog box for the configuration of this button can look like this:

Add element

Toolbar - Button

☒ Beschriftung anzeigen

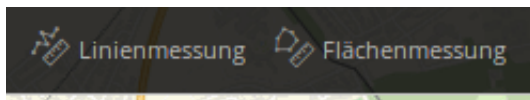
Title	Flächenmessung
Target *	area ▼
Group	messen
Tooltip	Fläche messen
Icon	Area ruler ▼

Back

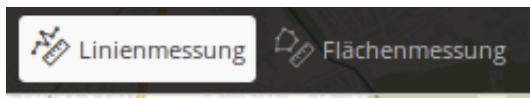
Add

Cancel

Attention should be paid to the field *Group* with the group name "measure". It has to be the same as the one of the button line ruler. Both buttons can look like this in the application:



If the button line ruler is active, it looks like this:



As soon as you click on the button area ruler, the feature line ruler is deactivated and the feature area ruler activated.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```

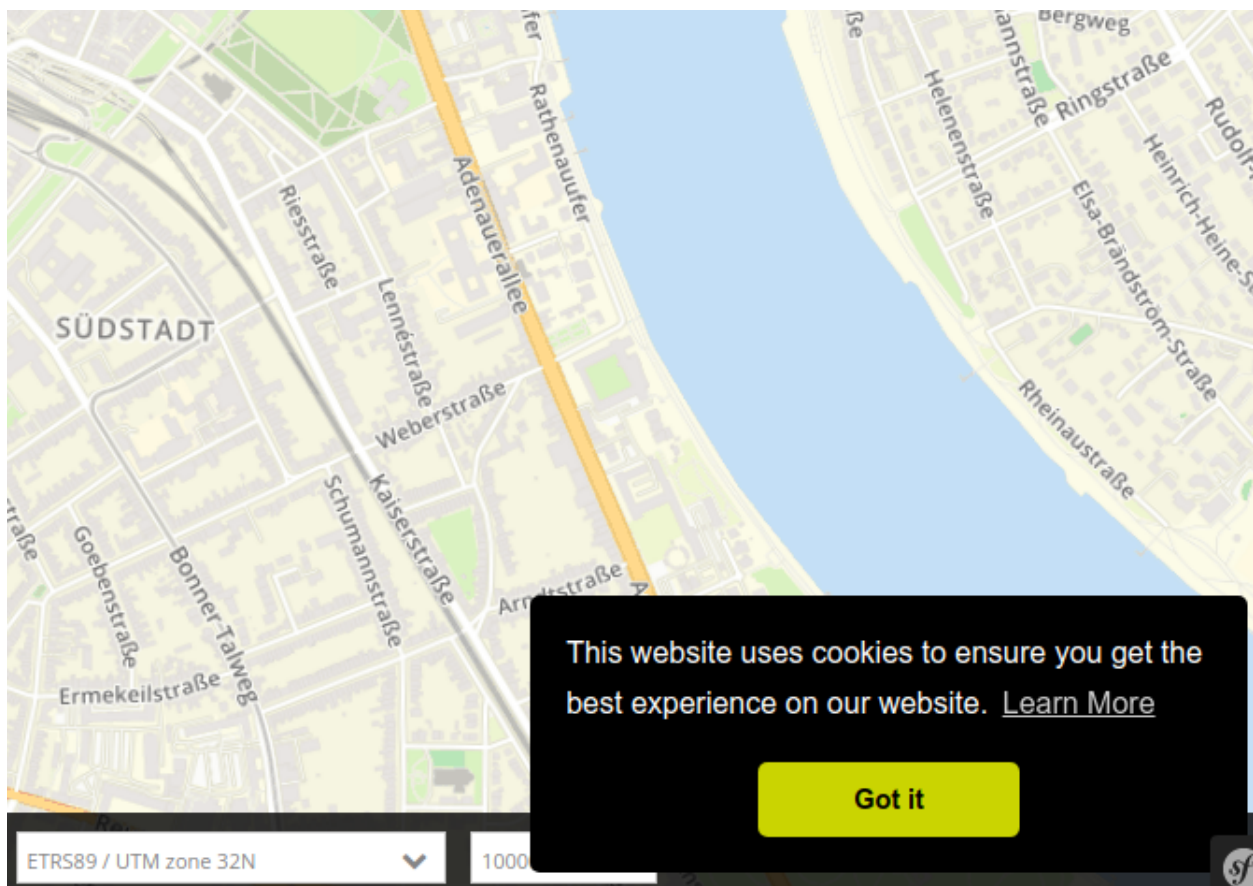
title:      # title
tooltip:   # text to use as tooltip
icon: ~    # icon CSS class to use
label: true # false/true to label the button, default is true
target: ~  # title (Id) of target element
click:     # refer to a website or script like http://mapbender.org
group: ~   # group to put the button into. Only one button per group can be active
action: ~  # method of target to call when button is activated
deactivate: ~ # method of target to call when button is deactivated

```

3.6.4 Cookie Banner

Applications support displaying a cookie banner which is customized via the Mapbender configuration file `parameters.yml`. We use the code from [Cookie Consent](#) without making an additional call to the internet.

The banner is displayed in any application on the first run.



After this banner is dismissed it doesn't appear again until you have deleted the cookie in your web browser. Mapbender depends on cookies and stores its PHP-session there.

Configuration

Configuration takes place in the `parameters.yml` file under `app/config/parameters.yml` and accounts for the whole Mapbender instance. Please add the parameter `mapbender.cookieconsent:` with the value `true` or `false`. If the parameter is missing or the value is set to `false`, the banner will not be displayed in your applications.

Example:

```
#Mapbender Cookie Consent Message
mapbender.cookieconsent: true
```

Tip: After changing these parameters you have to clean the contents of the cache-directory (`app/cache/``*`).

3.6.5 Link

The link element provides a button widget that refers a defined link like a website or script.

Configuration

Edit element

☒ Show label

Title

Mapbender Website

Tooltip

Visit the Mapbender Website

Icon

About



Target URL *

<https://mapbender.org>

Save

Cancel

- **Show label:** Enables or disables text (title) next to the button (default: true).
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Tooltip:** Text, that will be indicated if the mouse hovers over the button for a longer time.
- **Icon:** Symbol of the button. Based on a CSS class.
- **Target URL** Reference to a website or a script.

Icons

For some symbols you can choose between two different types of icons:

- A symbol based on a graphic (e.g. “About”),
- A symbol based on a font (e.g. “About (Font Awesome)”).

The latter are based on a [IconSet](#), which is delivered with Mapbender as a module. We recommend to use the symbols from this library.

More information on that topic:

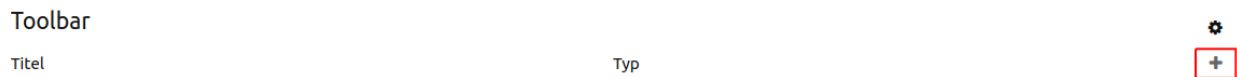
- <https://github.com/mapbender/icons>
- <http://rawgit.com/mapbender/icons/master/demo.html>

Example

It is possible to create and adjust different buttons with different functions. Buttons can refer to features which are included in the Map area. For example, it is possible to create a Legend button or Line- and/or Area Ruler buttons:

Link to a Webpage

First, you have to select the link element by clicking on the + - symbol in the Toolbar section in the Layouts tab.



After the selection of the link element, the “Add element - Link” dialog box opens, where you can configure the element.

You can set the name of the link button in the field *Title*. This title will be displayed as label next to the icon if *Show label* is active.

In the field *Tooltip*, you can define a text that will be displaced as tooltip during hovering over the button. You can choose from a variety of icons to set the icon for your link button.

Edit element

☒ Show label

Title	Mapbender Website
Tooltip	Visit the Mapbender Website
Icon	About ▼
Target URL *	https://mapbender.org

Save

Cancel

YAML-Definition:

This template can be used to insert the element into a YAML application.

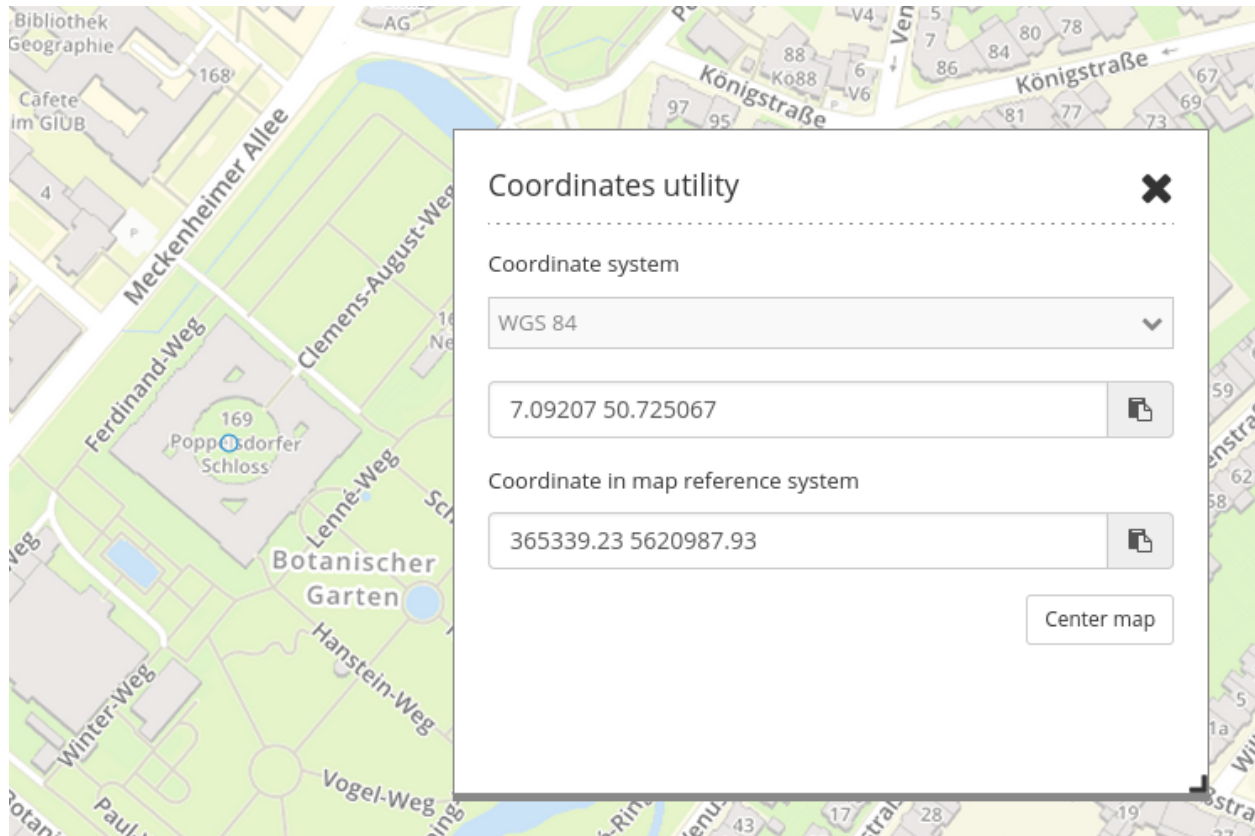
```
title: Link # title
class: Mapbender\CoreBundle\Element\Button
tooltip: Visit the Mapbender Website # text to use as tooltip
icon: iconInfoActive # icon CSS class to use
label: true # false/true to label the button, default is true
click: https://mapbender.org # refer to a website or script
```

3.6.6 Coordinates Utility

The element *Coordinates Utility* allows two different things:

1. Zoom to a given point coordinate
2. Show the clicked point of the map

Both functions can be accessed with one user interface:



Both cases allow a dynamic coordinate transformation so that also points in a different coordinate system can be used. Additionally, you have the possibility to copy the coordinates into the clipboard. The element can be configured in the Mapbender backend as a dialog with a button or as an element directly into the Sidepane.

Configuration

Add element

Sidepane - Coordinates utility

Title

Coordinates utility

Srs list

Zoom-Level *

6

☒ Add map's srs list

Back

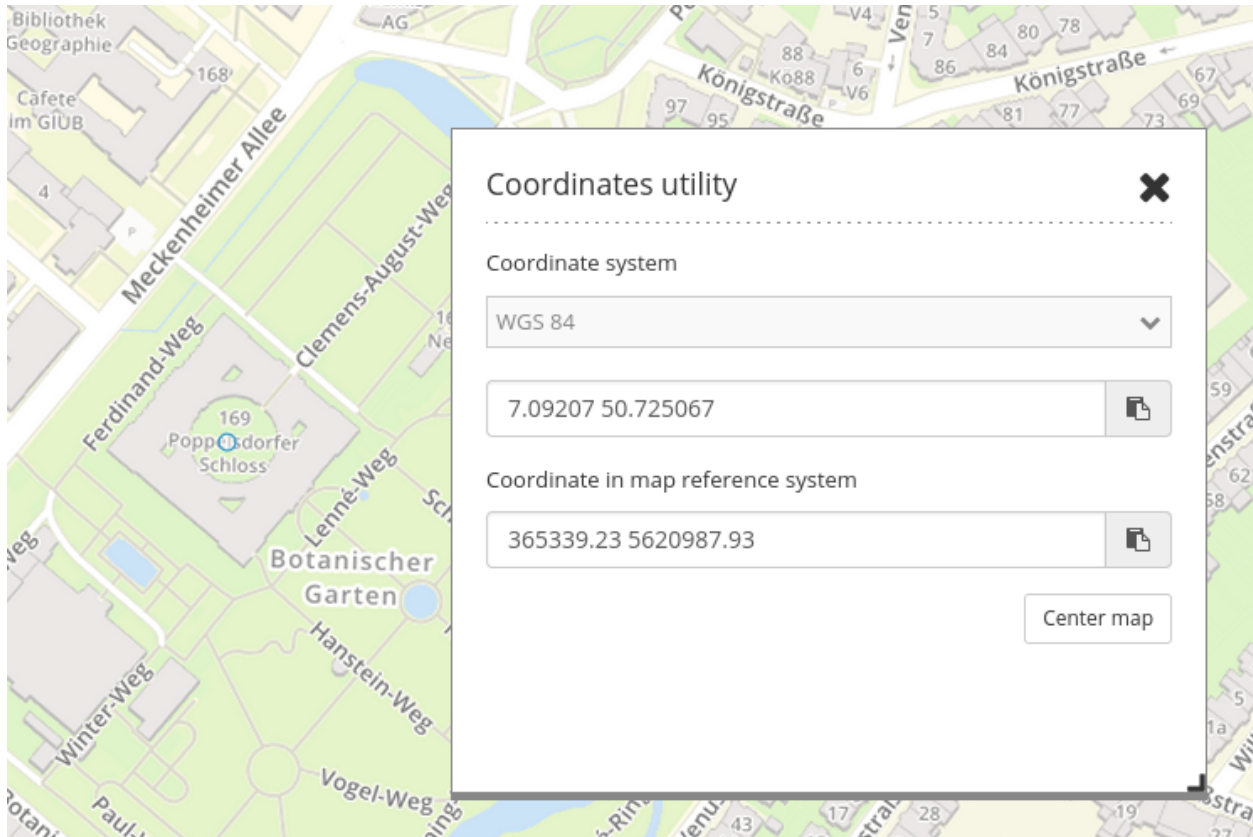
Add

Cancel

- **Title:** Title of the element.
- **Srs List:** You can define additional SRS to which the tool has to transform the coordinates. This list can be left empty.
- **Zoom-Level:** Zoom level of the map (default: 6)
- **Add map's srs list:** The supported coordinate systems defined in the [Map element](#) are automatically used. These coordinate systems, defined in the map, are also used by the [SRS Selector](#) (default: true).

If you define Coordinates Utility as a dialog, you need a [Button](#) that you place in the Toolbar.

Using the tool



Get Coordinate:

- If Coordinates Utility is opened as a dialog, the map reacts on a click. Click into the map and the click-coordinate is displayed in the dialog.
- Change the coordinate system with the dropdown-list. The click-coordinate is displayed in the given coordinate system.
- The last line therefore shows the click-coordinate in the original coordinate system of the map.
- The button at the end of each text-field allows to copy the coordinate directly into the clipboard.
- If Coordinates Utility is opened via the Sidepane, the button *Coordinate search* will appear that allows switching between a pan-cursor and a coordinate-cursor.

Zoom to coordinate:

- The text field can be used to edit your own coordinates. They must be given in the coordinate system that is chosen in the upper dropdown-list.
- With a click on the **Center map** button, the map zooms to the given coordinate and shows the position with an orange symbol.

YAML-Definition

This template can be used to insert the element into a YAML application.

```
coordinatesutility:
  title: 'Koordinaten Utility'
  class: Mapbender\CoordinatesUtilityBundle\Element\CoordinatesUtility
  type: element
  target: map
  srsList:
    -
      name: 'EPSG:31466'
      title: '31466'
    -
      name: 'EPSG:31468'
      title: '31468'
    -
      name: 'EPSG:25833'
      title: '25832'
    -
      name: 'EPSG:4326'
      title: '4326'
      addMapSrsList: true
```

3.6.7 Copyright

The copyright element can be used to show simple text or HTML in a window dialog. It can be opened automatically on application start. It is also possible to integrate links and images into the window (see [HTML](#)).



Configuration

Edit element

☒ Auto-open

Title

Copyright

Dialog width

400

Dialog height

Automatic

Content *

```
<p><b>Mapbender</b></p>
<p>Mapbender is an open source WebGIS solution.</p>
<p>Read more at <a href="https://mapbender.org/"
target="_blank">https://mapbender.org</a>
```

[Save](#)[Cancel](#)

- **Autoopen:** Enable or disable autoopening of the copyright window while starting the application (default: off).
- **Title:** Title of the element. It will be indicated next to the button.
- **Tooltip:** Text used as a tooltip. It will be displayed when hovering with the cursor over the button. It also used as a header in the copyright window.
- **Popup width:** Width of the Popup window (default: 300).
- **Popup height:** Height of the Popup window (default: 170).
- **Content:** Content of the copyright window, displayed when clicking on the button (or autoopened by starting the application, if enabled).

Refer to a twig file

You can refer to a twig file in the Content area. Please note that the twig file has to contain valid HTML.

Edit element

☐ Auto-open

Title

News

Dialog width

500

Dialog height

Automatic

Content *

```
{% include 'WorkshopDemoBundle::news.html.twig' %}
```

Save

Cancel

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
class: Mapbender\CoreBundle\Element\Copyright
title: "Copyright"           # Title of the element
popupWidth: 300
popupHeight: 170
tooltip: "Copyright"        # Text to use as tooltip
content: "Lorem ipsum"      # Edit the text you want to display as copyright text
autoOpen: true              # Automatically open the dialog when you start the application
```

3.6.8 Dimensions handler

The dimensions handler can be used to integrate WMS services with a time dimension. WMS-Time services are registered as a normal WMS data source. If a dimension is specified in the service, it is displayed in the layer metadata.

```
Dimension:
  name: 'time', units: 'ISO8601', unitSymbol: '', default: '2018-01-01', multipleValues: '',
  nearestValue: '1', current: '', extent: '2014-01/2018-01/P1M'
```

WMS Dienst mit Zeitangabe (WMS-T)

Name: wms_time
Titel: WMS Dienst mit Zeitangabe (WMS-T)
Beschreibung: WMS Dienst mit Zeitangabe (WMS-T) - für Demo- und Testzwecke nutzbar, letzte Änderung im Januar 2018
LatLon: EPSG:4326 7.08705, 50.7373, 7.09439, 50.7428
Grenzen:
BoundingBox: EPSG:4326 50.7373,7.08705,50.7428,7.09439
 EPSG:31466 5.62298e+06,2.57677e+06,5.62361e+06,2.5773e+06
 EPSG:31467 5.62415e+06,3.36504e+06,5.62478e+06,3.36557e+06
 EPSG:25832 365018,5.62234e+06,365551,5.62297e+06
 EPSG:25833 -58021.7,5.65047e+06,-57438.1,5.65115e+06
SRS: EPSG:4326, EPSG:31466, EPSG:31467, EPSG:25832, EPSG:25833

Punkte nach Zeit

Name: punkte
Titel: Punkte nach Zeit
kaskadiert: 0
LatLon: EPSG:4326 7.08705, 50.7373, 7.09439, 50.7428
Grenzen:
BoundingBox: EPSG:4326 50.7373,7.08705,50.7428,7.09439
 EPSG:31466 5.62298e+06,2.57677e+06,5.62361e+06,2.5773e+06
 EPSG:31467 5.62415e+06,3.36504e+06,5.62478e+06,3.36557e+06
 EPSG:25832 365018,5.62234e+06,365551,5.62297e+06
 EPSG:25833 -58021.7,5.65047e+06,-57438.1,5.65115e+06
SRS: EPSG:4326, EPSG:31466, EPSG:31467, EPSG:25832, EPSG:25833
Styles: Name: default
 Titel: default
 Beschreibung:
 Legenden-URL: 75X20 image/png http://wms2016.wherogroup.com/cgi-bin/mapserv?map=/data/umn/germany/wms_time.map&version=1.3.0&service=WMS&request=GetLegendGraphic&sld_version=1.1.0&layer=punkte&format=image/png&STYLE=default
 Stylesheet-URL:
 Style URL:

```
Dimension:
  name: 'time', units: 'ISO8601', unitSymbol: '', default: '2018-01-01 12:00:00', multipleValues: '', nearestValue: '1',
  current: '', extent: '2000-01-01/2030-01-01'
```

WMS-T are inserted almost like WMS in the layersets, with the exception that the time parameter still has to be activated. If it is not activated, the dimensions of the service are ignored and the default value is used when calling the layer in the map.

If a service supports a time dimension, the instance displays the “Dimensions” button. By clicking on this button, the supported time parameters are displayed and time values can be activated by clicking the checkbox. After another click on the button, the detailed form opens, in which the usage can be further defined. Here you can further restrict the values from the WMS service. To set up the service, the following definitions of time parameters are necessary:

- **Query type:** multiple, nearest, current
- **Name:** value TIME (name=time)
- **Units:** format for temporal dimensions (ISO 8601:2000)
- **Unit symbol:**
- **Default:** default time
- **Extent (extent slider):** Supported extent for the time dimension

Dimensions

time ☒ active

☐ multiple ☐ nearest ☐ current

Name time

Units ISO8601

Unit symbol

Default 2018

Extent 2006/2018/P2Y

Extent slider

The element supports the following time variables:

- Single time parameter
- List of times
- Time interval

Controlling time parameters

There are two ways to control the time of the WMS. On the one hand, each service with an active time parameter can be controlled via the context menu of the layer in the layertree. In addition, a central slider can be integrated, which can be displayed in any area of the application. The slider can be used to combine several layers with the same extent to control them centrally.

Timeslider in context menu

A timeslider can be integrated via the layertree as an option in the context menu of the layer. To do this, the “Dimension” option must be activated in the *Layertree* element.

Edit element

☒ Show base sources ☐ Autoopen

Title: Layer tree

Menu:

- Remove layer
- Opacity
- Zoom to layer
- Metadata
- Dimension**

☐ Hide visibility by folders ☐ Hide info

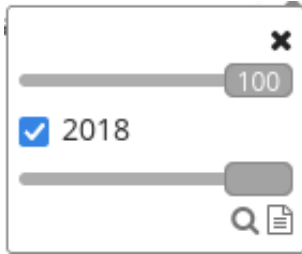
☐ Thematic layer

Themes: main ☒

☒ Allow Reorder at TOC

Save Cancel

After activation in the layertree, a time slider appears in the context menu. To use the feature, the element must first be activated via the checkbox. Then it's possible to alter the timeslider with the cursor.



Timeslider as element

The layers can also be controlled by the dimensions handler element via a central slider. The dimensions handler can be integrated into sidepane, toolbar or footer. The configuration of the dimensions handler is done in three steps:

- **Creating the element:** First, the element needs to be created. To do that, integrate the Dimensions handler as new element in the Layouts tab of your Mapbender application.
- **Defining a dimensionsset:** Next, define a dimensionsset. You can create a new set in the element via the "+" button. Type in a title for the set and select a Group.
- **Defining the slider:** At last, you can select the layer instances for the dimensionset that you want to control via the slider. Multiselect is supported, but only instances that have the same extent can be combined with each other. In addition, after the first selection of an instance, a slider appears. There you can restrict the extent for the time display.

When an instance is selected, all instances that do not conform to this default are no longer selectable. Finally, save the configuration to add the Dimensions handler element to the Mapbender application.



Configuration

Add element

Top toolbar - Dimensions handler

Title

Dimensions handler

Tooltip

Dimensionsets *

+

Title *

Set 2008-2012

Group *

257-0-time-interval

Extent *

2008/2012/P2Y - 2012

Back

Add

Cancel

- **Title:** Title of the element
- **Tooltip:** Labeling of the element on mouseover
- **Dimensionset:** Group for display of time extent with a title (**Title**), layergroup (**Group**) and time dimension extent (**Extent**)

The following section lists the YAML definitions for the element and the inclusion in the service and lev-eltree.

Element configuration

```
target: ~                                # ID of the map element
dimensionsets:
  title: Time
  group:
    dimension:      # Dimension specification as in the layer configuration
      origextent: 2014-01/2018-01/P1M
      active: null
      type: interval
      name: time
      units: ISO8601
      unitSymbol: null
      default: 2017-11
      multipleValues: true
      nearestValue: true
      current: true
      extent: 2014-01/2017-11/P1M
```

Layer configuration

```
dimension:      # Dimension specification in layer configuration
-   name: time
    units: ISO8601
    unitSymbol: null
    default: 2018-01
    multipleValues: false
    nearestValue: true
    current: false
    extent: 2014-01/2018-01/P1M
```

Layertree configuration

```
title: Layertree
class: Mapbender\CoreBundle\Element\Layertree
configuration:
  [...]
  menu:
    - [...]
    - dimension
```

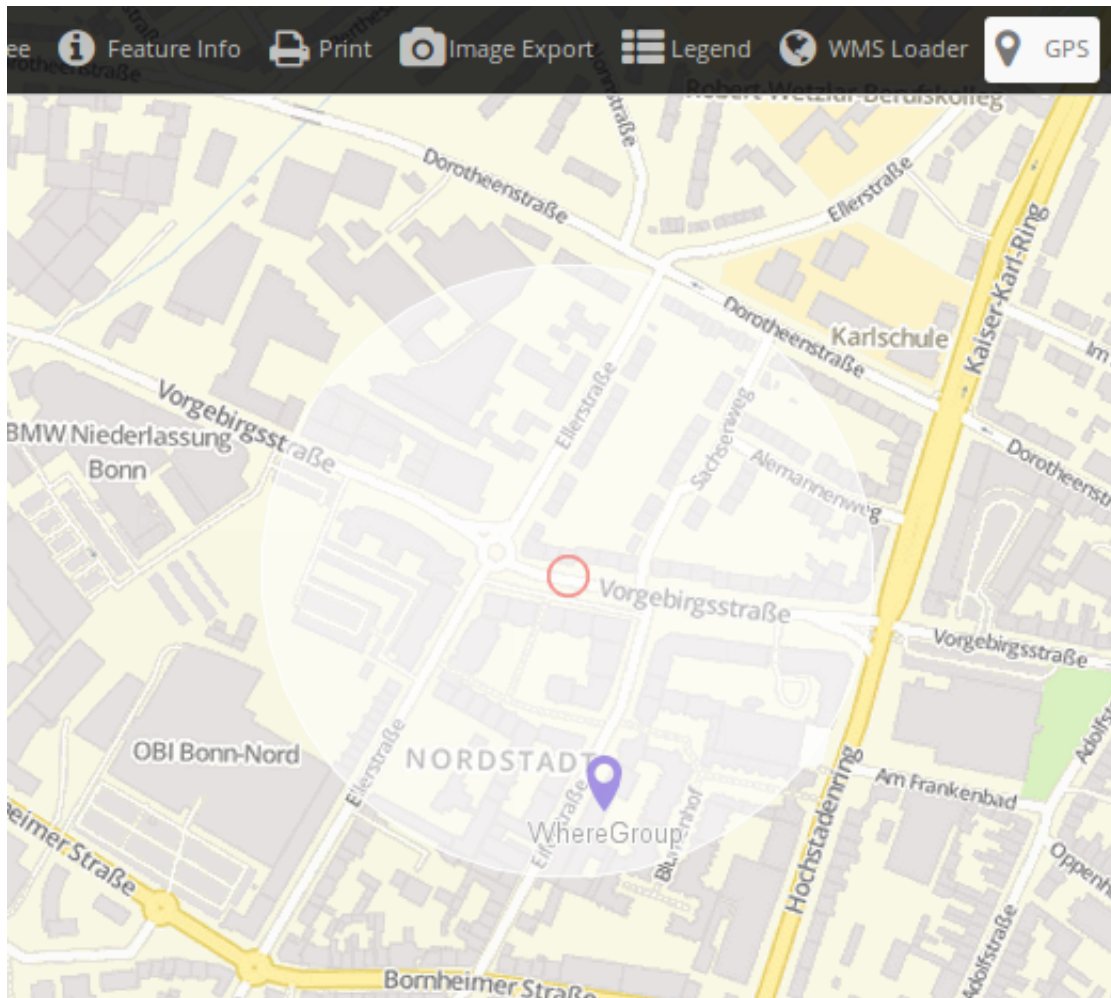
3.6.9 GPS Position

This element provides a button to navigate to your current position and display a symbol at that position. The scale will not be changed until you activate zoom to accuracy (zoom to accuracy on first position).

The element is built upon the [Geolocation-API](#) by the W3C. To validate that your browser supports this functionality please take a look at the [Can I Use](#) page. The element uses the High Accuracy Parameter that forces the positioning via GPS. If your device is shipped with a GPS-receiver and if it is activated, the positioning is more accurate. Otherwise the WIFI access points are used for positioning.

The midpoint shows the probable position of the device, the outer circle the accuracy of the positioning, that means the region where the position is probably to find.

Compatibility: Internet Explorer and MS Edge deliver without a GPS-reciever at the machine imprecise information. This behaviour is also observable with other applications.



Configuration

Add element

Sidepane - GPS Position

☒ Show label☐ Autostart

Title GPS Position

Tooltip Display GPS Position

Icon GPS (FontAwesome) ▼

Average 1

☐ Follow☒ Center on first position☒ Zoom to accuracy on first position

Back

Add

Cancel

- **Show label:** Switches the button label on/off (default: on).
- **Autostart:** Starts element when opening the application (default: off).
- **Title:** Title of the element.
- **Tooltip:** This text will be displayed during hovering over the element with the cursor.
- **Icon:** Icon to display on button.
- **Average:** Calculates the average of the last at parameter average defined amount of received GPS coordinates (default: 1).
- **Follow:** Refreshes the map for every received GPS position received, only use with WMS in tiled mode (default: off).
- **Center on first position:** Centers map only on first received GPS position (default: on).

- **Zoom to accuracy on first position:** Zoom map according to first received gps position accuracy (default: on).

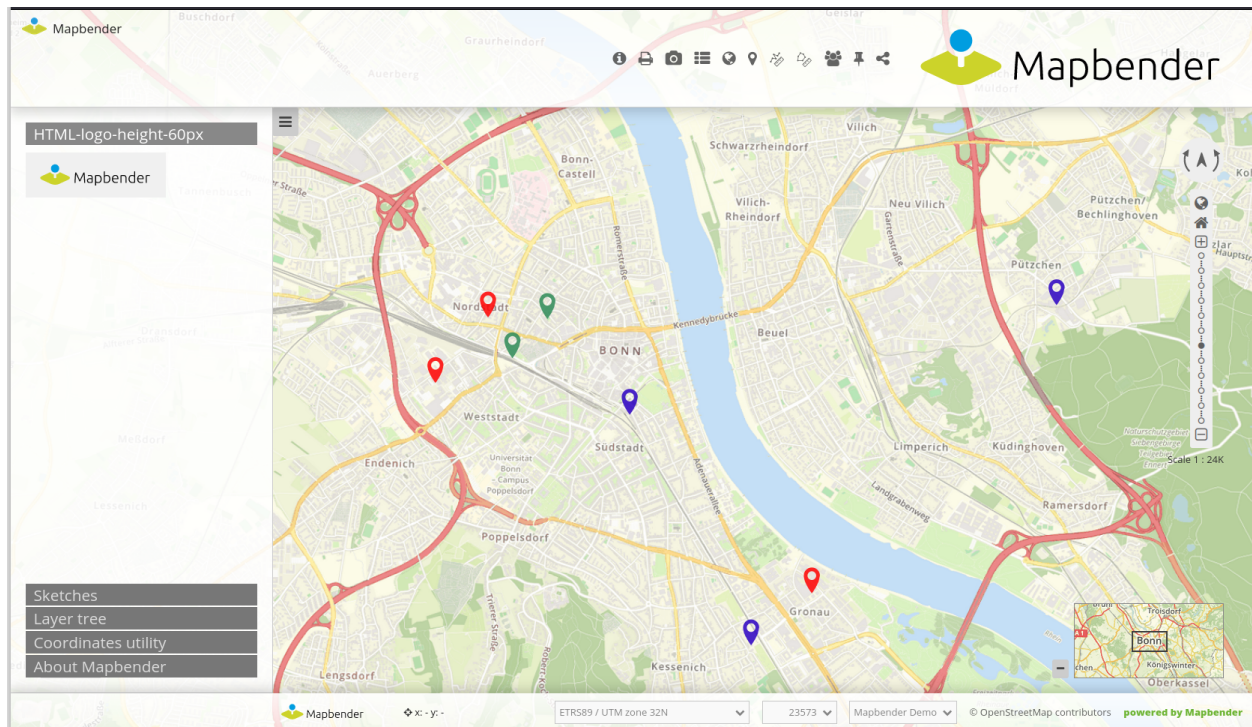
YAML-Definition:

This template can be used to insert the element into a YAML application. The element is placed as a button into the toolbar.

```
class: Mapbender\CoreBundle\Element\GpsPosition
label: true # true/false to label button (default: true)
autoStart: false # true/false (default: false)
title: GPS-Position # title of the button
tooltip: GPS-Position # text to use as tool tip
icon: gpsposition # icon to display on button
target: map # Id of Map element to query
average: 1 # calculates the average of the last at parameter average defined
↪ amount of received GPS coordinates, default 1
refreshinterval: 5000 # refresh interval in ms, default is 5000 ms
follow: true # default false, true refreshes the map for every received GPS
↪ position received, only use with WMS in tiled mode
centerOnFirstPosition: true # center map only on first received gps position
zoomToAccuracy: false # zoom map according to received gps position accuracy
zoomToAccuracyOnFirstPosition: true # zoom map according to first received gps position accuracy
```

3.6.10 HTML

This element allows the integration of HTML anywhere in an application. Figures, links or e.g. texts can be inserted. The following illustration shows the integration of a Mapbender logo in footer, toolbar and sidepane:



The following variables can be integrated in HTML:

- “application” (Entity Application)
- “entity” (Entity HTML-Element)

Configuration

This element has to be integrated in the backend.

Add element

Toolbar - HTML

Title

HTML

Content

Classes

html-element-inline

Back

Add

Cancel

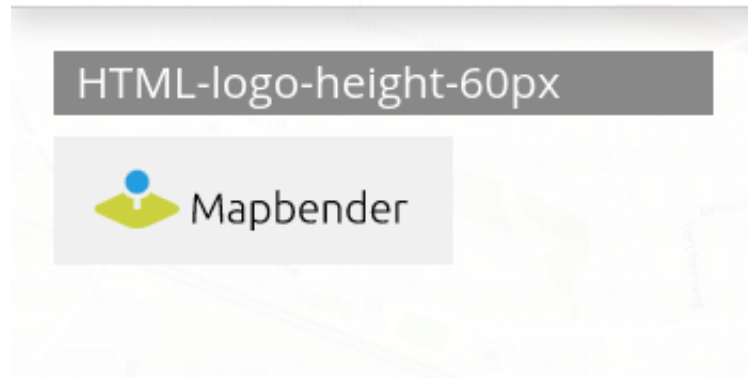
- **Title:** Title of the element. The title will be listed in “Layouts”.
- **Content:** Content of the HTML-element. The variables “application” and “entity” are available.
- **Classes:** You can refer to a CSS class for the HTML element. The class can be defined in the css editor.

Configuration examples

Insert picture:

In this example, the Mapbender Logo was integrated in the sidepane (``). It can be adjusted with a predefined height (`height='60px'`), background color and transparency (`background-color:rgb(240, 240, 240, 0.9)`) as well as a certain padding (`padding:10px`). These configurations were defined as styling through `style=`.

```
<img src='https://mapbender.org/fileadmin/mapbender/resources/images/logos/Mapbender-Logo.svg'
height='60px' style='background-color:rgb(240, 240, 240, 0.9); padding:10px'>
```



Insert picture with link:

A linkage was added to the previously inserted logo. It references to the Mapbender website (`href='https://mapbender.org'`) if the logo is clicked on. The link is opened in a new tab because of `target='_blank'`.

```
<a href='https://mapbender.org' target='_blank'>
<img src='https://mapbender.org/fileadmin/mapbender/resources/images/logos/Mapbender-Logo.svg'
height='60px' style='background-color:rgb(240, 240, 240, 0.9); padding:10px'></a>
```

Variables & HTML-Element

Several variables can be integrated in the HTML-Element.

- Variable: **“application.title”**

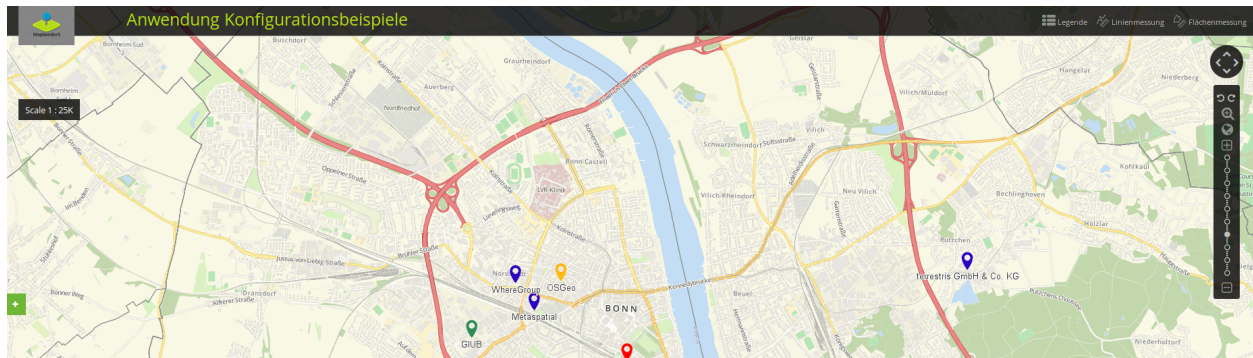
This variable allows the integration of the application title. In the example, this corresponds to “Konfigurationsbeispiele”.

The HTML-Code could look like this:

```
<b><span style="font-size:25px;color:#b6dd18;margin-right:50vw"> Anwendung {{ application.title }} </
<span></b>
```

The application title is defined through `{{ application.title }}`. The term “Anwendung” is an addition and will display independently from the actual title. The style-block (`style=`) defines font size (`font-size:25px`), font width (``), font color (`color:#b6dd18`) as well as position (`margin-right:50vw`) of the title.

The result for the configuration example looks like this:

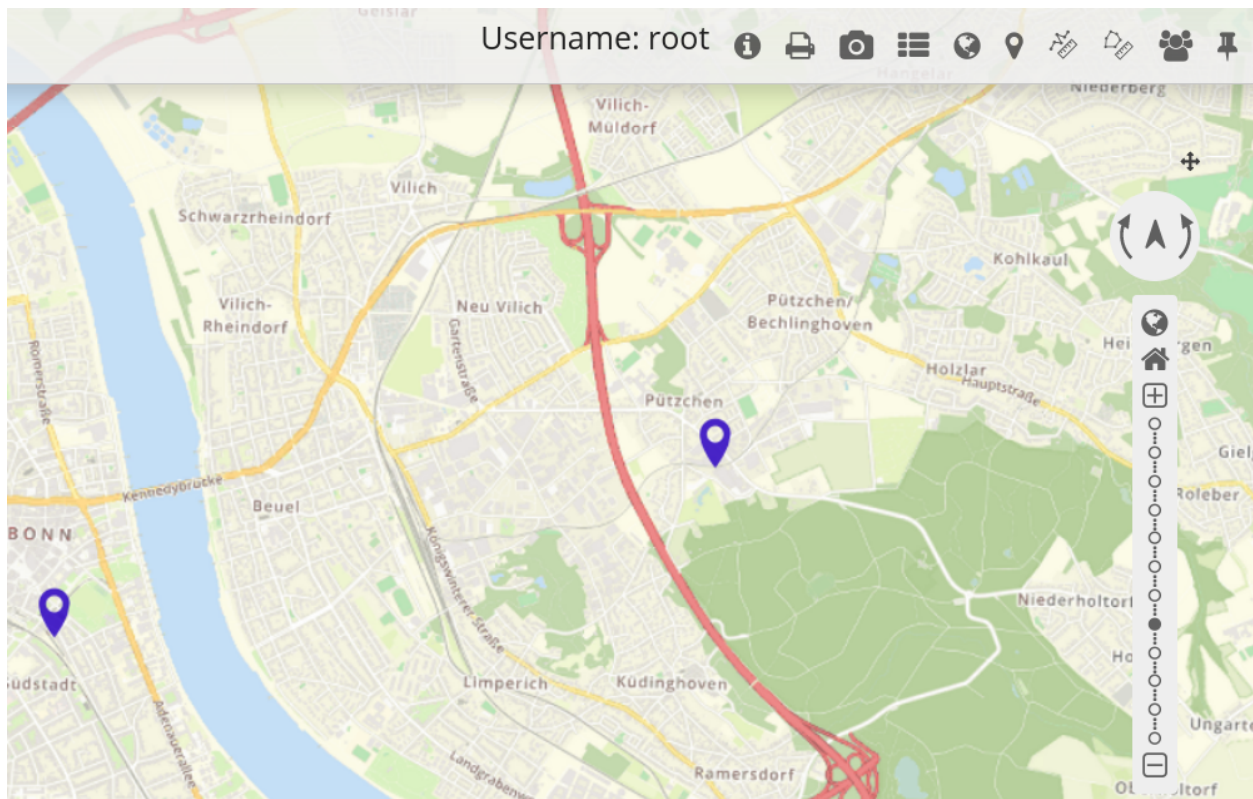


- Variable: **app.user.username**

This variable displays the name of the active user:

```
<p>Username: {{ app.user.username }}</p>
```

In this example, the user name is displayed in the toolbar:

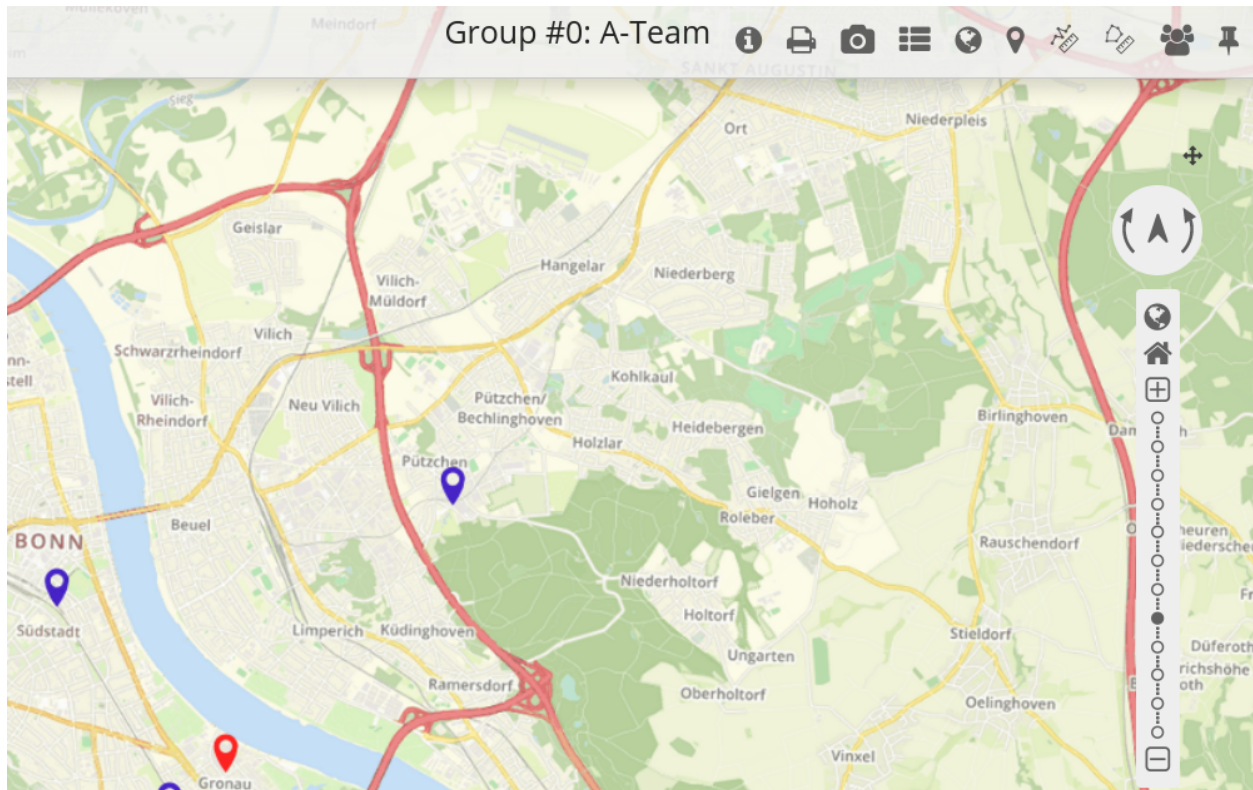


- Variable: **group.title**

The group of a user cannot be defined in a single expression, because Twig only supports the map-filter in higher versions. In order to integrate this variable, a loop will be used:

```
{% for index, group in app.user.groups %}
  <p>Group #{{ index }}: {{ group.title }}</p>
{% endfor %}
```


In this example, index and group name are displayed in the toolbar:



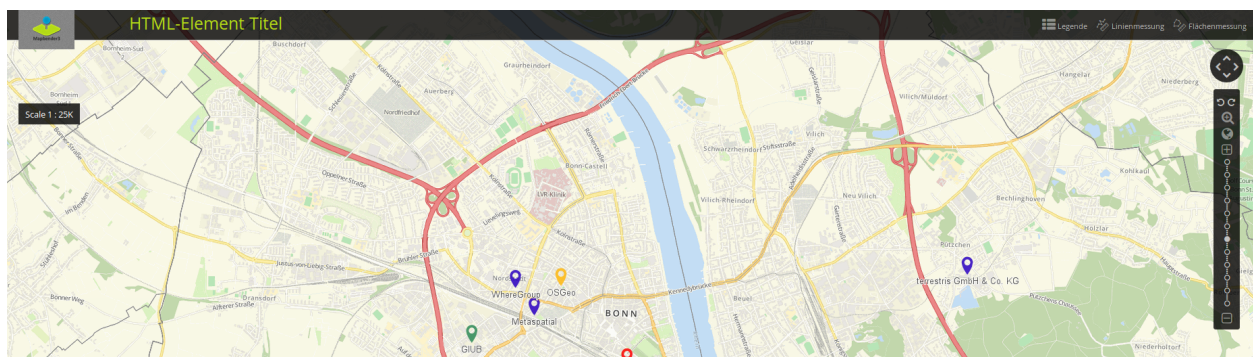
- Variable: **“entity”**

The variable `{ entity }` displays the ID and `{ entity.title }`, which displays the title of the HTML-element.

In the following, the variable `{ entity.title }` was integrated with the text addition “HTML-Element”. The Styling parameters correspond to those of the example with *application.title*. Text additions, variables and position were simply adjusted for *entity.title*.

```
<b><span style="font-size:25px;color:#b6dd18;margin-right:60vw"> HTML-Element
{{ entity.title }} </span></b>
```

These variables could look as follows:



Refer to a twig file

You can refer to a twig file in the Content area. Please note that the twig file has to contain valid HTML.

Edit element

Title	<input type="text" value="News"/>
Content	<pre>{% include 'WorkshopDemoBundle::news.html.twig' %}</pre>
Classes	<input type="text" value="html-element-inline"/>

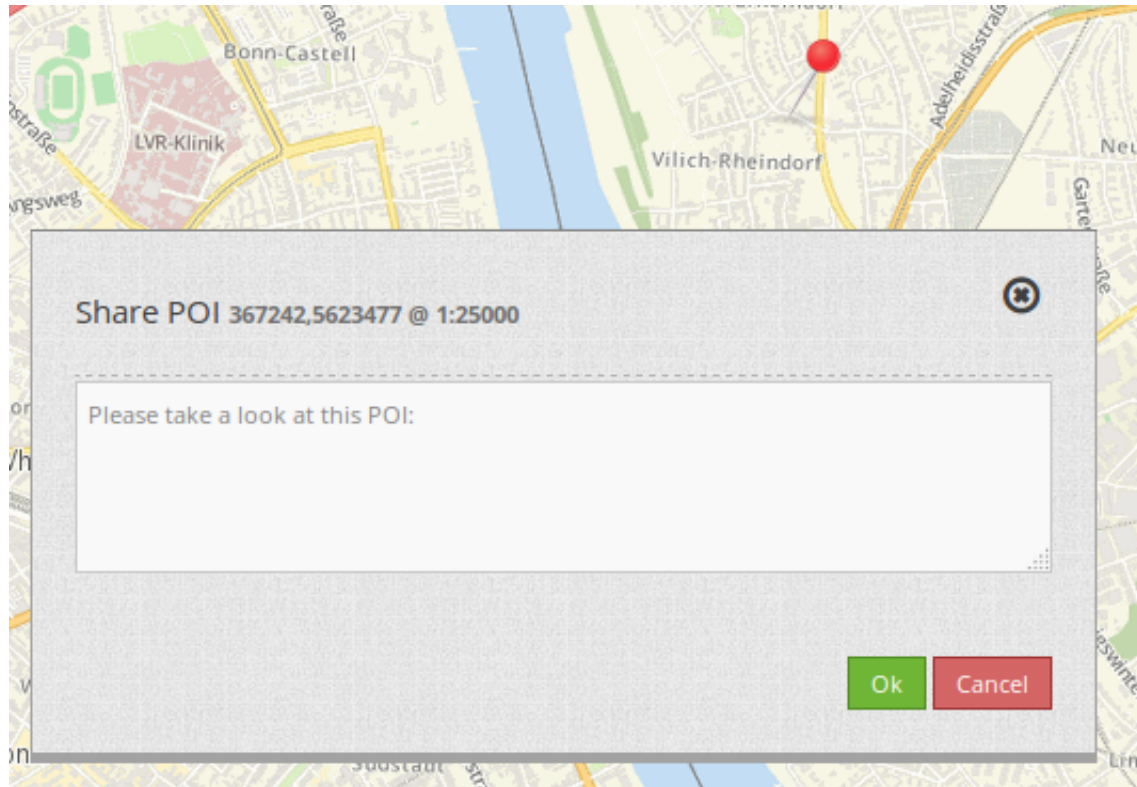
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
title: 'HTML-Element'  
class: Mapbender\CoreBundle\Element\HTML-Element  
content: <p>Hello, World!</p><p>Application: {{ application.title |trans }}</p> # The variables  
→ "application" and "entity" are available.  
classes: my-special-css-class
```

3.6.11 POI

Generate POI-URLs (aka meeting points) suitable for e-mail. The generated point is projected and displayed in the coordinate system of the map.



Configuration

Add element

Toolbar - POI

☒ Use Mailto

Title	<input type="text" value="POI"/>
Body *	<input type="text" value="Please take a look at this POI"/>
Gps	<input type="text" value="GPS Position"/> ▼

Back

Add

Cancel

- **Use Mailto:** Sends POI by e-mail.
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Body:** Defines text to display.
- **GPS Position:** Defines GPS Position in the map.

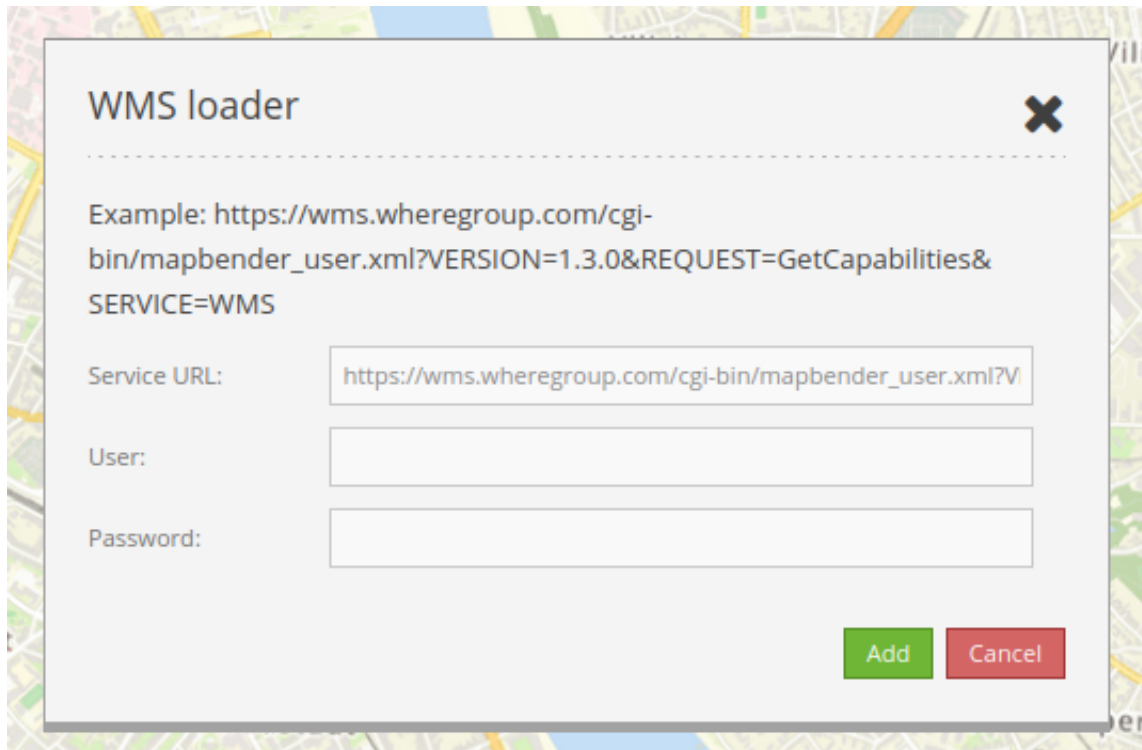
YAML-Definition:

This template can be used to insert the element into a YAML application.

```
target: map                                # only map-element is possible
body: 'Please take a look at this POI'      # define a text to display
```

3.6.12 WMS Loader

Opens a dialog in which a WMS can be loaded via the getCapabilities request. It is possible to load WMS 1.1.1 and WMS 1.3.0.



Configuration

Edit element

☐ Auto open ☐ Split layers

Title

WMS loader

Default format *

image/png



Default info format *

text/html



Save

Cancel

- **Auto open:** Opens the element when application is started (default: false).
- **Split layers:** Splits layer on load of the service (default: false).
- **Title:** Title of the element. The title will be listed in “Layouts” and allows to distinguish between different buttons. It will be indicated if “Show label” is activated.
- **Default format:** image/png, image/gif, image/jpeg (default: image/png).
- **Default info format:** text/html, text/xml, text/plain (default: text/html).

You need a [Button](#) to show this element.

YAML-Definition:

This template can be used to insert the element into a YAML application.

```
target: ~                # Id of Map element to query
tooltip: 'WMS Loader'    # text to use as tooltip
autoOpen: false          # true/false open when application is started (default: false)
defaultFormat: 'image/png' # image/png, image/gif, image/jpeg (default: image/png)
defaultInfoFormat: 'text/html' # text/html, text/xml, text/plain (default: text/html)
splitLayers: false       # split layer on load of the service (default: false)
```

How to add a WMS by defining a link

You can add a WMS to Mapbender by defining a link, e. g. in your *FeatureInfo* or your search results. The link has to look like this:

```
<a href="#"
mb-action="source.add.wms" mb-layer-merge="1" mb-wms-merge="1"
mb-wms-layers="Gewaesser,Fluesse"
mb-add-vendor-specific="bplan=123"
mb-url="http://wms.wherogroup.com/cgi-bin/germany.xml?VERSION=1.1.1&REQUEST=GetCapabilities&SERVICE=WMS"
↪">load service</a>
```

```
mb-action="source.add.wms"          # defines action to add a WMS
mb-wms-merge="1"                    # adds the WMS only once, if WMS is already part of the application
↪it will use the WMS which is there (default: 1)
mb-layer-merge="1"                  # activate the layers passed mb-wms-layers and do not disable the
↪layers which are already active (default: 1)
mb-wms-layers="Gewaesser,Fluesse"   # defines the layers to be activated, _all activates all layers,
↪default all layers are deactivated
href oder mb-url                    # refer to the WMS getcapabilities URL
mb-add-vendor-specific="bplan=123" # define a vendor specific that will be added to the requests
↪(version 3.2.9 and up)
```

BACKEND

The backend is the administration interface of Mapbender: It allows the management of applications and data sources as well as securing them through users and groups. To install and set up Mapbender, please visit the [Installation](#) pages or the [Quickstart](#). In its local standard configuration, the backend can be accessed via <http://localhost/mapbender/app.php>. After a successful login with the root user, the following pages are available:

4.1 Applications


4.1.1 Base data

Base data determines the fundamental information and settings of an application. They can be configured through a mask in the application settings in the backend as follows:

- **Title:** The application's name.
- **URL Title:** How the application is shown in the URL field. No special characters allowed.
- **Thumbnail:** Uploadable image file from the file directory that will be displayed in the application overview. Click on **Select File**.
- **Map engine:** Choose an engine, OpenLayers 2 (deprecated) and OpenLayers 6 are available.
- **Description:** Text input field that can store the application's description, if provided.
- **Persistent map state:** Stores the state of specific map parameters and settings across sessions. For more information, please see [Share Elements](#).

Mapbender Demo

[Base data](#)
[Layouts](#)
[Layersets](#)
[CSS](#)
[Security](#)

 View

Title *

URL Title *

Thumbnail

Demo.png

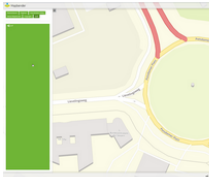

Map engine *

OpenLayers 6

Description

Desktop Demo

☐ Persistent map state

You can find more details on application creation under [Quickstart](#).

4.1.2 Layouts

In the Layouts section of an application you get an overview of the regions (layout sections) of the application. In the regions, the elements of the application are listed. An overview of all elements is available under [Elements](#).

Note: Different templates can have different regions: In the demo applications you can see two different region layouts. Not every element can be used in every region. Mapbender cares about that.

Layout of the Fullscreen template:


- Top toolbar (region for Buttons, Links, HTML,...)
- Sidepane (region for Layertree, Legend, Search, Print, HTML,...)
- Map area (region for Map, Scalebar,...)
- Footer (region for Copyright, Activity Indicator, Scale select,...)

Layout of the Mobile template:

- Footer (region for Copyright, Activity Indicator, Scale select,...)


- Map area (region for Map, Scalebar,...)
- MobilePane (region for dialogs like Layertree, Legend, BaseSourceSwitcher, FeatureInfo,...)



The  button located at the top right of each region allows adding elements. After pressing the button, a dialog will open, which allows for the selection of an element and its subsequent configuration.

All elements in an application can be moved between regions using drag and drop.



The regions themselves can be configured using the  button at right. There is no configuration option for region Map area and MobilePane.

The regions Top toolbar and Footer provide the following configuration options:

Configuration of the Top toolbar and the Footer

The Top toolbar and the Footer provide the following configuration options:

- **Screen type** (Any, Mobile, Desktop. Default: Any): The region will not be displayed when other screen types are used. *Any* will always show the region.
- **Alignment** (Left, Right, Center. Default: Right): Sets the alignment for the buttons, texts, links.
- **Generate menu for buttons**: Creates a dropdown menu for the elements.
- **Menu label input field**: Labeling for the dropdown menu.

Tip: Note: Using the dropdown menu is especially handy on mobile devices. There is a code snippet in [CSS editor](#) that adds a scroll bar to the menu and helps to increase the user experience.

Configuration of the Sidepane

The Sidepane provide the following configuration options through their  button:

Sidepane

Title	Type
Layer tree	Layer tree
Sketches	Sketches
Coordinates utility	Coordinates utility
About Mapbender	HTML



Type	Accordion	▼
Screen type	Any	▼
Width	350px	
Position	Left	▼
<input type="checkbox"/> Initially closed		

- **Type** (Accordion, Buttons, Unstyled. Default: Accordion): See explanation below.
- **Screen type** (Any, Mobile, Desktop. Default Any): The region will not be displayed when other screen types are used. *Any* - will always show the region.
- **Width** (in px) (in px. Default: 350px): Width of the Sidepane in Pixels.
- **Position** (Left, Right. Default: Left): Defines the placement of the sidepane.

- **Initially closed** (Default: disabled): Defines whether the Sidepane should be closed or open on start of the application.

The option **Type** adjusts the inserted elements:

- **Accordion** shows elements via tabs.
- **Buttons** shows elements via buttons.
- **Unstyled** does not contain any styling options at all and displays the elements in the configured backend order.

Button area (Elements)

Every element offers a set of buttons for configuration. The button area helps configuring elements in their specific regions. The following button functions are available:



- **Toggle show/hide element:** A shown element is visible in the application. A hidden one is not visible in the application itself, but can still be adjusted in the backend.
- **Show on mobile screens:** Displays an element only on mobile-sized screens.
- **Show on Desktop screens:** Displays an element only on Desktop-sized screens.
- **Edit:** Adjusts an element.
- **ACL element:** Sets specific visibility permissions for an element.
- **Delete:** Removes an element from both front- and backend.

Edit



The button opens a configuration mask for a specific element. The [Elements](#) overview page can help you to look up a setting for a specific element.

Acl element



The button opens a **Secure element** window that allows configuration of the [Access Control Lists View](#) right for users/groups.

Without configuration, an element has no access restriction and is available to the users/groups that can access the application.

When via Acl element View access rights are set, only the defined users/groups get access to the element.



Add users to restrict access to them with the button. A set checkmark next to the user account provides the necessary rights.

Secure element

+

Filtern	<input type="text"/>	view
root		view
tester		view

Speichern
Abbrechen

After setting specific access rights, the security key turns red. If you hover over the key with the cursor, you will see the names of the users who have rights to the element.

Toolbar					
Title	Type				
Scale display	Scale display				
Kartenkoordinaten	Coordinates display				
FeatureInfo	Button				
Print client	Button				
Image export	Button				
Legend	Button				

Acl element
 User: John Doe
 User: root

You can find more security details under [Security Concepts](#).

Delete

Removes an element from both front- and backend.






4.1.3 Layerset

Layersets are logical containers that can contain one or more layerset-instances (WMS services). In the demo applications two layersets are defined: Layerset “main” for the main map and layerset “overview” for the overview map. The names of the layersets can be freely chosen. Moreover, more than one layerset can be chosen in the map element. Layertree shows the layerset name when thematic layers is activated.




Layerset Overview Page

This page shows all layersets of an application. Every layerset is displayed as a container that holds services. On the page, there are buttons that help to manage the layersets:

Layerset actions

-  : Add layerset (large button)
-  : Add instance (small button)
-  : Edit layerset
-  : Set layerset as active on application startup
-  : Delete layerset

Instance actions

-  : Toggle show/hide instance
-  : Edit instance
-  : Delete instance




Mapbender Anwendungen Datenquellen Sicherheit Angemeldet als: root

Demo







Basisdaten Layouts Layerset CSS Sicherheit

Anzeigen




Filtern




+   

main

Id	Titel	Datenquelle	Typ
410	Mapbender Users	#41 Mapbender Users	↓ WMS   
411	OSM Demo	#51 OSM Demo	↓ WMS   

overview

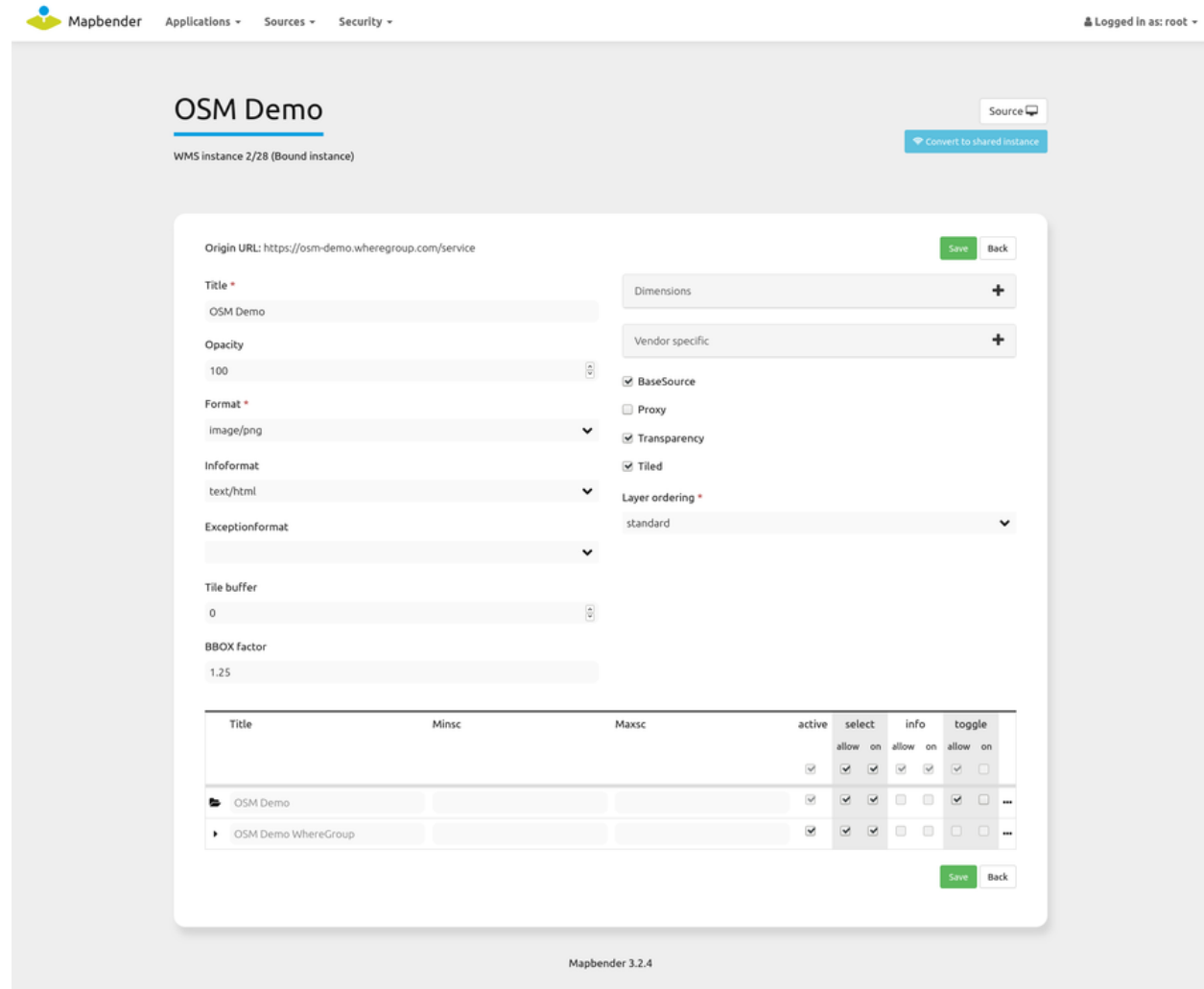
Id	Titel	Datenquelle	Typ
412	OSM Demo	#61 OSM Demo	↓ WMS   

+   

Speichern Abbrechen

Layerset-instances

Layerset-instances contain options to configure certain WMS parameters: image-format, info-format, exception-format, scales for the different layers etc. Layerset-instances can be configured by clicking on the gear symbol.



As soon a WMS service is integrated in a layerset, it is linked as a layerset-instance within the application. The screenshot above shows the *bound instance* 2/28 based on a WMS service. The associated data source is number 2, the layer's instance itself has the number 28. It is based on the WhereGroup OSM service and can be configured for the application in this dialog.

The properties of the WMS-Capabilities:

- **Title:** Title of the instance that is shown in the layerset list and their instances.
- **Format:** The image format which is used to get the map images for the application via the GetMap request. For raster data and aerial imagery the JPG format is recommended, in case of street maps the PNG format should be preferred. If you are in doubt use PNG.
- **Infoformat:** The format which is used for the GetFeatureInfo requests to the WMS. If you are in doubt use text/html or an analog HTML format that can be used in the dialog of the *FeatureInfo* element. Another possibility is text/plain.

- **Exceptionformat:** The format for error-messages that are returned by the WMS service.

The properties for the application

- **Opacity:** Choose the Opacity in percent. This value can be changed by the user in the [Layertree](#), if it's made available in the corresponding menu.
- **Tile buffer:** This parameter applies to services that are tiled and specifies whether to retrieve more surrounding tiles. With that they are already downloaded and visible during a pan movement. The higher the value, the more surrounding tiles are retrieved. Default: 0.
- **BBOX-Factor:** This parameter applies to services that are not tiled. You can specify how big the returned image should be. A value greater than 1 will request a larger map image. Default: 1.25 and you are free to set it to 1.
- **Vendor Specific Parameters:** See below.
- **Visible:** The service can be set visible with this option.
- **BaseSource:** The service should be treated as a BaseSource. This affects the [BaseSourceSwitcher](#), which should only display BaseSources, and the [Layertree](#), where these BaseSources can be hidden. See also the [hints](#) below.
- **Proxy:** If activated, the service will be requested via a proxy in Mapbender. See the [hints](#) below.
- **Transparency:** If this switch is enabled (that is the default), the service is requested with a transparent background. So in the WMS GetMap request with the parameter TRANSPARENT=TRUE.
- **Tiled:** The service is requested in tiles. The standard is not tiled. See the following [hints](#).

Layer-Order:

There are two ways to pass a layer order to the layer tree:

- **Standard**
- **QGIS Style**

This distinction has its origins in the way WMS services and capabilities documents are built.

The [OGC e-learning module for Web Map Services](#) describes the structure of an example WMS.

This WMS displays some point, line and polygon objects and raster images. The Capabilities document lists images (e.g. DTM) more at the top of the document than points (e.g. osm_points as the penultimate layer). Mapbender takes this approach both in the DataSources, where the service is loaded and registered in Mapbender, as well as in the Layerset instances.

In the layer tree, however, this order revolves. That is because, from our experience, users are accustomed to find layers that are higher up in the layer-tree also displayed as top layers. This configuration is also commonly found in a desktop GIS. Furthermore, points are displayed in the layer-tree above raster data.

The QGIS Server behaves differently here. A WMS can be easily deployed to a QGIS server using the QGIS project file. Order of the layers in the WMS Capabilities is then the same as the order used in your own QGIS project. So in general: points above, further below the lines, then finally the polygon data or raster images. QGIS and QGIS servers are not the only programs that can do this. But they are the best. However, this information is irrelevant for the GetMap call, because the order of the requests and how they are delivered to the server and back is solved in the WMS specification.

The following table summarizes the behavior again:

	layer order standard	layer order QGIS style
WMS Capabilities (from top to bottom)	polygon, line, point	point, line, polygon
Layerset-Instance (from top to bottom)	polygon, line, point	point, line, polygon
Layertree (from top to bottom)	point, line, polygon	point, line, polygon

This allows Mapbender to respond in the different ways that a WMS Capabilities document can be built up by simply adjusting the order in the layer tree.

Shared and bound instances

By default, all layerset-instances are created as bound instances. Bound instances must be configured individually for each application. Shared instances allow the use of a configured layerset-instance across multiple applications. Therefore, if multiple applications share the same configuration of a layerset-instance, the use of a shared instance saves the time of manually configuring the layerset-instance for each application. Changes made to a shared instance take effect in all applications using said instance.

A bound instance can be transformed into a shared instance by clicking on the blue button labeled **Convert to shared instance**. To transform a shared instance back into a bound instance the same button, now labeled **Convert to private instance**, can be used.

The screenshot shows the configuration page for 'OSM Demo', identified as 'WMS instance 2/27 (Bound instance)'. In the top right corner, a blue button labeled 'Convert to shared instance' with a Wi-Fi icon is highlighted with a red box. The configuration form includes fields for 'Origin URL' (https://osm-demo.wherogroup.com/service), 'Title' (OSM Demo), 'Opacity' (100), and 'Format'. There are also sections for 'Dimensions', 'Vendor specific', and checkboxes for 'BaseSource' (checked) and 'Proxy'.

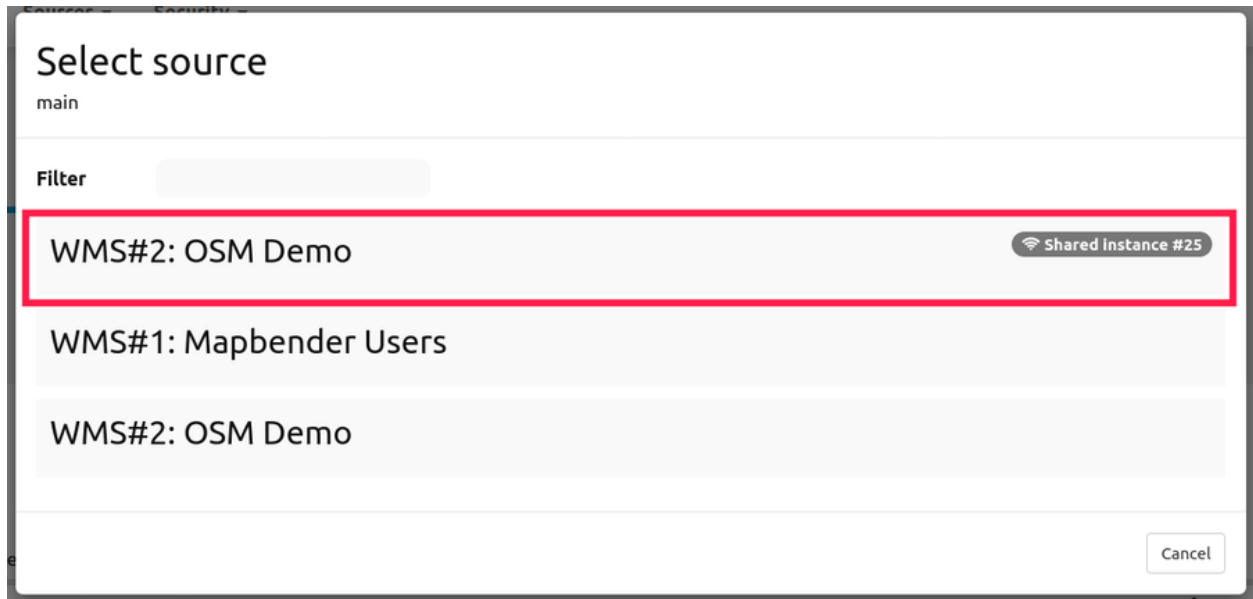
Shared instances are marked with a Wi-Fi symbol in the “type” column of the layerset overview.

The screenshot shows the configuration page for 'OSM Demo', identified as 'WMS instance 2/27 (Shared instance)'. In the top right corner, a blue button labeled 'Convert to bound instance' with an anchor icon is highlighted with a red box. The configuration form is identical to the one above, with the same fields and checkboxes.

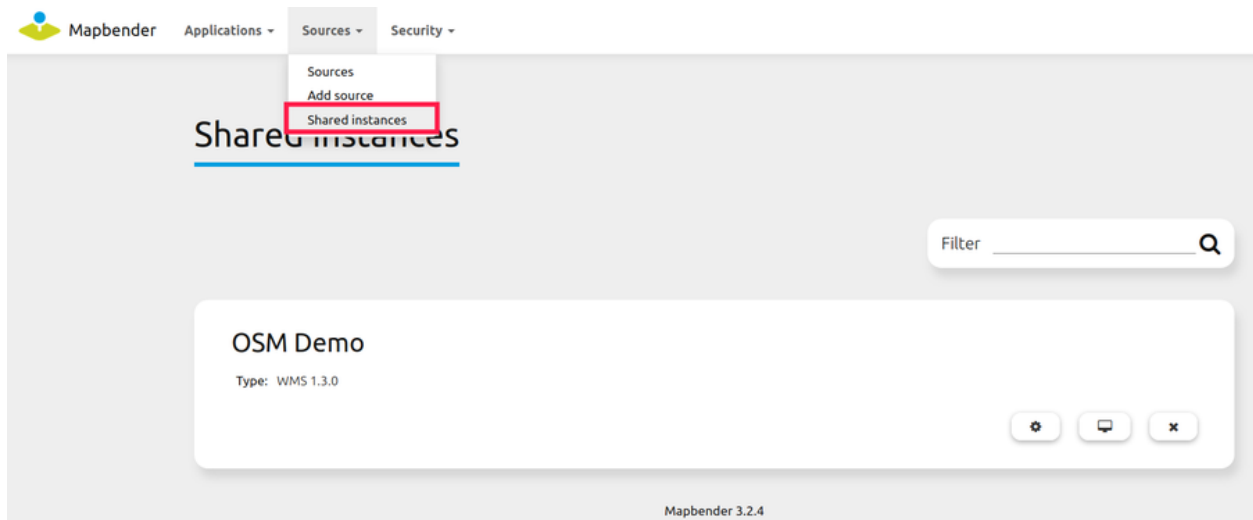
Bound instances are marked with an anchor symbol in the “type” column of the layerset overview.

Id	Title	Type	+	⚙	×
1/22	Mapbender Users (Mapbender Users)	⚓ WMS	⚙	×	
2/25	OSM Demo (OSM Demo)	📶 WMS	⚙	×	

Shared instances will appear as separate data sources. To incorporate a shared instance into an application simply add a data source to the application and select the shared instance from the list.



All shared instances can be viewed under **Sources** -> **Shared instances**.



Layer Configuration

The layer table offers several checkboxes and two scale visibility fields that adjust the functionality of your Service Instance. Please note that the checkbox on top changes all the layer instances for the respective configuration at once.

Title	Minsc	Maxsc	active	select		info		toggle		
				allow	on	allow	on	allow	on	
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Mapbender Users			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	...
User			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	...
Name			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	...

- **Title:** Layer title from Service information shown in Mapbender, adjustable.
- **Active (on/off):** Enables/disables a layer for the individual application layer. If not set, all other checkboxes of the same layer will be ignored.
- **Minscale / maxscale:** the scale range in which the layer should be displayed, 0 or no entry = no scale limitation.
- **Select allow:** Layer is active when the application starts.
- **Select on:** Selectable in geodata explorer.
- **Info allow:** layer info is active when the application starts.
- **Info on:** Layer provides feature info requests, info default activates the feature info functionality.
- **Toggle allow:** Folder can be opened via Layertree.
- **Toggle on:** Opens the folder on start of the application.
- **Reorder:** Allows to reorder the layers with drag & drop while using the application.
- ... -> Opens a dialog with more information:
- **Layer's name:** Layer name of the service information (for getMap-Requests, not adjustable).
- **Style:** If a WMS provides more than one style, you can choose a different style than the default style.

Notes on the effects of each configuration

Basesources

There are many ways to fill the Layertree and work with basic services:

- e.g. by hiding them in the layer tree and using the [BaseSourceSwitcher](#).
- or by work with the possibilities of the [thematic layer tree](#). This option will show the layerset name with a folder and checkbox in the layertree. You can define whether the layerset's folder should be shown and whether it should be opened on start.

Which option you choose depends entirely on your preferences.

Proxy

The use of a proxy makes sense if you want to avoid that the web browser accesses the service as a client directly, which is the default for OpenLayers-based applications. If this switch is activated, Mapbender accesses the services from its own URL, processes the images and displays them on the map. With that it is easy to provide a network-protected service secured by firewalls that can only be accessed by the web server on which Mapbender is running.

Tiling, map-size and performance

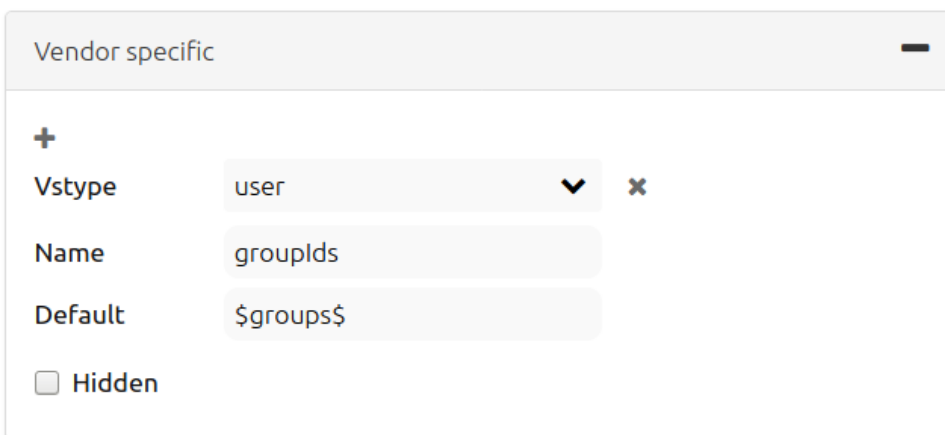
The “Tiled” parameter is used to request the map image in individual tiles rather than as a whole image. This should be turned on in general, if you use [Mapproxy](#) to provide a tiled service. But it also makes sense for normal, un-tiled services, since the perceived waiting time for the user gets lower: The map image appears, although not all tiles have been retrieved yet.

Keep in mind that the number of requests to a WMS increases rapidly: Depending on the screen resolution and the set tile size in the [Map](#), many requests are sent to the server. Although the returned images are not very large (usually you set tile sizes of 256x256 or 512x512 pixels), they are large in numbers. This is also valid in regard to the **tile buffer**. So it is a trade-off and a case-by-case distinction how to address the service. The performance can also be increased by setting the scales of a layer in the layerset-instance.

There are also some WMS services that only support a maximum image size that cannot be used with the high resolutions request that Mapbender can call. The Fullscreen template can be sized to the maximum screen width and the requested map image is then approximately the width and height of the visible browser window.

Vendor Specific Parameter

In a service instance, vendor specific parameters are appended to the WMS request. In Mapbender, the vendor specific parameters can be used to append user and group information of the logged-in user to the WMS request. Fixed values can also be transmitted. The following example shows the definition of a parameter “group”, which passes on the group of the user currently logged into Mapbender.



Vendor specific		
+		
Vstype	user	✕
Name	groupIds	
Default	\$groups\$	
<input type="checkbox"/>	Hidden	

- Name: Parameter name in WMS request.
- Default: Default value.
- Vstype: Mapbender specific variables: group, user (users), simple.

- Hidden: If this value is set, the requests are sent on the server so that the parameters are not directly visible.

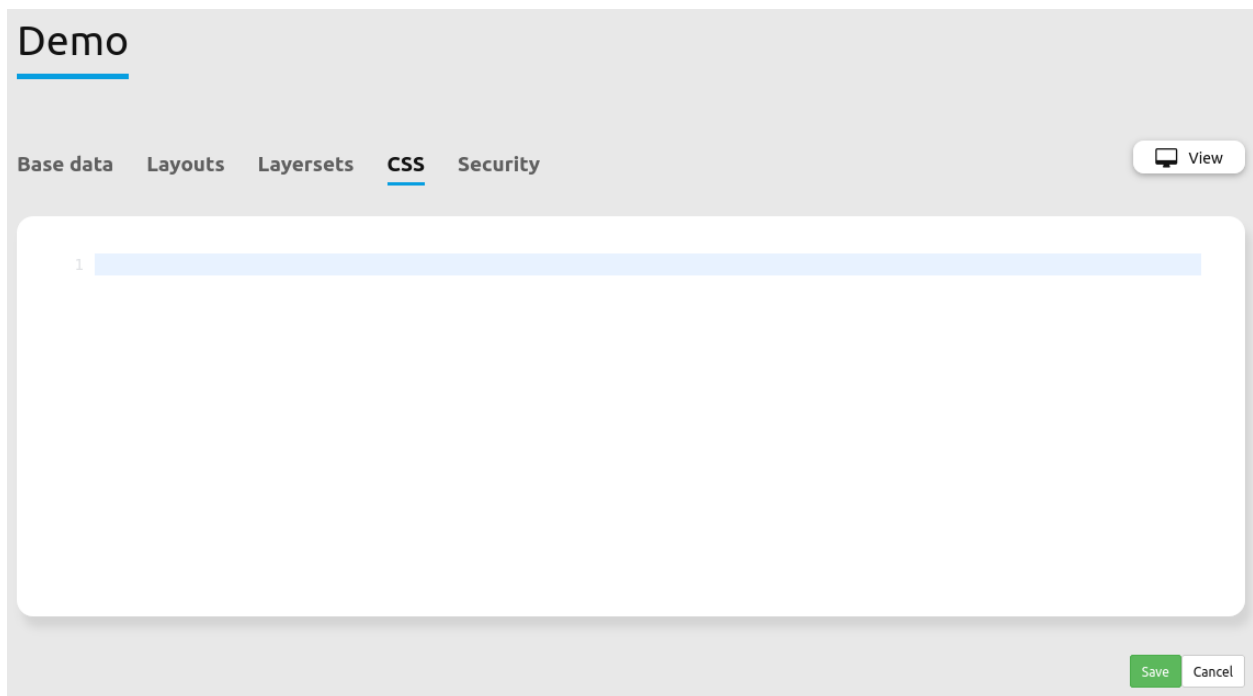
The item is useful for passing the service on only to specific users and groups. This happens e.g. for users via the `id` and for groups via the parameter `$groups$`.

Further information

- You can find information about using layersets in the [Quickstart](#).
- Further tips on manipulating WMS can be found in the [FAQs](#).
- The relevance of layersets for the display in the layertree is described in the Thematic Layers section of the [layertree documentation](#).
- Disabling one or more layersets for an application is possible in its [Map element](#) settings.
- Likewise, individual thematic layersets can be (de)activated via checkbox in the layerset configuration so that they are (not) visible in the map on application startup.

4.1.4 CSS editor

Mapbender offers an CSS editor for every application which easily allows you to change its style (colors, sizes, icons, ...). It is possible to define additional CSS classes that will overwrite the default style. You can use scss in the editor. And you can refer to an scss-file in your bundle.



Tip: A browser inspect tool can help you to find out the CSS classes you want to change.

Example

Add scrollbar to the top toolbar

When you use the menu option in the top toolbar you can define at what size of the screen the height should be reduced and a scrollbar should appear. This enhances the usability of the application on mobile devices.

```
// Scrollbar in Toolbox
.dropdown-menu {
  overflow-y: auto;
  max-height: calc(100vh - 100px);
}
```

The function itself can be found in the Layouts tab of the backend: Click the gear button and activate the checkbox `Generate menu for buttons`.

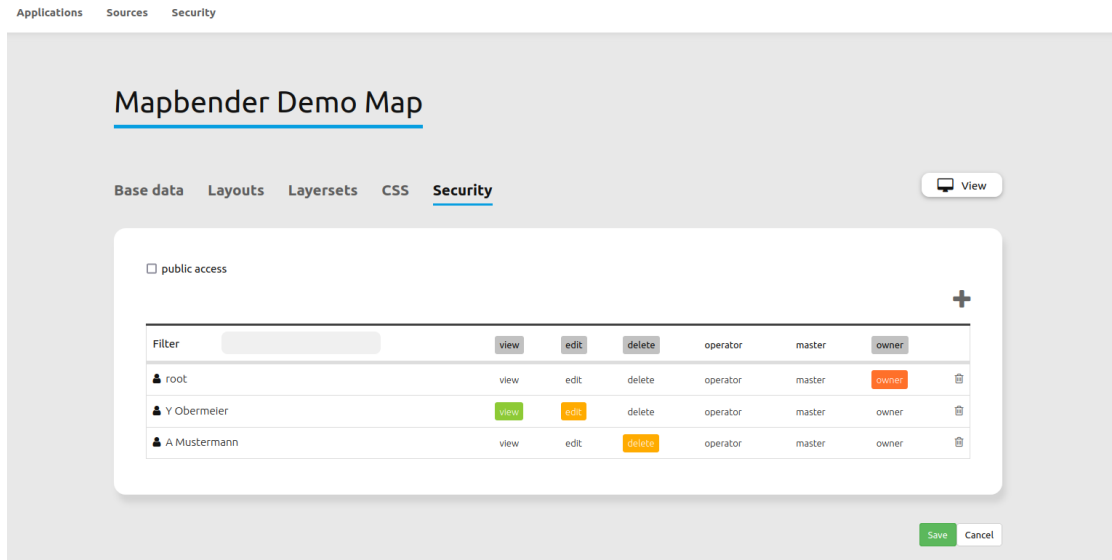
4.1.5 Security (Application tab)

Each application has its own security tab. Here, you can set individual permissions for a single application or its elements.

Assign an Application to a User/Group

Per default, all applications are available to all users/groups that have specific permissions to the backend. Nevertheless, it is possible to hide applications from individual users/groups:

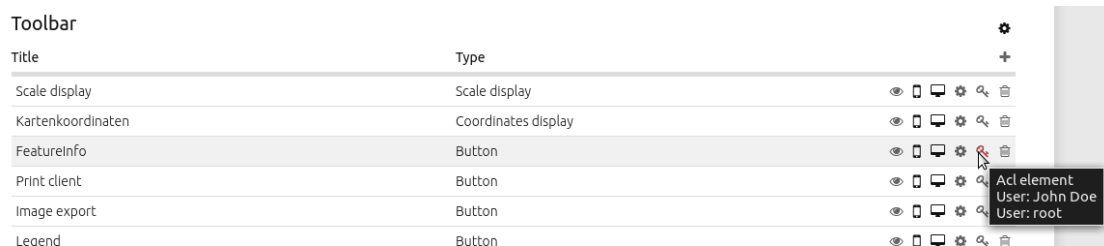
1. Edit your application by clicking **Application**.
2. Choose **Security**.
3. Publish your application for everyone by clicking **Security** ☐ **public access** or in the application overview by clicking the **Publish** button.
4. Alternatively and for an individual configuration, click the **Add users and groups** button and configure your selection. Then, set permissions like view, edit, delete, operator, master or owner via the rights table.
5. Logout from Mapbender by **Logout** and log in again with a configured account to test the configuration.
6. Another method would be to choose **Security** ☐ **Global Access Control Lists** ☐ **Applications** to quickly set permissions for several users/groups to all applications.



Assign single elements to a User/Group

Per default, all elements are available to all users/groups that have permission to an application. It is possible to hide single elements from individual users/groups:

1. Edit your application by clicking **Application**.
2. Choose **Layouts**.
3. Every element has a ACL element button (key). Choose the **ACL element** button from the element that should be only available for selected users/groups.
4. Now, add the users/groups via the **Add users and groups** button. Then, set permissions like view, edit, delete, operator, master or owner via the rights table.
5. Test your configuration. For example, open the application with a user account that has (no) rights to a previously configured element.




4.2 Sources

With Sources, you can register OGC WMS and OGC WMTS/TMS services in version 1.1.1 and 1.3.0 in Mapbender.

Further information about the registration process of services and their usage in applications is available in the Quickstart chapter [Load sources](#).

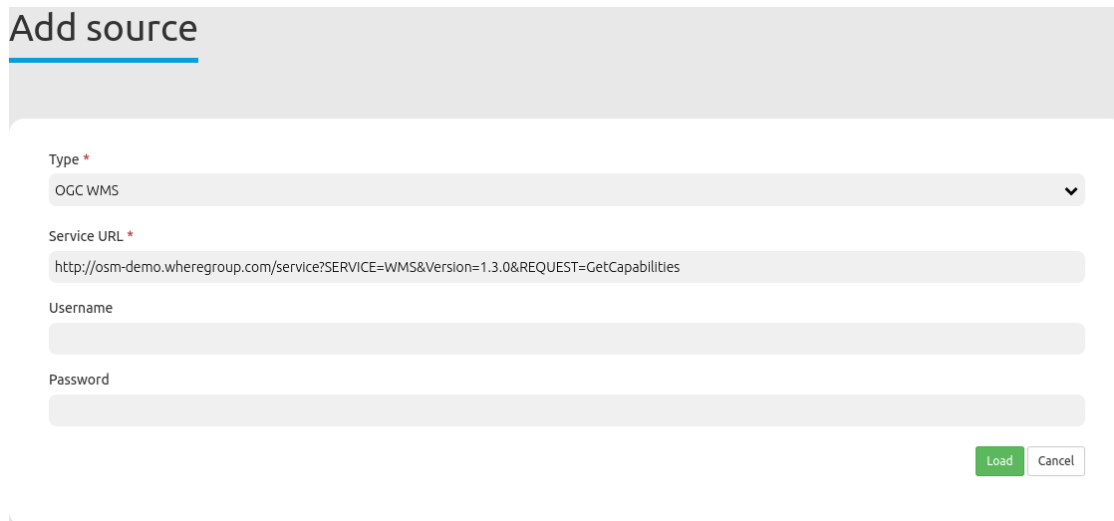
4.2.1 Load sources

Tip: It is important to check the service before it is registered in Mapbender. To do that, open the getCapabilities-Request (see example in Service URL) in your browser.

To register a service, click on  **Add source:**

- **Type:** Mandatory service type selection (OGC WMS or OGC WMTS / TMS).
- **Service URL:** URL to the Capabilities document of the service (e.g. [OGC WMS Version 1.3.0](#)).
- **Username / Password:** Input of the username and the password for secured services.

Click on Load to register the service to the repository.



After a successful registration, Mapbender will provide an overview of the service.

4.2.2 Sources Overview

The sources and shared instances sections list and provide additional information about the services registered:

- **Filter:** Search for services names, URLs, types and descriptions.
- **Show metadata:** Shows metadata about a specific service. Opens a new field that lists specific metadata, Mapbender applications that use the source, contact information, details (e.g. service version) and layers.

- **Update source:** Updates service information by reloading the getCapabilities document.
- **Delete source:** Removes the registered service from Mapbender.

The screenshot shows the 'Sources' tab in the Mapbender interface. At the top, there is a 'Sources' tab and a 'Shared instances' tab. Below the tabs, there is a '+ Add source' button and a 'Filter' search bar. Two service cards are listed:

- Mapbender User #25**
 - Type: WMS 1.3.0
 - URL: https://wms.wherogroup.com/cgi-bin/mapbender_user.xml?VERSION=1.3.0
 - Abstract: Mapbender User Reference
 - Actions: A menu button (top right) with icons for update, refresh, and delete.
- OpenStreetMap (OSM) Demo WhereGroup #19**
 - Type: WMS 1.3.0
 - URL: <https://osm-demo.wherogroup.com/service?VERSION=1.3.0>
 - Abstract: This is an OpenStreetMap Demo Service of WhereGroup
 - Actions: A menu button (top right) with icons for update, refresh, and delete.

4.2.3 Sources menu button

In the metadata dialog of a specific service, it is also possible to click on the menu button (top right) that allows:

- **Update source:** Updates service information by reloading the getCapabilities document.
- **Create shared instance:** Creates a new shared instance from the specific service. The instance is listed in the Shared instances tab.
- **Delete:** Removes the registered shared instance from Mapbender.

WMS OpenStreetMap (OSM) Demo WhereGroup

Metadata Applications Contact Details Layers

Title: OpenStreetMap (OSM) Demo WhereGroup
Version: 1.3.0
Origin URL: <https://osm-demo.wherogroup.com/service?VERSION=1.3.0>
Abstract: This is an OpenStreetMap Demo Service of WhereGroup
Online Resource: <https://osm-demo.wherogroup.com/service>
Exception Formats: XML, INIMAGE, BLANK
Fees: None
Access Constraints: This service is intended for private and evaluation use only. The geodata is licensed as Open Data Commons Open Database Lizenz (ODbL) (<http://opendatacommons.org/licenses/odbl/>)

Update source
Create shared instance
Delete

4.2.4 Updating sources

To update a source in the backend, you first need to navigate to the Sources backend list. On this page, look for the layer you wish to update via scrolling or use the search box. After you've found it, click on its



Refresh button. You can then update the WMS: If you wish, modify the URL or other settings, such as user name and/or password.

Hint: Of course, it is possible to update a source without changing any parameters. The Capabilities document is loaded again.

Moreover, there are two checkboxes handling layer updates:

Update source (OGC WMS)

Service URL *

<https://osm-demo.wherogroup.com/service?VERSION=1.3.0>

Username

Password

☒ Activate newly added layers

☒ Select newly added layers

Load Cancel

- **Activate newly added layers:** If active, the newly added layers will automatically set active in embedded applications. If the checkbox is not checked, new layers will not appear in the layertree.

- **Select newly added layers:** If active, the newly added layers will automatically be visible and set active in embedded applications. However, `Activate newly added layers` must also be set for this. If `Select newly added layers` is not set, the layer will appear in the layertree but will not be activated.

If you want to save the changes, click the `Load` button to refresh the WMS. This will re-read the `getCapabilities` document. The updated version will be displayed in the configuration settings, with changes applied in applications using the service.

4.3 Security (FOMUserBundle)

4.3.1 Security Concepts

Security as provided by the FOMUserBundle is anchored on these base concepts:

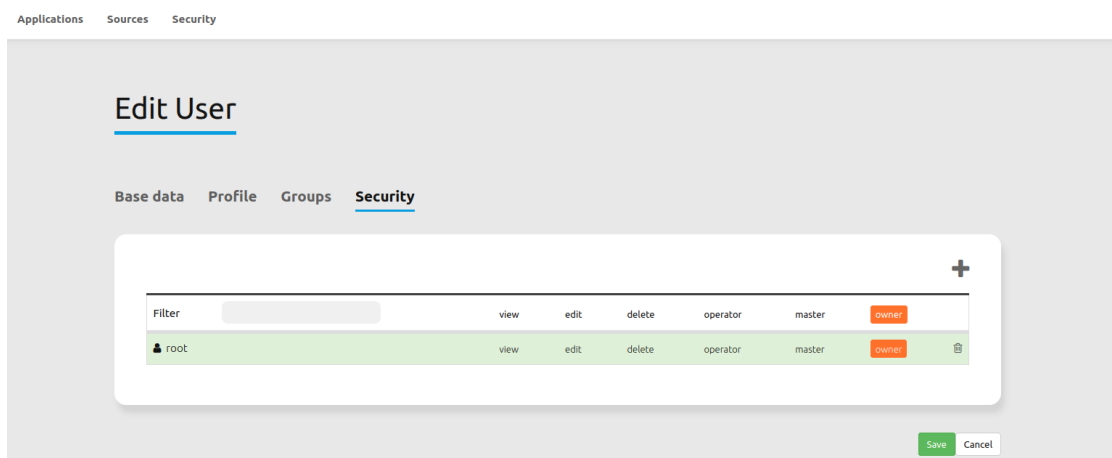
- *Users*
- *Roles and Groups*
- *Access Control Lists*

Rights management

Mapbender provides different rights. They refer to the *Access Control Lists*.

- view - Whether someone is allowed to view the object.
- edit - Whether someone is allowed to make changes to the object.
- delete - Whether someone is allowed to delete the object.
- operator - Whether someone is allowed to perform all of the above actions.
- master - Whether someone is allowed to perform all of the above actions and in addition is allowed to grant any of the above mentioned permissions to others.
- owner - Whether someone owns the object. An owner can perform any of the above actions and grant master and owner permissions.

Assign roles to a user by **Users** → **Edit your User** → **Security**.



Assign a user to another User/Group

1. Edit a user by clicking **Security** ▢ **Users**.
2. In the user administration, choose **Security**.
3. Give users/groups individual rights on the selected user: Add users/groups via the **Add users and groups** button. Thereafter, set permissions within the rights table.
4. You have now assigned a user/group controlling options over another user account. Test your configuration with the entitled user accounts.

4.3.2 Access Control Lists

The security for domain objects is implemented in Mapbender using Access Control Lists. ACLs provide flexible permissions for individual objects like applications, services or user- and group management.

Mapbender offers these rights to create individual permissions and secure the domain objects:

- View : View an existing object
- Create : Create a new object
- Edit : Edit an existing object
- Delete : Delete an existing object
- Operator : View, Edit, and Delete permission.
- Master : Operator permission, can manage all permissions up to operator level.
- Owner : Master permission, can grant master permission as well.

4.3.3 Users

User are implemented as FOM\UserBundle\Entity\User and stored in the database. The entity has only some basic information about the user itself, more complex user data will have to be implemented by user profiles (yet to be done).

The bundle provides all means to administrate users by admin as well as self-registration and password recovery.

The user with the id 1 (root) is special, as this user is created during installation and will always be given full access. If all is lost, you can use this user to manage everything. And in the event that the credentials for this user are also lost, a console command (fom:user:resetroot) is available for resetting.

Forgot Password

If a user has forgotten his/her password, he can use the “Forgot password?” link in the Login screen to request a new one. For that he types in his username or email address.



Forgot password

To reset your password please provide your username or your e-mail address.

User name or Email *

► Back to login

Require password

After that, the user should receive an e-mail with a link which leads to a page where a password reset is possible. The link is not valid anymore after this operation. The text of the mail can be customized in the `/FOM/UserBundle/Resources/translations/messages.en.xlf` file.

The functionality can be switched off in the `config.yml`.

```
fom_user:  
  reset_password: true # true/false
```

Registration

Users can self-register themselves in Mapbender. For this you have to adjust the setting `fom_user:selfregister` in the `config.yml` to true.

```
fom_user:  
  selfregister: false # true/false
```

The Login-dialog contains a “Register” link. This opens a page where the user can type in his/her name, password and e-mail adresss.



Registration

User name *

Email *

Choose password *

Confirm password *

► [Back to login](#)

Register

After that he gets a confirmation mail to complete the registration. Until that time he is only managed as an inactive user in Mapbender.

The text of the confirmation mail can be customized in the `/FOM/UserBundle/Resources/translations/messages.en.xlf` file.

Activation of users

Users can be set activated or deactivated by Administrators with the User-ACL-Right of at least *edit*. For this purpose, a checkbox exists in the Edit User dialog. A user with administration rights cannot activate or deactivate himself.

A user who is deactivated cannot login into Mapbender anymore until he gets activated again.

Users which have self-registered themselves but have not approved the activation mail can now be activated by an administrator.

Login Failures

Login failures are responded with the Message “Bad credentials”. For security reasons it is not shown if the error is based on a wrong username or a wrong password. Login failures will not lock the account indefinitely after four attempts. Rather the account will be locked for a given period of time.

The config.yml allows to adjust this behaviour:

```
fom_user:

  # Allow to create user log table on the fly if the table doesn't exists.
  # Default: true
  auto_create_log_table: true

  # Time between to check login tries
  login_check_log_time: "-5 minutes"

  # Login attempts before delay starts
  login_attempts_before_delay: 3
```

(continues on next page)

(continued from previous page)

```
# Login delay after all attempts are failed
login_delay_after_fail: 2 # Seconds
```

- **auto_create_log_table:** Backwards compatibility parameter (default: true).
- **login_check_log_time:** Cleaning of the login-failure table (default: -5 minutes)
- **login_attempts_before_delay:** Number of login failures before the login delay starts (default: 3)
- **login_delay_after_fail:** Number of seconds of the login-delay (default: 2).

4.3.4 Roles and Groups

Roles are used for global permission checks when no domain object is involved and can be used as security identities of Access Control Entries in Access Control Lists of domain objects.

In Mapbender, there are currently two roles for rights allocation in the backend available:

- **All authenticated users:** Gives global set permissions for one or more domain object to all users that are logged in while working with Mapbender.
- **Anonymous users:** Sets global permissions for users which are browsing Mapbender without an individual account.

Groups are individually created database entities which can be assigned to one or more users. Therefore their primary use is to collect rights which are assigned to every user of that group.

CUSTOMIZATION / TIPS AND TRICKS

5.1 How to create your own Template?

Mapbender comes with application templates out of the box, you can find them in the Mapbender Core-Bundle `/application/mapbender/src/Mapbender/CoreBundle/Template`. But usually you want to use your own templates with your own corporate design.

To prevent overwriting your custom templates after an Mapbender upgrade you should create an extra bundle to safely store your custom files.

You can also change the style of your application with the built-in *CSS-Editor*.

5.1.1 How to create your own template?

Steps for including your templates:

- Create your own bundle
- Create a new namespace
- Create a template PHP-file to register your template
- Create your own Twig-file
- Create your own CSS-file(s)
- Register your bundle in `app/AppKernel.php`
- Use your template

To help you we prepared a Workshop/DemoBundle, which can be used not only for application templates, but also for customizing the administration interface. For the following steps, you can download the files with the following links:

- <https://github.com/mapbender/mapbender-workshop/tree/master>

5.1.2 Create your own bundle

User bundles are stored in the src-directory /application/src.

This is how the structure can look like:

```
src/Workshop/DemoBundle/  
  
    WorkshopDemoBundle.php  
  
    /Resources  
        /public  
            demo_fullscreen.css  
  
        /image  
            workshop.ico  
            workshop_logo.png  
            print.png  
  
        /views  
            /Template  
                fullscreen_demo.html.twig  
  
    /Template  
        DemoFullscreen.php
```

The following files have to be altered for design changes:

- twig-file: to change the structure (e.g. - delete a component like the sidebar)
- css-file: to change colors, icons, fonts

5.1.3 Create a new namespace

The file WorkshopDemoBundle.php creates the namespace for the bundle and refers to the template and to your css-files.

```
<?php  
  
namespace Workshop\DemoBundle;  
  
use Mapbender\CoreBundle\Component\MapbenderBundle;  
  
class WorkshopDemoBundle extends MapbenderBundle  
{  
    public function getElements()  
    {  
        return array(  
            // 'Workshop\DemoBundle\Element\MapKlick',  
        );  
    }  
    public function getTemplates()  
    {  
        return array('Workshop\DemoBundle\Template\DemoFullscreen');  
    }  
}
```

5.1.4 Create your own template file

In our example the template file is called FullscreenDemo.php. You find it at `src/Workshop/DemoBundle/Template/FullscreenDemo.php`.

In the template file you define the name of your template, the regions that you want to provide and refer to a twig file.

```
<?php

namespace Workshop\DemoBundle\Template;

use Mapbender\CoreBundle\Template\Fullscreen;

class DemoFullscreen extends Fullscreen
{
    protected static $title           = "Fullscreen Template Workshop";
    protected static $regions         = array('toolbar', 'sidepane', 'content', 'footer');
    protected static $regionsProperties = array(
        'sidepane' => array(
            'tabs' => array(
                'name' => 'tabs',
                'label' => 'mb.manager.template.region.tabs.label',
            ),
            'accordion' => array(
                'name' => 'accordion',
                'label' => 'mb.manager.template.region.accordion.label'
            )
        );
    protected static $css             = array(
        '@MapbenderCoreBundle/Resources/public/sass/template/fullscreen.scss',
        '@WorkshopDemoBundle/Resources/public/demo_fullscreen.scss',
    );
    protected static $js              = array(
        '@FOMCoreBundle/Resources/public/js/frontend/sidepane.js',
        '@FOMCoreBundle/Resources/public/js/frontend/tabcontainer.js',
        '@MapbenderCoreBundle/Resources/public/mapbender.container.info.js',
        '/components/jquerydialogextendjs/jquerydialogextendjs-built.js',
        '/components/vis-ui.js/vis-ui.js-built.js'
    );
    public $twigTemplate = 'WorkshopDemoBundle:Template:demo_fullscreen.html.twig';
}
```

5.1.5 Create your own twig-file

You find the twig-files at the following path:

- `application/mapbender/src/Mapbender/CoreBundle/Resources/views/Template`

The easiest way to create your own twig file is to copy an existing twig, save it under a new name and change the content like colors.

Use the existing template from `mapbender/src/Mapbender/CoreBundle/Resources/views/Template/fullscreen.html.twig` and copy it to `fullscreen_demo.html.twig`.

5.1.6 Create your own css-file

Create an empty css-file and fill it with content. You only have to define the parts that have to look different from the default style of the element.

Firebug can help you to find out the styles you want to change.

Your file could be named like this: `src/Workshop/DemoBundle/Resources/public/demo_fullscreen.css` and have the following definition:

```
.toolBar {
    background-color: rgba(0, 29, 122, 0.8) !important;
}

.toolPane {
    background-color: rgba(0, 29, 122, 0.8) !important;
}

.sidePane {
    overflow: visible;
    background-image: url("");
    background-color: #eff7e9;
}

.sidePane.opened {
    width: 350px;
}

.logoContainer {
    background-color: white !important;
    background-image: url("") !important;
    -webkit-box-shadow: 0px 0px 3px #0028AD !important;
    -moz-box-shadow: 0px 0px 3px #0028AD !important;
    box-shadow: 0px 0px 3px #0028AD !important;
}

.sidePaneTabItem {
    background-color: #0028AD;
}

.layer-opacity-handle {
    background-color: #0028AD;
}

.mb-element-overview .toggleOverview {
    background-color: #0028AD;
}

.button, .tabContainerAlt .tab {
    background-color: #0028AD;
}

.iconPrint:before {
    /*content: "\f02f"; */
    content: url("image/print.png");
}

.popup {
```

(continues on next page)

(continued from previous page)

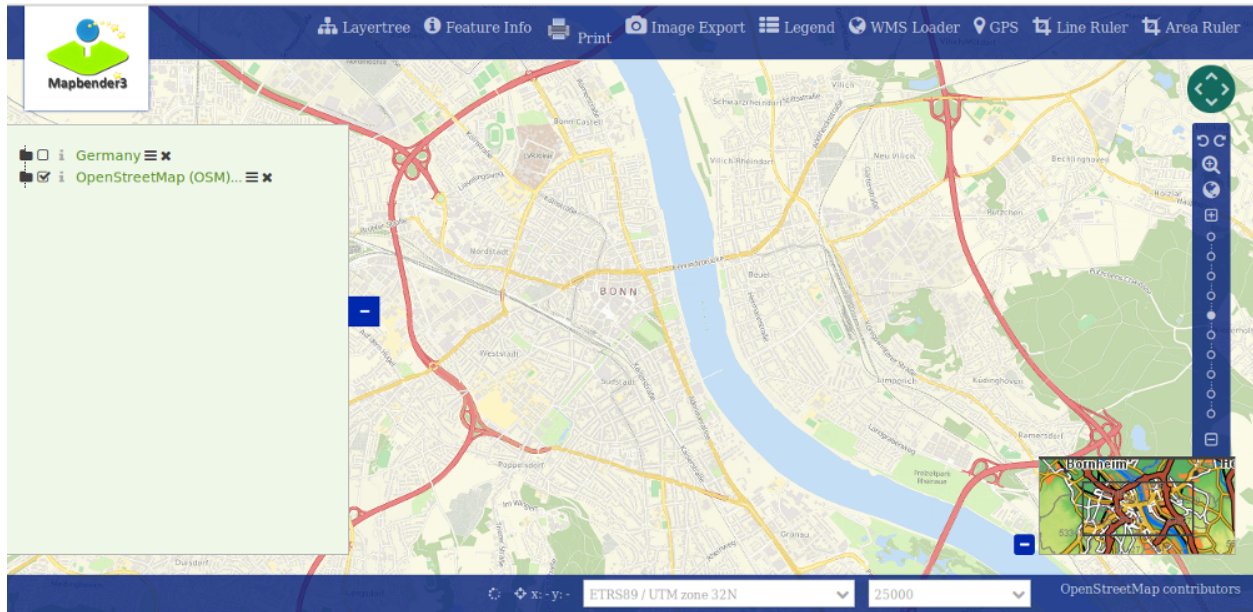
```

background-color: #eff7e9;
background-image: url("");
}

.pan{
background-color: rgba(0, 93, 83, 0.9);
}

```

The result of these few lines of css will look like this:



When you open your new application a css-file will be created at:

- web/assets/WorkshopDemoBundle__demo_fullscreen__css.css

If you do further edits at your css file you may have to delete the generated css file in the assets directory to see the changes. You should also clear the browser cache.

```
sudo rm -f web/assets/WorkshopDemoBundle__demo_fullscreen__css.css
```

Style the administrative pages

Please change the following css-files for the backend pages:

- login.css : Change the design of the login page
- manager.css : Change the design of the administration pages (e.g. application overview)
- password.css : Change the design of the password pages (e.g. Reset Password - page)

You only have to define the parts that have to look different than the default page style.

Firebug can help you to find out the styles you want to change.

Referencing the CSS-files is possible with FOMManagerBundle and FOMUserBundle. They must be filed under app/Resources/. The already contained twig-files overwrite the default settings if configured correctly (Requirements from manager.html.twig file). Alternatively, it is possible to copy a twig-file and adjust it afterwards.

```
cp fom/src/FOM/ManagerBundle/Resources/views/manager.html.twig app/Resources/
↪ FOMManagerBundle/views/
```

5.1.7 Register your bundle in app/AppKernel.php

You have to register your template:

- mapbender/app/AppKernel.php

```
class AppKernel extends Kernel
{
    public function registerBundles()
    {
        $bundles = array(

            ....

            // Mapbender bundles
            new Mapbender\CoreBundle\MapbenderCoreBundle(),
            ...

            new Workshop\DemoBundle\WorkshopDemoBundle(),

        );
    }
}
```

Add write access to the web-directory for your webserver user.

```
chmod ug+w web
```

Update the web-directory. Each bundle has it's own assets - CSS files, JavaScript files, images and more - but these need to be copied into the public web folder:

```
app/console assets:install web
```

Alternatively, as a developer, you might want to use the symlink switch on that command to symlink instead of copy. This will make editing assets inside the bundle directories way easier.

```
app/console assets:install web --symlink --relative
```

Now your template should show up in the template list when you create a new application.

5.1.8 How to use a new template

There are different ways of how to use the new template:

Usage in YAML-applications

You can adjust the YAML-applications in app/config/applications and change the template parameter.

```
template:    Workshop\DemoBundle\Template\DemoFullscreen
```

Usage in new applications from the backend

If you create a new application in the administration interface of Mapbender, you can choose the new template.

Usage in an existing application

For existing applications you can change the parameter in the Mapbender database in the column template of the table mb_core_application.

For the *WorkshopDemoBundle* you change the entry from Mapbender\CoreBundle\Template\Fullscreen to Workshop\DemoBundle\WorkshopDemoBundle.

5.1.9 Usecases

How do I change the logo, the title and the language? This and more tips can be found here: [YAML Configuration \(Configuration and Application files\)](#).

How do I change the buttons?

Mapbender uses 'Font Awesome Icons' font icon collection:

```
@font-face {
  font-family: 'FontAwesome';
  src: url("../bundles/fomcore/images/icons/fontawesome-webfont.eot?v=3.0.1");
  src: url("../bundles/fomcore/images/icons/fontawesome-webfont.eot?#iefix&v=3.0.1") format(
↪ "embedded-opentype"), url("../bundles/fomcore/images/icons/fontawesome-webfont.woff?v=3.0.1")
↪ format("woff"), url("../bundles/fomcore/images/icons/fontawesome-webfont.ttf?v=3.0.1") format(
↪ "truetype");
  font-weight: normal;
  font-style: normal;
}
```

In your css-file you can refer to a font images like this:

```
.iconPrint:before {
  content: "\f02f";
}
```

If you want to use an image you could place the image in your bundle and refer to it like this

```
.iconPrint:before {
  content: url("image/print.png");
}
```

5.1.10 Try this out

- you can download the Workshop/DemoBundle at <https://github.com/mapbender/mapbender-workshop>
- change the color of your icons
- change the size of your icons
- change the color of the toolbar
- use an image instead of a font-icon for your button
- move the position of your overview to the left
- Have a look at the workshop files to see how it works

5.2 YAML Configuration (Configuration and Application files)

The following Configuration files are under application/app/config.

5.2.1 parameters.yml

The following fundamental Mapbender parameters are specified here.

Database

The files `parameters.yml` and `config.yml` are needed to configure databases in Mapbender. In `parameters.yml`, (multiple) variables for database connection(s) can be defined. These variables are being processed in `config.yml`. An alias is assigned to each database connection.

- **database_driver: Database driver. Possible values are:**

- `pdo_sqlite` - SQLite PDO driver
- `pdo_mysql` - MySQL PDO driver
- `pdo_pgsql` - PostgreSQL PDO driver
- `oci8` - Oracle OCI8 driver
- `pdo_oci` - Oracle PDO driver

Please note: Necessary PHP drivers need to be installed and activated.

Example: Database configuration in `parameters.yml`, when PostgreSQL is used:

```
database_driver: pdo_pgsql
database_host: localhost
database_port: 5432
database_name: mapbender
database_path: ~
database_user: postgres
database_password: secret
```

Use of several databases

Mapbender can handle several databases. This is recommended if you want to keep your data separated from Mapbender data. Or if you want to use code that doesn't belong to a Mapbender bundle.

You need a second database for *geo data search* (with SearchRouter) and data collection (Digitizer).

The default database connection (default_connection: default) is used by Mapbender.

If you want to use another database, you have to define a database connection with a different name.

```
parameters:
  # database connection "default"
  database_driver:  pdo_pgsql
  database_host:    localhost
  database_port:    5432
  database_name:    mapbender
  database_path:    ~
  database_user:    postgres
  database_password: postgres

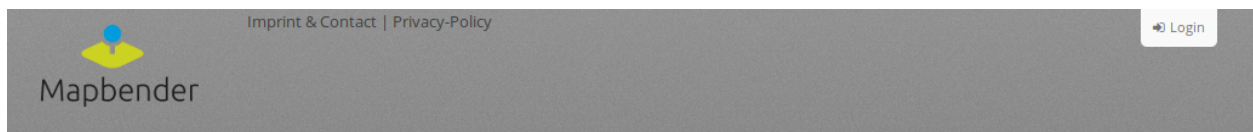
  # database connection "search_db"
  database2_driver:  pdo_pgsql
  database2_host:    localhost
  database2_port:    5432
  database2_name:    search_db
  database2_path:    ~
  database2_user:    postgres
  database2_password: postgres
```

Now, you can refer to the database **search_db** in the elements SearchRouter and Digitizer.

To learn more about this structure, visit the [Symfony documentation](#).

Mapbender uses Doctrine. Doctrine is a collection of PHP libraries ([Doctrine project](#)).

Disclaimer



A disclaimer can be added through the use of site links.

```
mapbender.sitelinks:
  - link: https://mapbender.org/en/legal-notice/           # Link URL
    text: Imprint & Contact                                # Link
  ↪ text
  - link: https://mapbender.org/en/privacy-policy/
    text: Privacy Policy
```

Site links will be separated by "|".

Language settings

Mapbender is automatically adjusted to your browser's language. Yet it is possible to set a language option in the configuration file **app/config/parameters.yml**. If a translation of your browser's set language is missing in Mapbender, it will then take a fallback language. We recommend en (English) or de (German) as fallback options.

Available language codes are:

- en for English (default)
- de for German
- es for Spanish
- fr for French
- it for Italian
- nl for Dutch
- pt for Portugese
- ru for Russian
- tr for Turkish
- uk for Ukrainian

Configuration example:

```
# locale en, de, es, fr, it, nl, pt, ru, tr, uk are available
fallback_locale: en
locale:          en
secret:          ThisTokenIsNotSoSecretChangeIt
```

More information in [Translation in Mapbender](#).

Logo

In parameters.yml, you can refer to your own logo and to an alternative image for the login page. This change has a global impact on the whole Mapbender installation.

```
branding.logo: ./bundles/mapbendercore/image/logo_mb.png
branding.login_backdrop: ./bundles/mapbendercore/image/body.png
```

The files must be accessible under application/web.

Mailer

Mailer information in parameters.yml (e.g. smtp or sendmail).

Configuration example:

```
mailer_transport: smtp
mailer_host:      localhost
mailer_user:      ~
mailer_password:  ~
```

The functions 'Self-Registration' and 'reset password' need a mailer.

More information in chapter [Users](#).

Project name

The name of the project (default: Mapbender) can be changed in `parameters.yml`. The change has a global impact on the whole Mapbender installation.

```
branding.project_name: Geoportal
```

Important note: In `parameters.yml` **tabulators may not be used for indentation** instead you need to use space.

Proxy settings

If you use a proxy, you need to change `parameters.yml`.

Hint: OWSPoxy3 is a transparent Buzz-based proxy that uses cURL for connection to web resources via/without a proxy server.

Configuration example:

```
# OWSPoxy Configuration
ows_proxy3_logging: false           # logging of requests, default is false, true logs in table
↪owsproxy_log
ows_proxy3_obfuscate_client_ip: true # obfuscates a client ip, default is true, true will hide the
↪last byte of the client's ip address
ows_proxy3_host: myproxy            # proxy definition for connection via a proxy server. Host
↪name of the proxy server
ows_proxy3_port: 8080               # proxy definition for connection via a proxy server. Port
↪name of the proxy server
ows_proxy3_connecttimeout: 60
ows_proxy3_timeout: 90
ows_proxy3_user: ~                  # user name for proxy server (set user for proxy server if
↪needed)
ows_proxy3_password: ~             # password for proxy server (set password for proxy server if
↪defined)
ows_proxy3_noproxy:                # list of hosts for connections without proxy server
- 192.168.1.123
```

SSL certificate

For productive environments, it is important to install a SSL certificate. After that, set the `parameters.cookie_secure` variable in your `parameters.yml` to true. This ensures that the Login cookie is only transmitted over secure connections.

5.2.2 config.yml

- **fom_user.selfregistration:** To enable or disable self-registration of users, change the fom_user.selfregistration parameter. You have to define self_registration_groups, so that self-registered users are added to these groups automatically, when they register. They will get the rights that are assigned to these groups.
- **fom_user.reset_password:** In the same way the possibility to reset passwords can be enabled or disabled.
- **framework.session.cookie_httponly:** For HTTP-only session cookies, make sure the framework.session.cookie_httponly parameter is set to true.

Database

Important: Every database defined in parameters.yml needs to have a placeholder in config.yml as well:

```
doctrine:                                     # Values, surrounded by %-marks, are variables
  dbal:
    default_connection: default               # Database connection, used as standard in
↳ Mapbender (``default_connection: default``).
    connections:
      default:
        driver:      "%database_driver%"      # More information below the code
        host:        "%database_host%"        # Database host on which the database runs.
↳ Either name of the host (e.g. localhost) or IP address (e.g. 127.0.0.1).
        port:        "%database_port%"        # Port, the database listens to (e.g. 5432 for
↳ PostgreSQL).
        dbname:      "%database_name%"        # Name of the database (e.g. mapbender). Create
↳ a database with the command ``doctrine:database:create`` bzw. ``doctrine:schema:create``.
        path:        "%database_path%"        # %database_path%, path to the file of the
↳ SQLite database. If you don't use a SQ-lite database, write (~) or ``null``.
        user:        "%database_user%"        # User name for database connection.
        password:    "%database_password%"    # Password.
        charset:     UTF8                    # Coding of the database.
        logging:     "%kernel.debug%"        # Option, SQLs won't be logged (standard:
↳ %kernel.debug%). More information: <http://www.loremipsum.at/blog/doctrine-2-sql-profiler-in-
↳ debugleiste>`.
        profiling:   "%kernel.debug%"        # Profiling SQL requests. This option can be
↳ turned of in production. (standard: %kernel.debug%)
```

Use of several databases

Example with two database connections in config.yml:

```
doctrine:
  dbal:
    default_connection: default
    connections:
      # database connection default
      default:
        driver:      "%database_driver%"
        host:        "%database_host%"
        port:        "%database_port%"
        dbname:      "%database_name%"
```

(continues on next page)

(continued from previous page)

```

    path:      "%database_path%"
    user:      "%database_user%"
    password:  "%database_password%"
    charset:   UTF8
    logging:   "%kernel.debug%"
    profiling: "%kernel.debug%"
# database connection search_db
search_db:
    driver:    "%database2_driver%"
    host:      "%database2_host%"
    port:      "%database2_port%"
    dbname:    "%database2_name%"
    path:      "%database2_path%"
    user:      "%database2_user%"
    password:  "%database2_password%"
    charset:   UTF8
    logging:   "%kernel.debug%"
    profiling: "%kernel.debug%"

```

More information under `parameters.yml`.

5.2.3 YAML Application files

YAML application files are stored under **app/config/applications**. “Mapbender mobile”, “Mapbender Demo Map” and “Mapbender Demo Map basic” are pre-implemented as example applications.

If you do not want the three example applications to be visible, you can change the variable ‘published’ to ‘false’.

```

parameters:
    applications:
        mapbender_mobile:
            [...]
            published: false

```

Now the applications will not be visible for users (except for root user).

New YAML applications can be placed in the folder and will be automatically recognized by Mapbender.

5.2.4 Mapbender Demo Map

This is the main Demo application. Should be used for a desktop based application.

Detailed descriptions of the elements at [Elements](#).

5.2.5 Mapbender Demo Map basic

Differences to the main Demo Map:

Toolbar Uses *Coordinates Utility* instead of *POI*.

Sidepane Has no elements pre-implemented.

Map area Uses *Coordinates Utility* instead of *Scale Display* and *POI*.

Detailed descriptions of the elements at *Elements*.

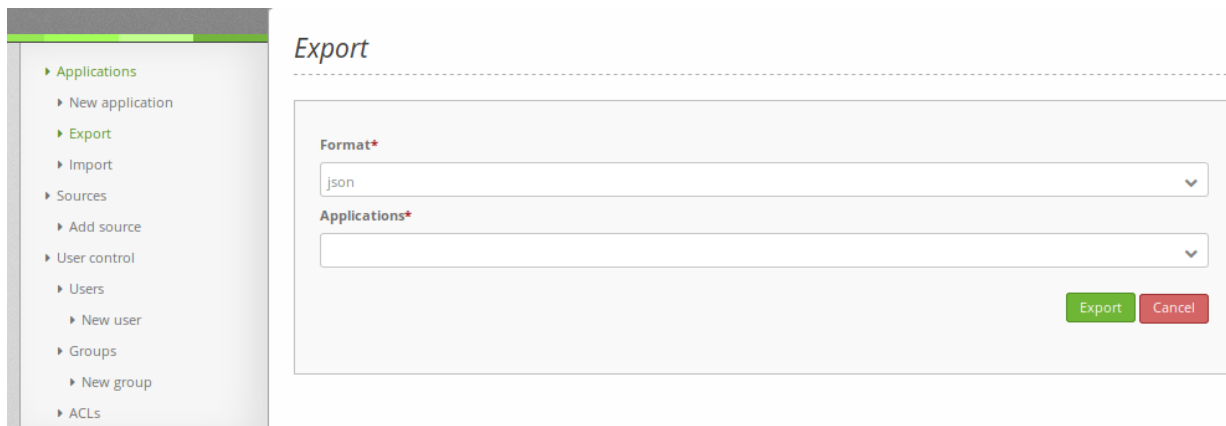
5.2.6 Mapbender mobile

For a mobile template on smartphones and tablets.

5.2.7 Export/import YAML application files over the user interface

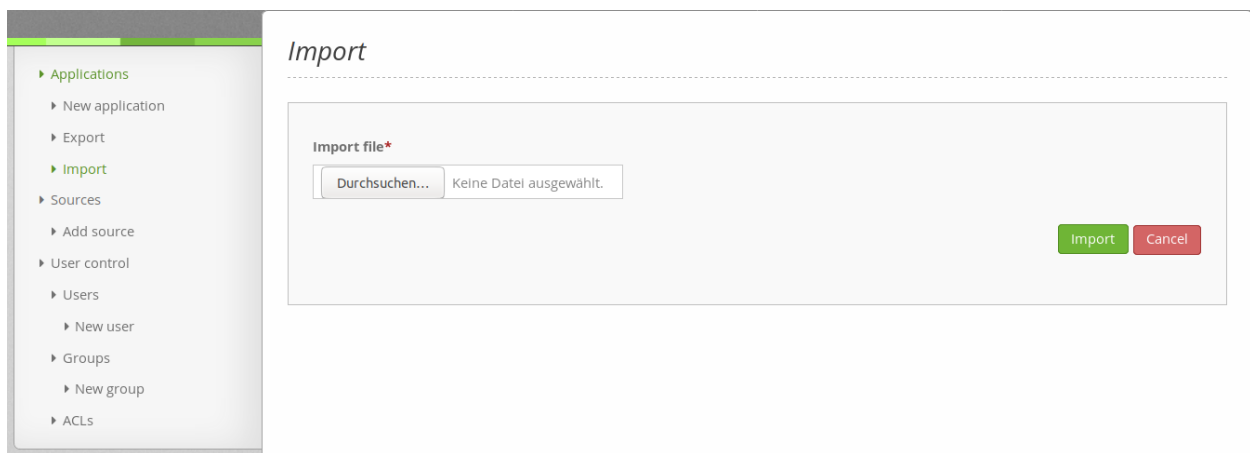
Export

You can export applications as JSON or YAML under **Applications** ▢ **Export**.



Import

You can import the export file into a Mapbender installation under **Applications** ▢ **Import**.



5.2.8 Export/import/clone YAML application files over the console

Please go to [Application Export, Import & Cloning](#) to see the console commands. Find a few introductory words about what's possible with applications over the console below.

Export

Applications can be exported as .json or .yaml -file over the console.

A YAML file that has been exported over the console cannot be placed under `app/config/application` to be imported in a Mapbender installation. The YAML format that is produced by exporting over the console is different from the YAML format of the files under `app/config/application`.

Import

YAML files that have been exported over the user interface or console can be imported over the console.

Clone

Clone/Copy an existing application.

5.3 app/console commands

Mapbender provides commands that can be executed via the command line. Some of the commands are provided by the Symfony components, others belong to Mapbender.

Some useful commands are described in the following document.

Make sure you are in the correct directory (above the app directory)

- `mapbender/application` (installation via GitHub)
- `mapbender` (installation via package)

5.3.1 Help command

```
app/console
```

The parameter `--help` displays the help message for every command, for example:

```
app/console mapbender:user:create --help
```

5.3.2 Application Export, Import & Cloning

app/console mapbender:application:export

You can export an application as JSON- or YAML-file. In the command you have to add the Url title (slug) of the application and define the export file.

```
app/console mapbender:application:export mapbender_user_db --format=json > export.json
```

app/console mapbender:application:import

You can import an application from a JSON or YAML-file. Mapbender will automatically choose a new name if the name already exists.

```
app/console mapbender:application:import export.json

Imported 1 applications
* mapbender_user_db_imp2
```

app/console mapbender:application:clone

You can clone an existing application in the Application backend. This will create a new application with a *_imp* suffix as application name. In the example below, the name of the new application becomes *mapbender_user_yml_imp1*.

```
app/console mapbender:application:clone mapbender_user_yml
```

5.3.3 User administration

app/console mapbender:user:create

Command to create a new user on the command line. User name, email and password are mandatory. User name and email have to be unique.

```
app/console mapbender:user:create --help
app/console mapbender:user:create --password <password> --email <email> <name>
app/console mapbender:user:create --password mypassword123 --email max.mustermann@mapbender.org 'Max
↳ Mustermann'
```

Update user settings

Information for a user can be updated.

Following information can be updated:

- email
- password

The user name cannot be changed.

```
app/console mapbender:user:create --update --password <password> --email <email> <name>

app/console mapbender:user:create --update --password mypassword8910 --email max.mustermann@mapbender.
↳ org 'Max Mustermann'
```

app/console fom:user:resetroot

Command to create or update the root account. User name, email and password must be assigned for creation.

During the update, the unique assignment is made via the already existing ID, therefore all three parameters mentioned above can be changed.

```
app/console fom:user:resetroot
```

```
app/console fom:user:resetroot --username="root" --password="root" --email="root@example.com"
```

app/console mapbender:user:list

Command to list all existing users with their ID and user name and the time of creation.

```
app/console mapbender:user:list
```

```
User #3 name: max_mustermann since 2019-10-14 12:10:44
```

5.3.4 Database**app/console mapbender:database:upgrade**

Command to update the Mapbender database.

```
app/console mapbender:database:upgrade

Updating map element configs
Found 28 map elements
28/28 [=====] 100%
Updated 28 Map elements
Exiting now
```

app/console doctrine:database:create

The command is used only once during installation and creates the administration database for Mapbender. The database connection can be found in the parameters.yml file.

```
app/console doctrine:database:create
```

app/console doctrine:schema:create

The command is used only once during installation and creates the database schema, which means that the tables required by Mapbender are created.

```
app/console doctrine:schema:create
```

app/console doctrine:schema:validate

Validate whether that the database is up-to-date.

```
app/console doctrine:schema:validate
[Mapping] OK - The mapping files are correct.
```

5.3.5 Print

app/console mapbender:print:queue:next

The queued print is disabled by default because it requires some external integration setup. To run print jobs via the command line, the following parameter must be added to the parameters.yml file and set to TRUE to enable queued printing.

```
mapbender.print.queueable: true
```

Read more about the general characteristics of queued print at [Queued Print](#). Also [here](#)

The print assistant is then updated in the backend of Mapbender and two new lines appear: mode and queue. Mode is set to “queue” and queue is set to “global”, if the print jobs are expected to be accessible to all colleagues. The new tab “Recent jobs” (which shows your scheduled print jobs) appears in the print client pop-up window.

To run the jobs the following commands can be used.

```
app/console mapbender:print:queue:next
```

The command mapbender:print:queue:next executes the next print job in the queue. For a potentially infinite process, the following options can be set to 0.

```
app/console mapbender:print:queue:next --max-jobs=0 --max-time=0
```

Optionally you can set a limit for the number of jobs to process and the maximum time for a job.

- `--max-jobs=MAX-JOBS`
- `--max-time=MAX-TIME`

app/console mapbender:print:queue:rerun

This command reruns a print queue job. The ID for the job must be set.

```
app/console mapbender:print:queue:rerun 1

Starting processing of queued job #1
PDF for queued job #1 rendered to /data/mapbender/application/app/./web/prints/mapbender_
↳ 20191104103745.pdf
```

app/console mapbender:print:queue:dumpjob

This command dumps the queued print job from the database to JSON or YAML. The ID of the print job is required. This ID can be determined from the open print queue in the Mapbender application.

```

app/console mapbender:print:queue:dumpjob [options] [--] <id>

app/console mapbender:print:queue:dumpjob 2 > print_configuration.json

{
  "template": "a4portrait",
  "quality": "288",
  "scale_select": "25000",
  "rotation": "-20",
  "extra": {
    "title": "My Title"
  },
  "layers": {
    "0": {
      "type": "wms",
      "sourceId": "8",
      "url": "https://osm-demo.wherogroup.com/service?_SIGNATURE=31
→%3AIIHZNt0zPZhFG95dN3Q0zsizeDwA&TRANSPARENT=TRUE&FORMAT=image%2Fpng&VERSION=1.3.0&EXCEPTIONS=INIMAGE&
→SERVICE=WMS&REQUEST=GetMap&STYLES=&LAYERS=osm_OLSALT=0.3940783483836241&CRS=EPSG%3A25832&BBOX=363375.
→30907721,5626747.0157598,368124.31589362,5620823.2546257&WIDTH=512&HEIGHT=512",
      "minResolution": null,
      "maxResolution": null,
      "order": 0,
      "opacity": 1,
      "changeAxis": false
    },
    "1": {
      "type": "wms",
      "sourceId": "7",
      "url": "https://wms.wherogroup.com/cgi-bin/mapbender_user.xml?_SIGNATURE=26%3Atq6ae-
→UqhnZLMjiQLLrj-wChIOI&TRANSPARENT=TRUE&FORMAT=image%2Fpng&VERSION=1.3.0&EXCEPTIONS=INIMAGE&
→SERVICE=WMS&REQUEST=GetMap&STYLES=&LAYERS=Mapbender_User&OLSALT=0.6831931928241708&CRS=EPSG%3A25832&
→BBOX=363375.30907721,5626747.0157598,368124.31589362,5620823.2546257&WIDTH=2400&HEIGHT=1141",
      "minResolution": null,
      "maxResolution": null,
      "order": 0,
      "opacity": 0.85,
      "changeAxis": false
    },
    "2": {
      "type": "wms",
      "sourceId": "7",
      "url": "https://wms.wherogroup.com/cgi-bin/mapbender_user.xml?_SIGNATURE=26%3Atq6ae-
→UqhnZLMjiQLLrj-wChIOI&TRANSPARENT=TRUE&FORMAT=image%2Fpng&VERSION=1.3.0&EXCEPTIONS=INIMAGE&
→SERVICE=WMS&REQUEST=GetMap&STYLES=&LAYERS=Mapbender_Names&OLSALT=0.6831931928241708&CRS=EPSG%3A25832&
→BBOX=363375.30907721,5626747.0157598,368124.31589362,5620823.2546257&WIDTH=2400&HEIGHT=1141",
      "minResolution": null,
      "maxResolution": null,
      "order": 1,
      "opacity": 0.85,
      "changeAxis": false
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "width": 1920,
    "height": 913,
    "center": {
        "x": 365749.81248542,
        "y": 5623785.1351928
    },
    "extent": {
        "width": 4749.006816409994,
        "height": 5923.761134099215
    },
    "overview": {
        "layers": {
            "0": "https://osm-demo.wherogroup.com/service?_signature=31
↪%3AIIHZNt0zPZhFG95dN3Q0zsizaDwA&TRANSPARENT=TRUE&FORMAT=image%2Fpng&VERSION=1.3.0&EXCEPTIONS=INIMAGE&
↪SERVICE=WMS&REQUEST=GetMap&STYLES=&LAYERS=osm&CRS=EPSG%3A25832&BBBOX=350757.32820012,5616536.5348653,
↪377637.46662208,5629318.6006879&WIDTH=250&HEIGHT=125"
        },
        "center": {
            "x": 364197.3974111,
            "y": 5622927.5677766
        },
        "height": 78125,
        "changeAxis": false
    },
    "mapDpi": 90.714,
    "extent_feature": {
        "0": {
            "x": 362505.8322437394,
            "y": 5625755.14826519
        },
        "1": {
            "x": 366968.4389051802,
            "y": 5627379.404257199
        },
        "2": {
            "x": 368994.48453732743,
            "y": 5621812.889632087
        },
        "3": {
            "x": 364531.877875887,
            "y": 5620188.63364008
        },
        "4": {
            "x": 362505.8322437394,
            "y": 5625755.14826519
        }
    },
    "userId": null,
    "userName": null,
    "legendpage_image": {
        "type": "resource",
        "path": "images/legendpage_image.png"
    }
}

```


app/console mapbender:print:runJob

Command to run a print job from a saved job definition. The JSON file created with the previously described command (app/console mapbender:print:dumpjob) will create a pdf print output.

```
app/console mapbender:print:runJob print_configuration.json /tmp/print.pdf
```

app/console mapbender:print:queue:repair

If a print job in the queue has crashed, e.g. a WMS service is not accessible, printing cannot be performed. The command resets the status of the print jobs so that they can be executed again.

```
app/console mapbender:print:queue:repair
```

app/console mapbender:print:queue:clean

This command purges old jobs from the print queue (database and files). This includes created PDFs as well as corresponding database entries for the print jobs which are listed in the table called "mb_print_queue". With the command the expiring age can be set, for example, 20 can be used to delete all jobs older than 20 days.

```
app/console mapbender:print:queue:clean 20
```

```
Print queue clean process started.  
Deleted 0 print queue item(s)
```

app/console mapbender:print:queue:gcfiles

gcfiles means "garbage collection files". This command deletes unreferenced files from print queue storage path. This can happen, for example, if a job is deleted from the database or the file path to the PDFs is no longer up-to-date.

```
app/console mapbender:print:queue:gcfiles
```

```
No unreferenced local files found
```

5.3.6 Mailer

app/console debug:swiftmailer

Command displays the configured mailer(s)

```
app/console debug:swiftmailer
```

5.3.7 Server

app/console server:run

This command runs the PHP's built-in web server. The terminal displays that the server is running on the given local address (<http://127.0.0.1:8000>). In this mode you can work locally with Mapbender.

Quit the server with CONTROL-C.

```
app/console server:run

[OK] Server running on http://127.0.0.1:8000
// Quit the server with CONTROL-C.
```

app/console server:start

The command starts the PHP's built-in web server in the background.

In the terminal appears a message saying that the web server is listening on the displayed address (<http://127.0.0.1:8000>)

```
app/console server:start

[OK] Web server listening on http://127.0.0.1:8000
```

app/console server:stop

The command stops the PHP's built-in web server. A message appears in the terminal that the server with this specified address was stopped (<http://127.0.0.1:8000>).

```
app/console server:stop
```

app/console server:status

Outputs the status of the built-in web server for the given address.

```
app/console server:status
```

Clear cache

app/console cache:clear

The command clear the cache directory.

Dev:

```
app/console cache:clear --env=dev
```

Prod:

```
app/console cache:clear --env=prod --no-debug
```

5.3.8 WMS Services

app/console mapbender:wms:add

Adds a new WMS Source to your Mapbender Service repository.

```
app/console mapbender:wms:add https://osm-demo.wherogroup.com/service?VERSION=1.3.0&Service=WMS&
↳request=getCapabilities

* <empty name> OpenStreetMap (WhereGroup)
* * osm OpenStreetMap
* * osm-grey OpenStreetMap (grey scale)
Saved new source #76
```

app/console mapbender:wms:parse:url

Command to parse a GetCapabilities document by url. The command can be used to validate a WMS Url.

```
app/console mapbender:wms:parse:url --validate https://osm-demo.wherogroup.com/service?VERSION=1.3.0&
↳Service=WMS&request=getCapabilities
```

app/console mapbender:wms:reload:file

Command to reload (update) a WMS source from given file.

```
app/console mapbender:wms:reload:url 76 /var/www/html/service.xml
```

The following additional options are possible:

- **--deactivate-new-layers** If set, newly added layers will be deactivated in existing instances. Deactivated layers are not visible in the frontend.
- **--deselect-new-layers** If set, newly added layers will be deselected in existing instances. Deselected layers are not visible on the map by default, but appear in the layer tree and can be selected by users.

app/console mapbender:wms:reload:url

Command to reload (update) a WMS source from given url.

```
app/console mapbender:wms:reload:url 76 https://osm-demo.wherogroup.com/service?VERSION=1.3.0&
↳Service=WMS&request=getCapabilities
```

The following additional options are possible:

- **--user=USER** Username (basicauth) [default: ""]
- **--password=PASSWORD** Password (basic auth) [default: ""]
- **--deactivate-new-layers** If set, newly added layers will be deactivated in existing instances. Deactivated layers are not visible in the frontend.
- **--deselect-new-layers** If set, newly added layers will be deselected in existing instances. Deselected layers are not visible on the map by default, but appear in the layer tree and can be selected by users.

app/console mapbender:wms:show

Command to displays layer information of a persisted WMS source. You have to parse the ID of the WMS Source to get the information.

```
app/console mapbender:wms:show 76

Source describes 3 layers:
* <empty name> OpenStreetMap (WhereGroup)
* * osm OpenStreetMap
* * osm-grey OpenStreetMap (grey scale)
```

app/console mapbender:wms:validate:url

Command to check the accessibility of the WMS data source. The available layers are listed, if the service is accessible.

```
app/console mapbender:wms:validate:url "https://osm-demo.wherogroup.com/service?VERSION=1.3.0"

WMS source loaded and validated
Source describes 3 layers:
* OpenStreetMap (WhereGroup)
* OpenStreetMap
* OpenStreetMap (grey scale)
```

5.3.9 Other

app/console mapbender:source:rewrite:host

Use this command to update the hostname in the source URLs, eliminating the need to reload service capabilities.

```
mapbender:source:rewrite:host [options] [--] <from> <to>
```

As usual, the [Help command](#) shows more options.

Example to update the hostname:

```
app/console mapbender:source:rewrite:host "http://osm-demo.wherogroup.com" "https://osm-demo.wherogroup.
↪com"

3 modified urls in WMS source #5 / OpenStreetMap (OSM) Demo WhereGroup
Summary:
1 sources changed
3 urls changed
4 sources unchanged
14 urls unchanged
```

app/console mapbender:config:check

Command to check the system configuration and mapbender requirements. Useful command to determine whether dependencies are compliant and database access works.

```
app/console mapbender:config:check
```

Hint: Please note that config:check will use the php-cli version. The settings may be different from your webserver PHP settings. Please use `php -r 'phpinfo();'` to show your PHP webserver settings.

The following requirements are checked and displayed:

- Database connections
- PHP Version
- System requirements
- Asset Folders
- FastCGI
- Apache modus (rewrite)
- PHP ini
- loaded PHP extensions
- Directory permissions

app/console mapbender:version

The command outputs the current version of Mapbender.

```
app/console mapbender:version
```

```
Mapbender 3.3.4
```

app/console debug:config

Command lists all registered bundles (packages) and, if available, their aliases.

```
app/console debug:config
```


FAQ - FREQUENTLY ASKED QUESTIONS

6.1 General

6.1.1 app.php and app_dev.php

Q: What purpose do app.php and app_dev.php have?

A: For productive use you'll use the app.php file. Only if you develop something (TWIG-files, CSS or JS-files) or for debugging, you'll open Mapbender with the app_dev.php in developer mode. This is because the developer mode provides more information and error messages.

For further information on the modes, please take a look at the chapter *Production and Development environment and Caching: app.php and app_dev.php*.

6.1.2 Cache

Q: What is the cache and when do I have to clear it?

A: The cache is a small storage area, where Mapbender accesses frequently needed application data. Cache clearing can be useful to refresh your Mapbender installation. You'll delete the contents of the mapbender/app/cache/ directory, not the folder itself. In detail, the prod and - if present - the dev directory.

It is no problem to delete these directories. When you run Mapbender again, new files will be stored again in the cache directory.

For further information on the cache, please take a look at the chapter *Production and Development environment and Caching: app.php and app_dev.php*.

6.1.3 Services and their usage in applications

Q: I would like to know in which applications a specific WMS service is registered. Is there a way to achieve that?

A: For users with permission, this information is provided in the Backend:

- Go to Sources and find your source,
- Select it with a click on the button Show Metadata, then go to the tab Applications,
- Here, you can see in which application the source is loaded. Moreover, you can see whether the source is integrated as shared or private Instance and whether it is activated or inactive.

- Via click on the application title or the source instance, you can switch to the related layerset or instance setting.

If you prefer SQL, you can also take the WMS ID from the Source, replace the <id_of_the_wms> with it in the following SQL and run:

```
SELECT mb_core_application.* FROM mb_core_application, mb_core_layerset, mb_core_sourceinstance, mb_wms_
wmsinstance, mb_wms_wmssource, mb_core_source WHERE
-- applications and their layersets
mb_core_application.id = mb_core_layerset.application_id AND
-- layersets and their instances
mb_core_layerset.id = mb_core_sourceinstance.layerset AND
-- layerset-instances and wms-instance
mb_core_sourceinstance.id = mb_wms_wmsinstance.id AND
-- wms-instance and wms-source
mb_wms_wmsinstance.wmssource = mb_wms_wmssource.id AND
-- wms-source and mb-core source
mb_wms_wmssource.id = mb_core_source.id AND
mb_core_source.id = <id_of_the_wms>;
```

6.2 Performance

6.2.1 My application cannot be duplicated

Q: I made a highly complex application and want to duplicate it, but it does not work. Why is that?

A: A possible reason for this is that PHP does not allow a workflow with large files. The problem occurs especially in FastCGI. To fix this, you need to adjust the PHP parameter MaxRequestLen (you can do that in the FastCGI configuration).

```
# mod_fcgi.conf (Windows)
# set value to 2 MB
MaxRequestLen = 2000000

# fcgid.conf (Linux)
# set value to 2 MB
MaxRequestLen 2000000
```

Simultaneously, you should check if the following PHP values are set in your php.ini:

```
max_execution_time = 240
memory_limit = 1024M
upload_max_filesize = 2M
```

6.2.2 Adjusting WMS tiling values for ImageExport and PrintClient

Q: My WMS service does not return images in my ImageExport or my PrintClient. Consequently, my output file does not contain layers from the service. What could be causing this?

A: This can have different reasons. Under some circumstances, the requested pixel dimension of a WMS can get too large.

In this case, add the following parameter to your parameters.yml file - note that you have to adjust the default value according to your service.


```
mapbender.imaageexport.renderer.wms.max_getmap_size: 8192
```

What does it do? The parameter sets the largest possible WIDTH= and HEIGHT= parameter values for WMS requests generated from the ImageExport and PrintClient elements. Moreover, the maximum resolution is defined in the MaxWidth or MaxHeight fields of the GetCapabilities request for the service, therefore it's helpful to check the service request for a value to be entered. WIDTH= and HEIGHT= parameters can also be limited separately.

Adjust the width with this parameter and an individual value:

```
mapbender.imaageexport.renderer.wms.max_getmap_size.x:
```

Adjust the height with this parameter and an individual value:

```
mapbender.imaageexport.renderer.wms.max_getmap_size.y:
```

6.2.3 Working with large WMS Services with many layers

Q: When I try to use a WMS Service with many layers (> 100) into an application, the configuration of the *Layerset* only takes and presents an incorrect amount of layers. In addition, the wms instance cannot be saved. Why?

A: To solve the problem, navigate to the php parameter `max-input_vars`. It defines the number of possible input variables. The default value is 1000 (depending on the php version). For a WMS with many layers, the number of input values is higher than the default value. You have to change the parameter to a higher value (e.g. 2000).

```
;; 1000 (default)
max_input_vars = 1000
```

6.3 Installation

6.3.1 Attempted to call function "imagecreatefrompng"

Q: I get an error while trying to generate a print output. I have looked into Mapbender's logfile (app/logs/prod.log) and found something like this:

```
CRITICAL - Uncaught PHP Exception Symfony\Component\Debug\Exception\UndefinedFunctionException:
"Attempted to call function "imagecreatefrompng"
from namespace "Mapbender\PrintBundle\Component"."
at /srv/mapbender-starter/application/mapbender/src/Mapbender/PrintBundle/Component/PrintService.php
↪ line 310
```

A: Please make sure you have installed the php-gd library, because it is necessary for this action. We recommend the installation of the extension before installing Mapbender. However, it is always possible to install it afterwards, on Linux-based systems like this:

```
sudo apt-get install php-gd
```

6.3.2 SSL certificate problem

Q: How can I fix my SSL certificate problem?

A: When you get an SSL certificate problem error on loading or updating an OGC WMS data source on Windows-based Mapbender servers, you have to update your `cacert.pem` file and refer to it in your `php.ini`.

The problem can occur while accessing a service via https. It looks like this:

```
cURL error 60: SSL certificate problem: unable to get local issuer certificate
```

Note: The file `cacert.pem` that is available online lists all trusted certificate authority. `cacert.pem` is base64-encoded with a definition for all trusted certificate authorities; you can download it from <https://curl.haxx.se/docs/caextract.html>.

The error above occurs if the file is not referenced in `php.ini` or if `cacert.pem` is not up-to-date.

How to refer to `cacert.pem` in `php.ini`:

```
curl.cainfo="C:[your path]\cacert.pem"

openssl.cafile="C:[your path]\cacert.pem"
```

If you use an individual self-signed certificate, you can add the information of your certificate authority to `cacert.pem`.

Find further information in the PHP documentation at: <https://www.php.net/manual/en/curl.configuration.php>

6.3.3 Deprecation Notices at composer or bootstrap script

Q: I get a deprecation warning when I call bootstrap or composer update:

```
Deprecation Notice: The callback ComposerBootstrap::checkConfiguration declared at
/srv/mapbender-starter/application/src/ComposerBootstrap.php accepts a Composer\Script\CommandEvent
but post-update-cmd events use a Composer\Script\Event instance.
Please adjust your type hint accordingly, see https://getcomposer.org/doc/articles/scripts.md#event-
classes
in phar:///srv/mapbender-starter/composer.phar/src/Composer/EventDispatcher/EventDispatcher.php:290
```

A: This depends on the PHP version the system is running on and occurs on PHP versions < 7. Depending on the Mapbender release, we recommend different PHP versions that do not trigger the notices.

6.4 Development

6.4.1 Manual updates of modules

Q: How can I checkout a specific branch of the Mapbender module and test it? How can I revert this again? Does Composer help me with that?

A: Alternative 1 (via Git): Go in the directory `application/mapbender` and checkout the specific branch. After your tests, checkout the original branch again. Do not forget to clear the Symfony cache directory.

Alternative 2 (via Composer): Change the entry in composer: "mapbender/mapbender": "dev-fix/meinfix" and do a Composer Update. Keep in mind that with that step all other vendor packages will be updated. To go back, specify the original branch. In addition go back to application/mapbender and checkout the original branch.

6.5 Oracle

6.5.1 Adjustments for Oracle database - point and comma

Q: I get an error when I run doctrine:schema:create on Oracle. Why? The error message is:

```
[Doctrine\DBAL\Exception\DriverException]
An exception occurred while executing
'CREATE TABLE mb_wms_wmsinstance (id NUMBER(10) NOT NULL,
                                [...]
                                PRIMARY KEY(id))':
ORA-01722: Invalid number
```

A: Probably Oracle can't handle the decimal separators and expects a comma instead of a point (e.g. 1,25 instead of 1.25). This can be adjusted with the following snippet at the end of the config.yml (clear cache afterwards).

```
services:
  oracle.session.listener:
    class: Doctrine\DBAL\Event\Listeners\OracleSessionInit
    tags:
      - { name: doctrine.event_listener, event: postConnect }
```

This is a relation to a service-class provided by Doctrine. After the connection to Oracle, this class sets Session-Variables (ALTER SESSION) so that PHP and Oracle can work together in a better way.

Reasons might be: Language and regional settings of the operating system (for example Windows), settings of the Oracle-client, settings done during the installation of Oracle.

6.5.2 Rights management in Oracle database

Q: Which rights does the Mapbender user need for the Oracle database?

A: Mapbender needs permission to:

- Create Sequence
- Create Session
- Create Table
- Create Trigger
- Create View

6.5.3 The access to an Oracle database is too slow

Q: Mapbender seems to have a poor performance while accessing Oracle databases. I noticed this because queries need more time than usual. How can I accelerate the process?

A: There are two parameters in `php.ini` which may tweak the performance of Mapbender with Oracle databases: `oci8.max_persistent` and `oci8.default_prefetch`. Adjust these parameters to:

```
oci8.max_persistent = 15
oci8.default_prefetch = 100000
```

Furthermore, change the respective persistent database connection parameter in `config.yml` to true.

```
persistent=true
```

ARCHITECTURE

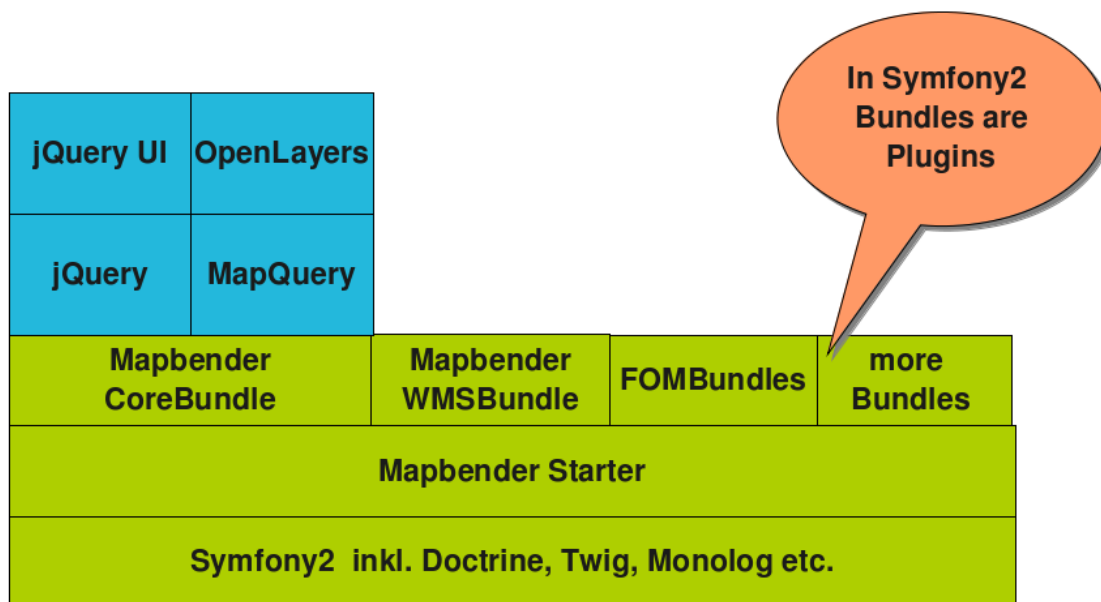
7.1 Components

Mapbender is made up of different components. On the server side we use Symfony as a framework which comes along with powerful components like Doctrine, Twig, Monolog and more.

On the client side we use OpenLayers, MapQuery and jQuery & jQuery UI.

We have a Mapbender core bundle with the Mapbender basic functionalities. And more Mapbender bundles which are optional.

We offer a Mapbender Starter package. With the Mapbender Starter package you can set up a Mapbender installation easily.



7.1.1 Symfony

Symfony is a full object oriented PHP Web Development Framework. It builds blocks for all modern web application needs. It is a collection of software and a development methodology. It relies on the philosophy of building blocks. It is optimized for speed. It uses Byte Code Cache.

Here is a list of some components Symfony offers:

- Symfony config.php to check the prerequisites
- Symfony Profiler
- Database abstraction via Doctrine
- User authentication, authorization
- Templating via Twig
- Translation using xliiff-files
- Logging via Monolog
- Security

The project has a very good documentation:

- TheBook <http://symfony.com/doc/current/book/index.html>

7.1.2 OpenLayers

OpenLayers is a powerful software for web maps. It supports a lot of data sources and functionality.

Read more about OpenLayers at <http://openlayers.org/>

You find example applications with OpenLayers at <http://dev.openlayers.org/examples/>

7.1.3 jQuery and jQuery UI

jQuery is a feature-rich JavaScript library. jQuery UI is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.

Read more about jquery at <http://jquery.com>

Read more about jquery UI at <http://jqueryui.com/>

7.1.4 Mapbender

Mapbender is a collection of bundles. Only the MapbenderCoreBundle and the FOMBundles are mandatory.

There are optional bundles like:

- WMSBundle
- WMTSBundle
- WMCBundle
- MonitoringBundle

CoreBundle

The Mapbender CoreBundle is the base bundle for Mapbender. It offers base classes for applications, elements, layers and more.

It provides jQuery, jQuery UI, OpenLayers and MapQuery for all other Mapbender bundles.

7.1.5 Mapbender Starter

Mapbender Starter is Symfony demo project which uses the Mapbender bundles to showcase a Mapbender application.

It contains demo applications which are defined in the mapbender.yml with WMS, WMTS. It provides a web interface with authentication which provides the possibility to create applications, create users/groups and build up a service repository.

Mapbender Starter can be used as a boiler template to start Mapbender projects.

Find the GitHub Repository here: <https://github.com/mapbender/mapbender-starter>

7.1.6 External Repositories

You find more code connected to Mapbender at GitHub, which is not part of the main project. Other providers can offer Bundles for Mapbender like the DesktopIntegrationBundle which is provided by [WhereGroup](#) and sponsored by customers.

WhereGroup offers Bundles for Mapbender at: <https://github.com/WhereGroup>

7.2 Directory structure in Mapbender

7.2.1 app

This directory contains:

- the php-Cache (app/cache)
- the logs (app/logs)
- the configurations (app/config)
- the applicationkernel (app/AppKernel.php) (this is called by the FrontendControllers and controls the whole application)
- the Autoloading (autoload.php)
- the application specific resource directory (Resources)
- the command line application for maintaining and management tasks (app/console)

app/config

Basic configuration files of Mapbender are placed in the app/config directory. Two files are of particular importance:

- parameters.yml
- config.yml

More Information: [YAML Configuration \(Configuration and Application files\)](#).

app/config/applications

The directory app/config/applications contains all applications that are defined in a YAML file.

More Information: [YAML Configuration \(Configuration and Application files\)](#).

7.2.2 bin

Here are symlinks to the following binaries placed:

- apigen
- composer
- coveralls
- doctrine
- doctrine.php
- phantomjs
- phing
- phpunit

7.2.3 documentation

Folder for this documentation.

7.2.4 fom

Directory of the [FOM submodule](#).

7.2.5 mapbender

Directory of the [Mapbender submodule](#). Provides the mapbender-specific bundles and the Mapbender code.

mapbender/...../translations

Directory: mapbender/src/Mapbender/CoreBundle/Resources/translations/

The translations are stored in [XLIFF textfiles](#). Every language needs an xliiff-file like messages.en.xlf for the English translation.

7.2.6 owsproxy

Directory of the [OWSProxy submodule](#).

7.2.7 vendor

Directory for external libraries (loaded by composer) and further Mapbender modules (a.o. Digitizer, Mapbender-Icons).

7.2.8 web

This directory has to be published by the webserver. The ALIAS has to refer to this directory.

It controls:

- the FrontendController (PHP-Script, which can be called). These are **app.php** for the productive-system and **app_dev.php** for the development version. The development version contains the profiler for performance tests and more.
- this directory contains the static resources like css, js, favicon etc.

web/bundles

- here the static resources of the single bundles are stored.
- the following command copies the resources from the bundles to the folder.

```
app/console assets:install --symlink web
```

- **Notice:** if you use Windows you can't create symbolic links and therefore you have to run the command (**app/console assets:install web**) after every change in the code to copy the files to the directory.

7.2.9 src

- directory for applications specific bundles (similar to the former x-directories in Mapbender 2.x)

7.2.10 vendor

- directory where all the Bundles which are used from Symfony are found. Resources are used by Symfony using the Autoloading.

7.3 Concepts

This chapter will introduce you to the main concepts of Mapbender. After reading you should have an understanding of the parts Mapbender is made of and how they interact.

7.3.1 Application

The application is a single Mapbender configuration which is usually used as an interactive web map. It's what was called a GUI in Mapbender 2.

Every application consists of several components:

- Element
- Layersets
- Application Template

7.3.2 Element

Elements are the building blocks of every application. Each element provides some functionality and can interact with other elements. The map element is probably the element you face most of the time as it provides all map viewing capabilities.

Each element consists of four parts itself:

- PHP class - describes the element with it's capabilities and also can provide an Ajax callback point, so that the client side widget (see below) can execute database queries and display the result for example.
- JavaScript widget - this is the client side part of an element. It's everything you do and interact with on your screen. Using Ajax, it can call it's server-side counterpart to do things like database queries.
- Template - HTML the element is using. In the most basic version, this would just be a DIV, but it can be as complex as is needed.
- CSS - of course, most elements want some style, so they may provide their own.

7.3.3 Application Template

Each application is an HTML page, and the application template renders the basic layout of that page. Each application can have a different template as needed. Think of HTML templates specialised for mobile viewing.

7.3.4 Frontend

The frontend is basically the application view of your Mapbender installation. This includes the application list and each application view. It's what your users will interact with.

7.3.5 Backend

The backend is where you configure your Mapbender installation. It's called the manager and allows your admins to manage users, roles, services, applications and elements.

7.4 Translation in Mapbender

Mapbender uses the translator service which is a Symfony component. You get more information at the [Symfony Translation Documentation](#).

In the code you use the function **trans** to translate a text into another language.

Example how the function **trans** can be used in a Twig template:

```
{% block title %}{{ application.title | trans }}{% endblock %}
```

or

```
{% trans %}{{ application.title }}{% endtrans %}
```

Example for PHP:

```
echo $translator->trans('Hello World');
```

7.4.1 yml-files for translations

The translations can be stored in different formats. We use yml-format for Mapbender.

We use place holder for every text e.g. **mb.core.featureinfo.popup.btn.ok**. Like this you can define different translations for the same word which occurs in different modules.

We translate the place holder to different languages. English is the default language that we provide. It is also defined as the fallback language in the parameters.yml file. The fallback language will be used if you define a language in parameters.yml that does not exist.

This is how a translation file messages.de.yml for German translation could look like.

```
mb:
  core:
    featureinfo:
      error:
        noLayer: 'Informationsebene existiert nicht.'
        unknownoption: 'Unbekannte Option %key% für %namespace%.%widgetname%.'
        noresult: 'kein Ergebnis'
      popup:
        btn:
          ok: Schließen
      class:
```

(continues on next page)

(continued from previous page)

```

    title: Information
    description: Information
    tag:
      featureinfo: Information
      feature: Objekt
      info: Info
      dialog: Dialog
    aboutdialog:
      content:
        versionprefix: Version
        learnmore: 'Lernen Sie Mapbender kennen '
        linktitle: 'Besuchen Sie die offizielle Mapbender Webseite.'
        website: '- zur Webseite'
    .....

```

Notice: Each time you create a new translation resource you have to clear your cache.

```
app/console cache:clear
```

7.4.2 How can you activate translation?

Mapbender is automatically adjusted to your browser's language. Moreover, you can set fallback language options in the configuration file **app/config/parameters.yml**. If a translation from your browser's default language is not translated in Mapbender yet, it will fall back to the predefined language instead. We recommend to set English and/or German as fallback options.

```

fallback_locale: en
locale:          en

```

Check whether translations (yml-files) for your language exist

- mapbender/src/Mapbender/CoreBundle/Resources/translations/
- mapbender/src/Mapbender/ManagerBundle/Resources/translations/
- mapbender/src/Mapbender/PrintBundle/Resources/translations/
- mapbender/src/Mapbender/WmcBundle/Resources/translations/
- mapbender/src/Mapbender/WmsBundle/Resources/translations/
- fom/src/FOM/CoreBundle/Resources/translations/
- fom/src/FOM/ManagerBundle/Resources/translations/
- fom/src/FOM/UserBundle/Resources/translations/
- ...

7.4.3 Create yml-files for your language

If your language is not translated yet, it is easy to add a new language.

- Check the translation directories and create a new file by copying the English locale (messages.en.yml)
- translate
- set locale in your parameters.yml to the new language
- clear your cache
- adjust your browser language to the translated language - Mapbender will be translated automatically.
- if everything is fine with your new language, give the files to the Mapbender community - best would be a pull request on GitHub in the Mapbender repository.

7.4.4 Naming conventions and locations

Symfony looks for translation files in the following directories in the following order:

- the <kernel root directory>/Resources/translations
- the <kernel root directory>/Resources/<bundle name>/translations
- Resources/translations/ directory of the bundle.

Bundle translations can overwrite translations of the other directories.

Naming

The naming convention is **domain.locale.loader**.

- domain - we use the default domain messages
- locale - locale that the translations is made for (e.g. de, de_DE);
- loader - defines the loader to load and parse the file. We use YAML.

7.4.5 Share your translations with the Mapbender community!

Supporting more and more languages would be great for Mapbender. The Mapbender project would be happy if you could share your translations with the community.

This is what you have to do:

- Option 1: send the new yml-files for your language to the Mapbender developer (mapbender@osgeo.org) or
- Option 2: create a pull request on GitHub.

We prefer option 2.

How to translate and make a pull request

Repositories

- <https://github.com/mapbender/mapbender/>
- <https://github.com/mapbender/mapbender-digitizer/>
- <https://github.com/mapbender/fom>
- <https://github.com/mapbender/data-manager/>
- <https://github.com/mapbender/data-source>
- <https://github.com/mapbender/map-tools/>

Since Git is a distributed versioning system, it is very convenient for each developer/contributor to have a personal public copy of the “official” repository (also known as fork).

Web hosting services like GitLab or GitHub provide this option if you visit the main code repository and press the button “Fork”. This way the developer can make changes to a personal isolated repository. Then one can ask the rest of the developers to review the code and merge accordingly through a “pull request”.

Github - editing on GitHub

- you can edit files directly on GitHub.
- navigate to the file,
- edit the file,
- save changes and create a new branch for this commit and start a pull request.

git - working on the command line

On Linux systems get the source code locally using:

```
git clone https://github.com/your_id/mapbender
```

In order to be able to get and send changes to your public repository, you need to link your local copy to your public copy. This is done automatically for you when you “git clone”. The repository that you cloned from has the alias “origin”.

In order to be able to get changes that others do to the main repository, you need to manually link to that using:

```
git remote add upstream https://github.com/mapbender/mapbender
```

On MS Windows systems, install TortoiseGit, which extends Windows Explorer to include git commands.

1. The first thing you should do when you install Git is to set your user name and e-mail address.

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

```
cd mapbender
```

2. Pull any updates from upstream project (master is the equivalent of subversion trunk)

```
git pull upstream master
```

optionally check to see what has changed.

```
git diff messages.de.yml
```

3. add the changes into stage area

```
git add messages.de.yml
```

4. commit changes locally

```
git commit -m "changed translation"
```

5. send the changes to your public repository

```
git push origin master
```

At this point you can let others know that you have some changes that you want to merge, so you can use the button “Pull Request” on GitLab or GitHub. Or you can continue until you feel ready to share your changes :)

6. Last step: pull request

In order to merge your work with the main repository, you have to make a pull request.

You can do it by logging in your github account and go to the branch you changed. Click on the New pull request green button. The changes you made previously while appear.

You can review and comment your request before submitting. To submit, click on the Create pull request green button. Then, you’re done ! Good job !

More information about Github pull request here: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>

Working with files:

to add a file

```
cd <dir>
```

create a file

```
git add <file>
git commit -m "commit message"
git push origin master
```


DEVELOPMENT

8.1 Introduction

This book is targeted at Mapbender developers and will cover useful topics not needed by administrators or users of Mapbender installations.

8.1.1 Things to consider

There are a couple of things you should be familiar with in order to contribute to Mapbender:

- Object-Orientated PHP: We're using PHP which offers full object orientation.
- Symfony: This is what we build upon. So read [The Book](#) to learn more about Symfony.
- JavaScript: We use jQuery a lot and especially the jQuery UI widget factory. These are essential to understand to write maintainable JavaScript code.

8.1.2 Installation

The installation procedure from Git is described under [Git-based installation](#).

8.1.3 Modules and bundles

Please refer to the respective sections in the *CONTRIBUTING.md* guide to understand what [modules](#) and what [bundles](#) are, and how the latter can be [created](#).

8.1.4 Twig

Symfony follows the template approach and we use this in Mapbender. Symfony uses a templating engine to generate HTML, CSS or other content. A template is a text file that can generate any text based format like HTML, XML. It is used to express presentation and not programm logic. You can use them to create a layout. You can create a base layout and then overwrite or append any of your layout blocks with individual templates. Read more about Templates in Mapbender at [How to create your own Template?](#) or in the [Contributing Guide](#) and find a good introduction about Twig in the [Symfony Template documentation](#).

8.1.5 Getting Help

Mailinglists:

- [Mapbender-Developer and -User mailinglist](#)

Libraries and frameworks:

- [Symfony framework](#)
- [PHPUnit documentation](#)
- [Composer documentation](#)
- [General GitHub documentation](#)
- [GitHub pull request documentation](#)

8.1.6 Topics

Controllers

This pages gives a quick overview of the controllers used in a Mapbender application.

The Front Controller - Using Routes

In Symfony, each HTTP request goes trough the (one-and-only) front end controller (app.php in the web directory) which determines the controller function to pass it to.

The mapping from request path to controller function is basically done in the configuration, where the routing.yml defines these mappings - called routes - or imports their definitions from bundles (or other files).

To get an overview off all defined routes by using the console command.

```
cd mapbender/application
app/console router:debug

[router] Current routes
Name                                     Method Pattern
_assetic_30d3bc4                       ANY      /css/30d3bc4.css
_assetic_30d3bc4_0                     ANY      /css/30d3bc4_part_1_base_1.css
_wdt                                   ANY      /_wdt/{token}
_profiler_search                       ANY      /_profiler/search
_profiler_purge                       ANY      /_profiler/purge
_profiler_import                      ANY      /_profiler/import
_profiler_export                      ANY      /_profiler/export/{token}.txt
_profiler_search_results               ANY      /_profiler/{token}/search/results
_profiler                             ANY      /_profiler/{token}
_configurator_home                    ANY      /_configurator/
_configurator_step                    ANY      /_configurator/step/{index}
_configurator_final                   ANY      /_configurator/final
mapbender_manager_layer_index          GET      /manager/layers/{page}
mapbender_manager_group_index          GET      /manager/group/{page}
mapbender_manager_repository_index     GET      /manager/repository/{page}
mapbender_manager_application_index2   GET      /manager/application
mapbender_manager_application_index    GET      /manager/applications/{page}
```

(continues on next page)

(continued from previous page)

mapbender_manager_application_new	GET	/manager/application/new
mapbender_manager_application_create	POST	/manager/application
mapbender_manager_application_view	ANY	/manager/application/{id}
[and so on...]		

The command lists all routes with their names, allowed methods and URL pattern. To get more information about a particular route, give it's name to the command:

```
app/console router:debug mapbender_core_user_login

[router] Route "mapbender_core_user_login"
Name      mapbender_core_user_login
Pattern    /user/login
Class      Symfony\Component\Routing\CompiledRoute
Defaults   _controller: Mapbender\CoreBundle\Controller\UserController::loginAction
Requirements
Options    compiler_class: Symfony\Component\Routing\RouteCompiler
Regex      #^
           /user/login
           $#x
```

To learn more about routing, read the [Symfony Book](#).

Defining routes using annotations

In Mapbender we use a decentralized route definitions: Instead of writing each and every route in the routing.yml, we import their definition from the controller classes in the activated bundles. This has the advantage of having the definition with the controller function. This should usually be fine and can be - if need arises - easily overwritten by adapting the routing.yml. Using the Symfony with the SE bundles like Mapbender does, routes can therefore be written using annotation comments for each controller function. You can read about the annotation syntax over at the [Symfony documentation](#).

Mapbender Controllers

A Mapbender installation uses a particular set of controller classes and functions. This chapter will give a short list of these, so you can inspect them more easily.

Frontend

The frontend is basically "the application" (or GUI as it has been called in Mapbender 2 - and even there this term was incorrect). Each application is routed to the ApplicationController class of the CoreBundle:

```
/application/{slug} => Mapbender\CoreBundle\Controller\ApplicationController->applicationAction($slug)
```

Elements of an application can provide Ajax endpoints for their client side widgets. These are routed as follows:

```
/application/{slug}/element/{id}/{action} => Mapbender\CoreBundle\Controller\ApplicationController->
elementAction($slug, $id, $action)
```

Note: This controller calls the **httpAction** method if the element class and passes the \$action parameter and returns the response given by that function. So for the real magic for element Ajax behaviour take a look at the httpAction method of the elements.

Backend

The backend is handled by the ManagerBundle, which provides (will provide) a consistent backend for managing all aspects of a Mapbender application: applications, layers, elements, users, settings.

For each section an own controller class exists within this bundle:

- ApplicationController - Manage applications
- GroupController - Manage user groups
- LayerController - Manage layers
- RepositoryController - Manage the layer repository
- SettingsController - Manage common settings
- UserController - Manage users
- plus an ManagerController which provides some common functionality for the other controllers.

Each of the these controllers (right now work is going on within the ApplicationController) is a good example of what we think of as good kinda RESTful URLs.

Conventions for Mapbender

Code conventions

- variable names / way of coding
- Code documentation
- trans convention - where to put translation
- **document the steps on the way to a new functionality**
 - define the topic
 - create a ticket
 - create a workflow
 - discuss the workflow with the core team and find a final solution
 - do the programming
 - insert License
 - test
 - documentation in mapbender-documentation -> rst
 - close the ticket
- where to put a module/element
- naming vor files (referred to symfony convention)

Git branch conventions

We follow the Git Flow branching model (Read more about it in the [original document](#) describing it). Basically that boils down to having at least two branches:

- develop - for the daily work, always has the latest merged commits, and is equal to or ahead of the latest release
- master - only changes on new releases (right now and until the 3.0.1 release, some of our repositories don't have a master branch but will get one then again)

Furthermore there might be more branches, which must always be namespaced:

- feature/<name> - Used for developing new features (feature/printservice)
- hotfix/<name> - Used for making hot fixes for releases (hotfix/bug_123)
- release/<name> - Used for preparing releases, very short-lived (release/3.0.1)

Some Linux distributions have a package called git-flow which will provide easy git command shortcuts to use the merge/branch model of Git Flow without having to do everything by yourself (which is still possible and you should always know how Git Flow uses plain git to achieve things).

Layout conventions

What to keep in mind, when you create a layout

- naming conventions
- where to put the css
- where to put the twig
- where to put the images / should be possible to easy switch an image collection an get other buttons

Translation where

- conventions to put the files? Groß-Kleinschreibung/welche Übersetzungen werden generell gepflegt? en/de weitere?
- also have a look at *Translation in Mapbender* <../translation.rst>

Issue conventions

Issues (bugs and features) are administrated in the **mapbender**-repository at:

- <https://github.com/mapbender/mapbender/issues>

We create a milestone for every version of Mapbender:

- <https://github.com/mapbender/mapbender/milestones>

There are some rules you should keep in mind:

- **Write understandable tickets:**
- **Write your title so that the issue is already described in the title:**
 - Browser - Backend/Frontend - element - issue

- like: Firefox - Frontend - layertree - option visible is not handled in frontend
 - see ticket <https://github.com/mapbender/mapbender/issues/48>
- **Write comments with all necessary information:**
 - for bugs: describe step by step how the error can be reproduced
 - for features: describe feature and functionality
- When you create a new ticket do not assign it to a milestone or developer, if you are not sure
- **Add labels to your ticket**
 - Bug - describes a bug that occurs in a special version of Mapbender (add info about the version)
 - Feature - new feature
 - Enhancement - stands for feature enhancement
 - WIP - work in progress
- When you work on a ticket or close it please **assign a user and milestone**
- **When you close a ticket, please:**
 - add a comment in the ticket and **refer to the commit**,
 - refer to the documentation at <https://doc.mapbender.org> or a demo if possible.

Version conventions

The Mapbender version is defined by a four digit numbering system, separated by dots.

3.0.10.20

- The **first** digit is constant and represents the Mapbender software cycle.
- The **second** digit describes all new features and major changes in Mapbender, with the highest difficulty level of a update process.
- The **third** digit describes new features and minor changes, which can be easily updated.
- The **fourth** digit represents only bugfixes and micro changes.

Increase a digit means always a reset for all digits before. For example - 3.0.10.20 -> 3.1.0.0

This numbering system started with Mapbender version 3.0.0.0

Release

- check whether all tickets are done
- build a build - check documentation -> How to build a new Mapbender build
- update Roadmap and milestones
- update demo.mapbender.org
- write release mail (mapbender-user / mapbender-dev / major releases osgeo-announce)
- twitter

How to build a new Mapbender build

- Resolve and close all tickets for the relevant milestone: <https://github.com/mapbender/mapbender/milestones>
- Update <https://doc.mapbender.org/en/book/versions.html>
- Update Changelog.md for mapbender-starter, mapbender, owsproxy, fom.
- Update version number in parameters.yml.dist and push
- Update version number in composer.json
- **Tagging: Tag at Github. You have nice capabilities for creating good tags and descriptions.**
 - Mapbender
 - OWSPROxy
 - FOM
 - Mapbender-starter
 - Documentation
- **Create Pull requests to merge release branch into master**
 - Mapbender
 - OWSPROxy
 - FOM
 - Mapbender-starter
 - Documentation
- Clone the source-code from the release branch

```
git clone -b release/3.0.6 git@github.com:mapbender/mapbender-starter
```
- Change to the directory

```
cd mapbender-starter
```
- Bootstrap

```
sh bootstrap
```
- Change to the directory

```
cd application
```
- Generate the docs

```
bin/composer docs
```
- Create the ZIP/Tar.gz

```
bin/composer build
bin/composer build zip
```
- Move TAR.GZ and ZIP file to `/sites/www.mapbender.org/builds`

- Update symbolic links ("version".zip and "version".tar.gz and current.tar.gz and current.zip)
- Move current doc to docs.mapbender.org (get it from web/docs. Remove the api folder).
- Edit actual release link <https://mapbender.org/en/download> (english) and <https://mapbender.org/mapbender-herunterladen/> (german)
- Write release mail to mapbender-user and mapbender-dev
- Only for major releases write release mail to news_item@osgeo.org (see also <https://www.osgeo.org/about/faq/osgeo-mailing-lists/>)
- Twitter on <https://twitter.com/mapbender>
- Update <https://demo.mapbender.org> and <https://sandbox.mapbender.org>
- Create a version based installation <https://version.mapbender.org>

Documentation conventions

- have a look at [How to write Mapbender Documentation?](#)

Understanding Elements

Note: This guide is under complete restructuring. We will provide a new documentation in the Contributing Guide in our Git-Repository:

<https://github.com/mapbender/mapbender-starter/blob/release/3.0.6/CONTRIBUTING.md#elements>.

Parts of an Elements

PHP Class

TODO

Twig Template

Every Element must have an HTML element it is represented by. In the most basic case, this can be a simple DIV element, but this can be complex as needed.

For HTML generation Mapbender uses [Twig](#). A minimal twig template for an element would like this:

```
<div id="{{ id }}" class="mb-element mb-element-myclass"></div>
```

As you can see you need to set the id (generated by Mapbender) as well as the general mb-element class and a class specifically for your element.

JavaScript Widgets

Element widgets are build using the jQuery UI [widget factory](#). This ensures a common pattern for widget development and offers

- default options.
- constructors and (optional) destructors.
- private and public methods.

The basic skeleton looks like this:

```
(function($) {

// This is the widget factory. It will create an widget class "mbMyClass" in the jQuery object as well
↳ as an
// "mbMyClass" object in the "mapbender" namespace in the jQuery object (they are used differently). Be
↳ sure
// to use the "mb" prefix for your widget name as not to overwrite existing jQuery functions.
$.widget('mapbender.mbMyClass', {
    // This sets up the default options which can be overridden in the Mapbender configuration.
    // This will be moved into the PHP class at a later point.
    // The final options object can be accessed as "this.options".
    options: {
        foo:    'bar',
        answer: 42
    },

    // This attribute is private for your widget.
    var1: null,

    // Constructor, gets called on widget initialization.
    _create: function() {
        // Do everything needed for set up here, for example event handling
        this.element.bind('mbmyclassmagicdone', $.proxy(this._onMagicDone, this));
        this.element.bind('click', $.proxy(this._clickCallback, this));
    },

    // Destructore, here set to the jQuery empty function
    destroy: $.noop,

    // Public function, callable like "$('#element-13').mbMyClass('methodA', parameterA, parameterB)"
    methodA: function(parameterA, parameterB) {
        this._methodB(parameterA);
    },

    // Private function, only callable from within this widget
    _methodB: function(parameterA) {
        // The triggered signal will be named "mbmyclassmagicdone" (all lowercase)
        this._trigger('magicdone');
    },

    _onMagicDone: function() {
        alert("Oh, magic!");
    },

    _clickCallback: function(event) {
        var target = $(event.target);
```

(continues on next page)

(continued from previous page)

```
        var id = target.attr('id');
        // ...
    }
});
})(jQuery);
```

For event handling, `jQuery.proxy` is your friend to ensure that the callback is guaranteed to be executed in the right context:

```
// ...
this.element.click($.proxy(this._clickCallback, this));
// ...
```

This way, “this” inside the `clickCallback` method is the `this` given as the second parameter here (usually the widget instance) and not the HTML element which triggered the event. To access the HTML event, use the `target` property of the event passed as the argument to the `clickCallback` method.

Element to Element communication

There’s an active and a passive way to communicate with another widget. The first - active - way is to call a public method of the other widget. For that you need to select the widget’s HTML element with jQuery and call the method like this:

```
var otherElement = $('#element-13').mbMyClass('methodA', parameterA, parameterB);
```

This is standard jQuery UI stuff and pretty much self-explanatory. The more tricky question is how do you know the other’s HTML element? As you can see, using the id to select the element is preferred, but these ids are generated on the fly by Mapbender when the application is started, so you can’t assume that the id is always the same. Luckily you can pass a Element id in the configuration as the `target` options for an Element. This will be replaced with the run-time id of that target Elements’ HTML element for you, so that in your widget code you can access the right id as “`this.options.target`”.

```
$('#' + this.options.target).mbMyClass('methodA', parameterA, parameterB);
```

The passive way for communication is to subscribe to events of another target. You also need to know the HTML element, but you can now listen for the other widget to call your widget. This is done using standard jQuery events. Well, almost standard jQuery events:

If you use the “`_trigger`” method provided by the jQuery UI widget factory

How to create your own Element?

Note: This guide is under complete restructuring. We will provide a new documentation in the Contributing Guide in our Git-Repository:

<https://github.com/mapbender/mapbender-starter/blob/master/CONTRIBUTING.md>.

Mapbender offers an app/console command to create different elements:

- general elements
- buttons
- elements for map-click events
- elements for map-box events

Hint: The new generated element contains only a skeleton and has to be modified after generation.

The following example show the generation and modification of a map-click element.

The steps to create your own Element?

There are some steps you have to follow on the way to your own element.

- create your own bundle
- create an element via app/console
- edit your new element for your needs
- add the new element to the function *getElements()* to make it available from the backend

Use app/console to generate your own bundle

Find out more about the command with help:

```
app/console generate:bundle --help
```

```
app/console generate:bundle --namespace=Workshop/DemoBundle --dir=src
```

You have to answer some questions before the element will be created:

```
Bundle name [WorkshopDemoBundle]: WorkshopDemoBundle

Determine the format to use for the generated configuration.
Configuration format (yaml, xml, php, or annotation): annotation

To help you get started faster, the command can generate some
code snippets for you.

Do you want to generate the whole directory structure [no]? yes

Summary before generation
```

(continues on next page)

(continued from previous page)

You are going to generate a "Workshop\DemoBundle\WorkshopDemoBundle" bundle in "src/" using the "annotation" format.

Do you confirm generation [yes]? yes

Confirm automatic update of your Kernel [yes]? yes

Confirm automatic update of the Routing [yes]? yes

After this steps you new bundle is available at src. The bundle is already registered in the File AppKernel.php. In the file routing.yml you will find a new entry for the bundle.

Use app/console to generate a new element

Find out more about the command with help:

```
app/console mapbender:generate:element --help
```

Generate a new element with the following command:

```
app/console mapbender:generate:element --type "map-click" "Workshop\DemoBundle" MapKlick src
```

You will get a summary of actions

```
Summary of actions
- Your element WorkshopDemoBundle\Element\MapKlick has been created.
- The following files have been created:
  - PHP class (src/Workshop/DemoBundle/Element/MapKlick.php)
  - jQuery widget (src/Workshop/DemoBundle/Resources/public/mapbender.element.mapklick.js)
```

Edit your new element for your needs

Change the title and description in the php file

You will find several functions in the php file. Change the return value of the functions *getClassTitle()* and *getClassDescription()*.

```
public static function getClassTitle() {
    return "MapKlick";
}
```

```
public static function getClassDescription() {
    return "Generates an Url with the the mapklick coordinates added";
}
```

Register the new Element

You can register an element by adding it to the function `getElements()` in the file `src/Workshop/DemoBundle/WorkshopDemoBundle.php`. After creation of the bundle this function does not exist. You also have to refer to the `MapbenderBundle` and define that your extends the `MapbenderBundle`.

This will make the element available in the backend when you configure your application.

```
<?php

namespace Workshop\DemoBundle;

use Symfony\Component\HttpKernel\Bundle\Bundle;
use Mapbender\CoreBundle\Component\MapbenderBundle;

class WorkshopDemoBundle extends MapbenderBundle
{
    public function getElements()
    {
        return array(
            'Workshop\DemoBundle\Element\MapKlick'
        );
    }
}
```

Add the new element to an application

Create a new application and add your element to the new application. Note that the configuration for your generated element is done in YAML syntax. If you want to use the map-element as target you have to find out the id of the map-element (e.g. via inspector tool).

Change the action on Click event

When you generated a map-click element you get an event on click and an action. The action can be modified. Have a look in the JQuery widget file (`mapbender/src/Workshop/DemoBundle/Resources/public/mapbender.element.mapklick.js`).

You will find the function `_mapClickHandler()` that determines the coordinates from the click event and passes them to the function `_mapClickWorker()`. The new generated element will show the coordinates of the click event in an alert box.

You can modify the action of the function `_mapClickWorker()`.

Default definition of `_mapClickWorker()`

```
_mapClickWorker: function(coordinates) {  
    alert('You clicked: ' +  
        coordinates.pixel.x + ' x ' + coordinates.pixel.y +  
        ' (Pixel), which equals ' +  
        coordinates.world.x + ' x ' + coordinates.world.y +  
        ' (World).');  
}
```

modified `_mapClickWorker()` opens OpenStreetMap

Alternatively you could open a new window with an URL and add the coordinates as parameters. You can open OpenStreetMap and center to the coordinates of the click event.

<https://www.openstreetmap.org/export#map=15/50.7311/7.0985>

```
_mapClickWorker: function(coordinates) {  
    window.open('https://www.openstreetmap.org/export#map=15/' + coordinates.world.y + '/' +  
    coordinates.world.x);  
}
```

Request / Response Workflow

Note: We will move that documentation into the [Contributing Guide](#).

This document explains the request/response workflow used in Mapbender. Essentially there are three workflows you should be familiar with:

- Application workflow
- Element workflow
- Manager module workflow

Application Workflow

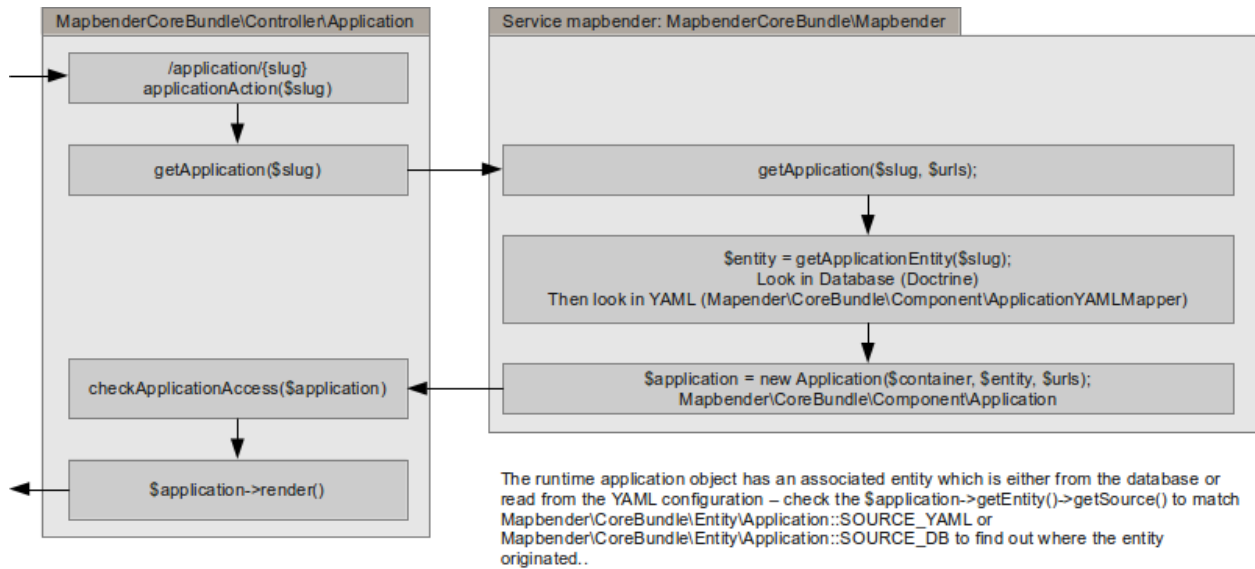
The standard application controller `/application/{slug}` is routed to the `Mapbender\CoreBundle\Controller\Application::applicationAction($slug)` function which does the following:

Basically the controller asks the mapbender service to look for an application with the given slug. The mapbender service first tries to find the application in the database and only if none can be found it looks in the applications defined in the YAML configuration.

The thus loaded entity of class `Mapbender\CoreBundle\Entity\Application` is tagged with its origin in the source attribute which is either `Mapbender\CoreBundle\Entity\Application::SOURCE_YAML` or `Mapbender\CoreBundle\Entity\Application::SOURCE_DB`.

The entity is passed in the constructor of the runtime application object of class `Mapbender\CoreBundle\Component\Application` which has all the logic while the entity holds all the data.

Access checks are not done in the mapbender service, entity or application object but in the controller's `getApplication` function - keep in mind if you develop your own controller.



Element Rendering Workflow

Upon application rendering, every element is asked to render itself. The workflow is a little tangled but, basically the application asks it's template to render itself which includes the template iterating over each element and calling the elements render function.

The render function usually uses twig to render a HTML fragment to stand for the element and this fragment **must** include an id attribute set to the id dynamically given to the element by the Mapbender application! As these ids are generated dynamically you must not use them for styling in your element's CSS.

Also, the application iterates over each element and calls the `getConfiguration` method to collect all elements' configuration arrays which are made available client-side in the `Mapbender.configuration.elements` variable - as well as given to each element's widget construct method.

To sum up, the elements configurations are available:

- In the element's render function which can pass the configuration to the twig file.
- In the element's widget methods as `"this->options"`.
- In the `Mapbender.configuration.elements` JavaScript variable (for all elements)

Element Callback Workflow

The standard element callback `/application/{slug}/element/{id}/{action}` is routed to the `MapbenderCoreBundle\Controller\ApplicationController::elementAction($slug, $id, $action)` function.

This function does the same as the `applicationAction` to get the application object, therefore application access is checked and no access to the element callback is possible if the user is not allowed to use the application.

The element is retrieved and then access to the element is checked based on the roles of the current user.

Finally, the element's callback function `MapbenderCoreBundle\Component\Element::httpAction($action)` is called. This function must return an `Symfony\Component\HttpFoundation\Response` object or throw an `HttpException` (not found, access denied, etc.)

It is left to the element to implement its logic in its `httpAction` method. Best practice is to do an `switch` statement over the `$action` parameter. Default should be an `HttpException`. Using the standard Symfony methods you can access the request as an Symfony service with `$this->get('request')` in the element class.

Element widgets can construct the callback URL for use in their Ajax calls by using their element id and the path provided in the Mapbender variable:

```
(function($) {
    $.widget('mapbender.mbDemo', {
        elementUrl: null,

        _create: function() {
            this.elementUrl = Mapbender.configuration.elementPath + this.element.attr('id') + '/';
        },

        _doMagic: function() {
            $.ajax({
                url: this.elementUrl + 'myAction',
                data: { foo: 'bar' }
            });
        }
    });
})(jQuery);
```

Manager Workflow

Manager modules are plain Symfony controller classes. To ensure their routes are prefixed with `/manager` - or whatever prefix is configured for the manager bundle - you should not use the `Sensio\FrameworkExtraBundle\Configuration\Route` annotation, but the special `FOM\ManagerBundle\Configuration\Route` annotation. Actually, that annotation class is not so special after all, but it enables the `ManagerBundle` to enforce the prefix.

It is best practice to use the `Route` annotation under a different name to make it visually easier to debug the controller for the use of the right annotation:

```
<?php

namespace Acme\DemoBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use FOM\ManagerBundle\Configuration\Route as ManagerRoute;

class DemoManagerController extends Controller
{
    /**
     * @ManagerRoute("/demo")
     */
    public function demoAction()
    {
        /* ... */
    }
}
```


8.2 How to write Mapbender Documentation?

This documentation is built and deployed from the mapbender-documentation repository on Github.

Please take a look at the documentation files in this repository:

<https://github.com/mapbender/mapbender-documentation/>

We generate the website code from the rst-files in the Mapbender documentation repository using Sphinx.

To build the website locally, you need to install Sphinx first. Find more information in the README.md file:

<https://github.com/mapbender/mapbender-documentation/blob/master/README.md>

VERSION HISTORY

German version of this document.

Note: This version history is no longer maintained. Please head over to GitHub to see the latest Mapbender [Changelog](#) and [Version Milestones](#).

9.1 Version 3.2.9

Release date: 28.01.2022

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v329>

9.2 Version 3.2.8

Release date: 02.11.2021

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v328>

9.3 Version 3.2.7

Release date: 07.09.2021

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v327>

9.4 Version 3.2.6

Release date: 09.08.2021

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v326>

9.5 Version 3.2.5

Release date: 08.06.2021

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v325>

9.6 Version 3.2.4

Release date: 04.03.2021

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v324>

9.7 Version 3.2.3

Release date: 21.12.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v323>

9.8 Version 3.2.2

Release date: 02.11.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v322>

9.9 Version 3.2.1

Release date: 06.08.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v321>

9.10 Version 3.2.0

Release date: 29.07.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v320>

9.11 Version 3.0.8.6

Release date: 15.09.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3086>

9.12 Version 3.0.8.5

Release date: 05.02.2020

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3085>

9.13 Version 3.0.8.4

Release date: 04.09.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3084>

9.14 Version 3.0.8.3

Release date: 05.07.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3083>

9.15 Version 3.0.8.2.1

Release date: 05.07.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v30821>

9.16 Version 3.0.8.2

Release date: 03.07.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3082>

9.17 Version 3.0.8.1

Release date: 14.05.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3081>

9.18 Version 3.0.8

Release date: 12.04.2019

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v308>

Upgrading

<https://github.com/mapbender/mapbender/blob/master/UPGRADING.md#308>

9.19 Version 3.0.7.7

Release date: 07.11.2018

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3077>

9.20 Version 3.0.7.6

Release date: 18.10.2018

Improvements and bugfixes:

- <https://github.com/mapbender/mapbender-starter/blob/master/CHANGELOG.md#v3076>

9.21 Version 3.0.7.5

Release date: 26.09.2018

Improvements and bugfixes:

- Find the description of the fixes in the repository links
- Update [Mapbender-Starter v3.0.7.5](#)
- Update Mapbender to [v3.0.7.5](#)
- Update Owsproxy to [v3.0.6.4](#), includes Owsproxy dependencies
- Update mapbender/vis-ui.js to [0.0.72](#)
- Update mapbender/data-source to [0.1.8](#)
- Update mapbender/digitizer to [1.1.66](#)
- Update bundled Composer to [1.6.5](#)
- Misc ComposerBootstrap cleanups

9.22 Version 3.0.7.4

Release date: 29.08.2018

Improvements and bugfixes:

- [Security] In some development environment a XSS error can occur with the assets. This error was only observed on some environments with specific debug PHP settings (error_reporting z.B. E_ALL).
- Revert the column type of the keyword-column to "varchar" to avoid incompatibilities with Oracle. Very long keywords are truncated back to 255 characters (#1000).
- Some JavaScript fixes that leads to problems with defunct Internet Explorer 11 (#990).
- Empty layer-names are not requested again in FeatureInfo (PR #1010).
- Doctrine optimizations to store layer-order settings in PostgreSQL.
- Regression fix in WMSLoader and image format / info format settings.
- Fix on Delete Cascade SQL statements with PostgreSQL, when deleting a WMS source.
- Fix in translations when only a placeholder was set. These fall back now to the fallback translation (per default: english).
- Update OSGeo Logo (PR #861)

Notes for the update:

Please call again the **app/console doctrine:schema:update** command to set back the keyword-table and column to varchar.

```
app/console doctrine:schema:update
```

If this statement fails, for example with the PostgreSQL error SQLSTATE[22001]: String data, right truncated: and 7 ERROR: Value too long for type character varying(255), you probably have a keyword-entry in the table mb_core_keyword, that exceeds the length of 255 characters. You can find out this entry with the following SQL-statement:

```
SELECT x, id, length(x) FROM (
  select value, id from mb_core_keyword
) AS t(x) order by length desc;
```

9.23 Version 3.0.7.3

Release date: 13.07.2018

General:

- Change of the Mapbender logo and the name: Mapbender3 is due to simplicity reasons renamed into Mapbender and we have change all texts in this documentation and the logos. Our URL were changed to <http://mapbender.org> some time ago.
- Mapbender requires at least PHP > 5.6 for running. We recommend PHP 7.

New functions:

- QGIS Server layer ordering, documented at layerset
- New element: *Coordinates Utility*
- Mouse-Over in SearchRouter
- GPS Button in POI
- Dynamic Loading of legend-images in the legend-element (PR #605, PR #606)
- Display of a cookie banner in applications. See *Cookie Banner*.

Changes:

- Die default applications are moved to the directory *(application)/app/config/applications*, each in its own file. This includes
 - the Mapbender Demo Map application
 - the Mapbender Demo Map basic application
 - the Mapbender mobile application

Additional YAML-applications can be placed there.

Improvements:

WMS Loader and WMS Services:

- Improvements in WMS Loader service compatibility, which logic now matches the backend
- Fix in the GetLegendGraphic request for a secured service via the tunnel
- Fixes and improvements for the URL-signing (#590)
- Many improvements in the WMS Backend
- Fix in instance-tunnel while requesting secured services
- Fix on accessing WMS services with undefined contact information.
- Various fixes to displaying and handling min / max scale definition from sublayers vs root layers
- Fix saving layer order on PostgreSQL
- Services loaded with WMS Loader and their metadata display. We cannot read the properties, but we don't throw an error anymore.

Design and CSS:

- Changed Opacity for zoombar and toolbar to get a unique button color
- Fix for creating an application and adding the screenshot

Print:

- Fix printing PNG8 maps if the image format was specified as "image/png; mode=8bit".
- Fix printing special font-sizes (especially at Windows with PHP 7.1)
- Fix in printing if PHP notices were switched on in php.ini and the yStartPosition was missing (#555)

FOM:

- Improvements in FOM: Wrong Type Definition in ACL Provider Constructor #641
- Improvements in FOM at SSPI

Translations:

- Improvements in the translations. Thanks to the Code-Sprint of the FOSS4G!
- Changes in translations from XLIF to YAML in the modules FOM and OWSProxy

Miscellaneous:

- Per default the maximum feature count for GetFeatureInfo is now 1000.
- Fix in the scale-selector, which did not want to refresh itself
- Fix in the call of Mapbender with the POI parameter (#642)
- Fix in the legend-element for oversized legend-images (#640)
- Fix on adding new elements in the backend
- Fix foreign key violation error in PostgreSQL when deleting data source (PR#840).
- Add cookieconsent code for Mapbender
- Change default prefix for printouts to mapbender

Code-Improvements:

- Update to Symfony 2.8 (please see the PHP requirements)
- Introduction of the Doctrine Migrations framework
- Fix possible URL signing spoof with input URLs missing query parameters.
- Doctrine Param Coverter definitions (PR #645)
- WMSLayerSource: getAuthority (PR #542)
- DimensionsHandler (#610). This will be published in the forthcoming versions
- Adding elements in backend can fail with "Warning: usort(): Array was modified ..." (#586)
- Element Template and AdminType Fixes (#743)
- Serialization of MetadataURL (#747)
- UnitTest and Pre-Conditions (#760)
- USort und array_multisort with a PHP-bug (#586)
- Fix strict SCSS warnings when compiling with ruby-sass
- Fix unbounded growth in "authority" on repeated export / reimport / cloning of applications

- Bypass (potentially very long) WmsLoader DTD / XSD validation of GetCapabilities document
- PHP 5.6 compatibility with Migrations

Documentation:

- New design of the documentation. We have changed the theme to the Sphinx RTD theme. The documentation is now easily readable on mobile devices. You can also print out specific pages.
- Restructuring of the documentation. The specific *Elements* are sectioned into:
 - *Base elements*
 - *Search*
 - *Export / Print*
 - *Editing*
 - *WMC*
 - *Backend*
 - *Security (FOMUserBundle)*
 - *Other*
- Improved documentation for the elements:
 - *Base source switcher*
 - *Button*
 - *Coordinates Display*
 - *HTML*
 - *Legend*
 - *Map*
 - *Navigation Toolbar*
 - *Overview*
 - *Search Router*
 - *SRS Selector*
- Improvements for the *PrintClient* and the new dynamic features on print-templates.
- Included the MS4W package for installation under *Installation on Windows*. Please take a look. And thank Jeff McKenna.

Notes for the update:

Please call the command **app/console doctrine:schema:update** for the Update to this version. The QGIS layer ordering needs a change in the Mapbender database. Also the 255 characters for WMS services require a change of the database.

`app/console doctrine:schema:update`

9.24 Version 3.0.7.2, 3.0.7.1 und 3.0.7.0

Due to tagging-errors of the code in Github these versions were never officially released. It's not a good idea to re-tag the code, so we continued with Version 3.0.7.3.

9.25 Version 3.0.6.3

Release date: 27.07.2017

Bugfixes:

- Regression: Fixed coordinate order at requests to a WMS 1.3.0. Coordinate reference systems with reversed axis-orientation are supported by map, print and export. (#529)
- Regression: Fixed ScaleHint for WMS services. Some WMS services that defined a Scale in their Capabilities couldn't be put into an application. (#584)

9.26 Version 3.0.6.2

Release date: 20.07.2017

Bugfixes:

- Search Router: Error with OpenLayers fixed (#543)
- Search Router: Auto Close after Click in mobile application (#548)
- Coordinate order (axis order) on requests to WMS 1.3.0 fixed (#529)
- Print: Rendering of points and labels on high-resolution Print (#573, #574, #492)
- Saving of WMC in WMC editor dialog (#577)
- ScaleHint for sub-layers of 1:1 fixed (#565)
- Widen the Title-Columns on LayerSet-Instances (#559)
- Command to update the image-path in existing map-elements (#530)
- Translation of Print-button in FeatureInfo dialog (#552)
- Change of default-value of "immediate" in measure-tools (#538)
- SRS: Update of definitions (#550, #562) and update of YAML standard applications (#561)
- Update documentation for handling of translations.

Additional update steps:

(1) Update of EPSG-Codes

Execute again the command `app/console doctrine:fixtures:load --fixtures=mapbender/src/Mapbender/CoreBundle/DataFixtures/ORM/Epsg/ --append`. Two new coordinate-systems are added to the Mapbender database-table `mb_core_srs`: `EPSG:4839|ETRS89 / LCC Germany (N-E)` and `EPSG:5243|ETRS89 / LCC Germany (E-N)`.

(2) Update of parameters in Map-Control

Execute the command `app/console mapbender:upgrade:database`, to update the `OL-imagePath` Parameter from `bundles/mapbendercore/mapquery/lib/openlayers/img` to `components/mapquery/lib/openlayers/img`. This is necessary if you use the `POI-Elements` or call `Mapbender` with the `poi-parameter` and see no bubble-icon for the `poi`. Example: [https://demo.mapbender.org/application/mapbender_user?poi\[{}point{}\]=366164%2C5623183&poi\[{}scale{}\]=25000&poi\[{}label{}\]=Please+take+a+look+at+this+POI%3A](https://demo.mapbender.org/application/mapbender_user?poi[{}point{}]=366164%2C5623183&poi[{}scale{}]=25000&poi[{}label{}]=Please+take+a+look+at+this+POI%3A)

9.27 Version 3.0.6.1

Release date: 24.05.2017

Bugfixes:

- Print map showed wrong scale in map-display.
- Specific WMS could not be printed due to HTTP response `image/png; charset=iso....`
- Backend: FOM dialogues with many entries made checkboxes unusable.
- `config.php` back in `mapbender-starter`.
- Update `bin/composer` build command for building `Mapbender`.
- Style-Fix for `administrations-dialog` of `Basesource-switcher`.
- Add `WmcEditor` Default Parameters for width and height.
- Update landing-page of this documentation.
- Some minor styling changes in backend.
- Some cleanup.

9.28 Version 3.0.6.0

Release date: 05.05.2017

Architecture:

- System Requirement PHP: 5.5.4 or higher.
- Support of PHP 7.
- `Mapbender`, `FOM` and `OWSPProxy` excluded into Modules. They are now bind in `composer.json`
- Documentation is part of the composer.
- Adjustments of the `ElementGenerator`
- Determining of user-roles
- Composer entries with `https`
- Adjustments of `Controllers` and `Bundles`.
- `Doctrine generate` commands are marked deprecated
- `Doctrine assets:dump` command is marked deprecated
- Update of the `JOII` library
- Introduction of symlinks to the different binaries in the `bin` directory of `mapbender-starter`

- Composer shipped in application/bin directory
- Check in the Composer-installation, if the SASS Compiler Binaries are executable. If not, they are set executable.
- New Composer commands for documentation: Generate API documentation only: bin/composer gen-api-docs
- New Composer commands for documentation: Generate User-documentation only: bin/composer gen-user-docs
- Use of own forks of open-sans, joi, compass-mixins and codemirror libraries.

Bugfixes und Features:

- Measuring shows the coordinates directly, by moving the mouse the calculated lengths of the segment are displayed live.
- New measuring results are shown on Top. The current result is visible at first place and you don't need to scroll.
- The login-dialog (registration, forgotten password) is optimized for mobile devices to achieve a better workflow to secured mobile applications.
- New added layerset instances are now per default not marked as base-source.
- The [Copyright element popup](#) can be defined with a height and a width.
- Deleting a layerset led in some cases to a corrupt map element and a wrong layertree.
- Adjustments and Simplification of the general style of the FullScreenTemplate
- Introduction of the check of the CSS statements in a application
- Fix in the delay when switching layers.
- Fix in GetMap request if the layer order was changed manually in the TOC.
- Fix for WMS support 1.3.0
- Fix for secured WMS services on GetMap, GetFeatureInfo, Print, Export and Legend.
- Fix for secured WMS services where the username or password included a hash-character.
- Fixes for the WMS parameter Exception Format for the GetMap and GetFeatureInfo Request (Github-Issue 400)
- Fixes of Layer-Styles for GetMap und GetFeatureInfo request
- Default Tile Size for the Map set to 512 (was 256)
- WMS Keyword limit (was: 255 characters) is changed. The column-type is now "text". The command `app/console doctrine:schema:update` is necessary to update the Mapbender database of a previous version.
- Fix when importing YAML based applications and creating duplicate WMS data-sources.
- Fix on minimal and maximal scale hint on WMS services.
- Print: Color can be set for variable texts.
- Print: Printout of the legend, if the service is built in with the proxy.
- Print: Services registered with PNG8 could not be printed or exported in some cases. The types `image/png8` and `image/png mode=24bit` is not supported.
- Print: In some cases, the legend wasn't printed if OWSProxy was activated

- BaseSourceSwitcher: Duplicate request on WMS which was not visible and where the BaseSourceSwitcher was used as a menu.
- Unnecessary requests on specific WMS configurations with scale.
- Saving of YAML applications in application/app/config/applications: mapbender_mobile.yml, mapbender_user_basic.yml, mapbender_user.yml and adjustments in their referenced WMS Version
- Fixes in administration interface of the YAML editing after saving (Github-Issue 350)
- Fixes in POI-Coordinate: Transformation and SRS and the trailing digits after the comma.
- Fix of a XSS error in POI dialog
- Fix in POI dialog, if useMailto is set to false
- GPS: Error messages if no GPS signal and the dependency to Chrome-browser and https.
- GPS: Pan the map only on first position.
- User-Interface: Scrolling of a drop-down list in backend, for example the icons for the buttons, did also scroll the background.
- “Only valid” Checkbox on [loading a WMS](#) is now per default not activated anymore.
- Reformatted messages if the schemes of a WMS are not accessible when adding a WMS.
- The [SearchRouter](#) shows, if placed in the sidebar, the Search and Reset buttons.
- Internet Explorer Compatibility: Adjustments in the [Zoombar](#)..
- Internet Explorer Compatibility: Adjustments in the [OverviewMap](#).
- MS Edge Compatibility: Trying to fix the Import Dialog (<https://connect.microsoft.com/IE/feedback/details/1574105/microsoft-edge-file-upload-bug-build-10240-rtm>)
- Improvement of the performance on *some* Windows installations through WinCachePHP and PHP Opcache (for details see [Installation under Windows](#))
- Change in System-Requirement for Windows: For PHP, the “Non-Thread-Safe” version is needed!
- Copying of applications through users who are not root (ACL Application: owner, Users: owner, ACLs: owner, Element: owner, Groups: owner, Service Source: owner, specific applications: owner)

FOM and Security:

- [Show the users](#) who have a access on an element in an application.
- Rework of the Secure Elements dialog.
- User with the role View for services are allowed to view the Metadata and to load the services into an application.

miscellaneous

- Design and presentation of the FeatureInfo dialog if shown as Accordion. Also if shown as “not from source”.
- Drag of Popups in an application.
- WMC Editor: Adjustments in size. XSS fix.
- Fix of translations
- YAML based applications can adjust the sidebar: align (left/right), closed (true/false), width (px/em/%)
- Backend: Target field: Empty option of a Drop-Down field.

- Adjustments of WMS Scale, ScaleHint and Min/Max values when a Layerset-Instance is opened.
- Display of WMS Title in the metadata of the TOC, when the WMS was updated.
- Display of the application logo in the Configuration.
- Display Issues of Simple Search and Search Router.
- Fixes for error messages on a wrong configured Layerset-Instance.
- Print: Introduction of setasign/pdf instead of toooni/fpdf
- Print: Fix of error messages on a missing print-template
- Measuring of lines and polygons in WGS84 (EPSG:4326)
- Adjustments of display of the element ACL
- WMS Update: remove user/password from web-browser autocomplete
- Display of number of digits in coordinates-display.
- Adjustments and extending EPSG import
- fix default visibility for a layer and the scale
- remove of DataSource Monitor icon (comes in version "next")
- Administration: Movement on the tabs with the tab-key
- Improvement of the display of the configuration interface
- Display of Source-ID in applications
- Improvements of Caching Mechanisms
- Improvements of Export and Copy mechanism.
- Improvements for the creation of new elements.
- Remove the provide ext-ldap statement in Composer. The components are released from the core. We will build in the LDAP module in version 3.0.7.
- Restructuring of DataManager and DataSource since [version 1.0.2 of data-manager](#).

Mobile template

- General improvements of the mobile template.
- Fix handle mobile template button click if target isn't defined
- Set mobile icon label font weight to normal
- Fix and improve mobile template button handling
- Fix register mobile application event handler "on moveend"

Digitizer

- Update [Digitizer](#) to version 1.1.
- Printing of Multipolygons.
- Objects don't appear in the printout if they are not displayed in the Digitizer.
- MinScale restriction added
- Objects with a line-width of 0 are now not shown anymore in the printout.
- Adjustments of the Close Button: "allowCancelButton" and "allowDeleteByCancelNewGeometry".

Form Generator:

- Adjustments: Add HTMLElement handling of service and DataStore configuration.

Dokumentation

- Introduction of the [FAQ](#).
- Introduction of Contributing Guide for [Mapbender-Starter](#) and [OWSProxy](#). Mapbender itself and FOM will follow. This is the main documentation for developers and contributors of Mapbender.
- The developer documentation will be maintained there and be transferred step-by-step from this user-documentation. So in the future this documentation here will be more for users and the developers have their documentation directly in the source code of the different modules.
- Better [Layertree](#) documentation

config.yml Anpassungen

DBAL-Parameter:

- `default_connection`: If more database entries are defined, this parameter
- `persistent`: Persistent connections to the database for performance reasons (Oracle)

```
doctrine:
  dbal:
    default_connection: default
    connections:
      default:
        ...
        persistent: true
```

mapbender-starter/application/app/config/applications/

Directory where YAML-based application definition are stored. As an example the well-known applications Mapbender-User, Mapbender-User-Basic and Mapbender-Mobile are placed here.

app/console doctrine:schema:update

```
app/console doctrine:schema:update --dump-sql
ALTER TABLE mb_core_keyword ALTER value TYPE TEXT;
ALTER TABLE mb_core_keyword ALTER value DROP DEFAULT;
```

9.29 Version 3.0.5.3

Release date: 04.02.2016

Bugfixes:

Notable Modifications:

- Performance: The CSS, JavaScript and Translation files are now held in the Symfony Cache for the [production mode](#). This can lead to better performance on slower machines. These cache is not used by the [development-mode](#).
- The package [eslider/sassc-binaries](#) offers now a sassc Compiler for 32-bit Linux systems. This led to a wrong display on 32-bit Linux-Systems (http://lists.osgeo.org/pipermail/mapbender_users/2015-December/004768.html)

- **Redlining:** The contents of the Redlining element is visible and Redlining can now be used as a Dialog or an Element in the Sidepane. See also the [documentation of the Redlining Element](#). The scroll bar for the Geometry-Types in the configuration dialog is now displayed correctly.

Users and security:

- Users can be switched active or inactive by an Administrator, who has at least the ACL-user-right “operator”. This can be used for users, who have self-registered but not yet activated their account. See the [documentation of user-management](#) for details.
- The text, translations and styles for the Self-Register process and the Password Reset are improved. Also the [Documentation](#) is adjusted.

Print and export image:

- The [Print module](#) can now also be used in the Sidepane.
- **Print legend:** The size of the legend in the print-out was scaled down to improve the quality.
- **Print-templates:** The default print-templates have changed. The padding of the dynamic texts to their border and their justification were improved.
- **Print:** The Print configuration messed up mandatory (required: true) and optional fields (required: false), if they were used in combination. Optional fields were partly shown as mandatory (Github #380).
- In some cases Mapbender printed the legend of all WMS-layers, even if the layer was not set active (seen in Mapbender_Users WMS).
- **Export Image:** Transparency of tiled and non-tiled services is supported in Export Image.

Copy and import:

- Copying an application under SQLite and MySQL: There was an error that applications could not be copied if the database was SQLite or MySQL.
- Errors at the import of application as JSON on MySQL (elements lose their target) was fixed.

Individual Elements:

- **WMC** and thematic layertree: If a WMC is loaded and Keep Sources is set to “no”, the thematic layers are now also removed from the layertree.
- **WMS-URL parameter** and legend: If a service was loaded with the wms_url parameter, the complete legend was shown. This behaviour is fixed.
 - *Note:* WMS services exist, which define a legend in the root-layer element. According to the WMS-specification, this legend will be inherited by sub-layers who themselves haven’t defined a legend (for example if they only contain the annotations). The effect is similar in MB3 but the cause is different, so that in these cases a change in the WMS capabilities is needed (define a static legend image for these layers).
- **Thematic Layer:** Fix in switching layers on and off which are in their own Layerset but not displayed as a thematic layer.
- **Coordinate display:** The coordinate-element display doesn’t show “null” as prefix or separator anymore, although the field was defined as empty. The element has got a fixed width so that the layout in the footer region is more stable. The value can be changed (see the chapter [CSS-customizing of the element](#)).
- **SearchRouter:** The content of the result uses the whole space of the dialog and fits itself to changes of the size. In the sidebar the whole height is used. The search router can be configured [with a width and a height](#).

- **ScaleSelector:** The width of the element can be [customized with a CSS-Statement](#) and is no more set to 155 pixel.
- If all layer in a **layerset-instance** are set to visible=off they were not visible in the layertree and the legend. This is fixed.
- Improvements in the styling of the **POI dialog** if “usemailto” is set to false.
- **Layertree:** Titles are now shown per default with a length of 40. The default value has been changed. You can set the [parameter Titlemaxlength in the configuration dialog](#).
- **GPS:** Improvements in the GPS handling.

General changes:

- Changing the Base Data, the Layout, the Layerset, the CSS and the Security of an application does not change the tab anymore and doesn't jump back to the base data tab.
- General improvements of the [Digitizer Code](#) version 1.0. Version 1.1 is compatible with Mapbender 3.0.5.3.
- Github files: Small clean up actions in the Github repository to improve the automatic build-processes.
- Transparency of services: Services with a transparency refreshed with a poor effect, caused by the “transitionEffect” in OpenLayers. The effect was removed.
- Group filter: The security configuration dialog got improvements at the selection of Groups, if the Groups had the same name but a different suffix.
- TileSize Parameter in the map configuration was not set in some cases.
- Display of symbols in Internet Explorer 11 and MS Edge 25 (also an error in MS Edge 20).
- mapbender.yml: At the initial import of the mapbender.yml the values of GetFeatureInfo are now set to text/html. The mapbender.yml can now customized with Redlining.

Change of the Mapbender domains:

- We have switched the URL www.mapbender.org to the Mapbender3 page. In future, the Mapbender3 page is also available via www.mapbender.org and www.mapbender3.org. Mapbender2 is now available at www.mapbender2.org
 - <http://www.mapbender.org>: Mapbender3,
 - <http://www.mapbender3.org>: Mapbender3,
 - <http://www.mapbender2.org>: Mapbender2.

Known Issues:

- The Sketch Tool doesn't work correctly and will be built into the [Redlining Tool](#).
- Share map doesn't work for Facebook, Twitter und Google+.

9.30 Version 3.0.5.2

Release Datum: 27.10.2015

Bugfixes:

- Copy applications: User-Rights and groups are copied. The user who copied the application becomes owner of the copied application.
- FOM: Changes in behaviour of wrong logins and user locking. It is only shown that the login failed, independent if the user exists or not.
- Fixed error message when creating a user with a too short password.
- Print: Fix of replace pattern.
- Print: Fix if a wrong configured WMS has special characters (%26) in the legend URL.
- Image export in Firefox.
- WMC Loader: Loading WMC and Behaviour of BaseSources.
- BaseSourceSwitcher: Tiles of a not visible service are not pre-fetched.
- BaseSourceSwitcher: If a group is defined, only one theme is switched on.
- SearchRouter: Fix of quotes for table-names.
- Copy applications: Fix of the search in the copied application.
- Simple Search: Catch the return key.
- FeatureInfo: Add WMS functionality and WMS Loader.
- Icon Polygon is visible in the toolbar of applications.
- Icons, which are not based on FontAwesome also work in the mobile application.
- Administration of the map element: The view of the configuration dialog in the backend starts on top.
- Administration data source: No form data auto-complete from the browser for username and password.
- Mobile application: Design in Firefox for Android.
- Update 3.0.4.x: FeatureInfo autoopen=true is kept.
- Doku: FOM [UserBundle translation](#) and [additional information for failed user logins](#).
- Doku: URL parameter scale in [map element](#).
- Doku: [WMC Loader](#) and [KeepSources](#).

Changes in config.yml:

- The following changes are optional parameters for the behaviour of the login (see also [the chapter in the FOM bundle for details](#)):

```
fom_user:

  # Allow to create user log table on the fly if the table doesn't exists.
  # Default: true
  auto_create_log_table: true

  # Time between to check login tries
```

(continues on next page)

(continued from previous page)

```
login_check_log_time: "-5 minutes"

# Login attempts before delay starts
login_attempts_before_delay: 3

# Login delay after all attempts are failed
login_delay_after_fail: 2 # Seconds
```

9.31 Version 3.0.5.1

Release Datum: 26.08.2015

New functions: in the [Map element](#) and in the [Print client](#):

- Map: OpenLayers TileSize: You can set the tile-size for the map. Default: 256x256.
- Map: Delay before Tiles: For WMS-T, for example with temporal parameters (in future)
- Print: Show coordinates in PDF print
- Print: get print scale depending on map-scale
- Print: print legend_default_behaviour
- Print: add print templates with the + symbol
- Print: user-defined logo and text

Bugfixes:

- Layertree: loading symbol and exclamation mark symbol.
- Layertree: zoom Symbol not for layers without a BBOX information
- WMS Reload: FeatureInfo
- WMS Reload: some WMS couldn't be reloaded.
- Export/Import of application and miscellaneous bugfixes
- WMC-Editor and WMC-Load fixes.
- WMC from a Mapbender 3.0.4.1 application
- Tile buffer and BBOX buffer fixes
- FeatureInfo: Fixes in design and when shown as an Accordion Panel
- FeatureInfo: Print
- Wrong JQuery-UI link in layerset instance
- Save Layerset and Save Layout leaves you on the page
- Classic Template: SCSS corrections
- Mobile Template: Bootstrap message hides close button
- Mobile Template: close SearchRouter window
- Mobile Template: Mozilla Firefox Fixes on layout
- Backend: Layerset Filter and +-Buttons doesn't hide everything anymore

- composer.json upgrade version of Digitizer to 1.0.*
- Documentation of the JS-UI Generator (Form-Generator): <https://github.com/eSlider/vis-ui.js>
- Restructured [Installations-Dokumentation](#) and some changes (php-pear, assets-Verzeichnis, init:acl, openssl).
- Better documentation of the [Mapbender3 Templates](#)
- Better documentation of the [Quickstart](#)

Known Issues:

- After copying an application from Mapbender 3.0.4.x you have to set the layerset in the map/overview element. Please save the map and overview element beforehand.
- Regional Template removed

9.32 Version 3.0.5.0

Release Date: 01.07.2015

For details have a look at: <https://github.com/mapbender/mapbender-starter/blob/develop/CHANGELOG.md>

- **WMS reload:** WMS sources can now be reloaded if the structure has changed.
- **Digitizer:** The digitizer allows the editing of geometries and their attributes. Right now it needs access to the database where the editable tables are. The definition of the digitizer is done in YAML syntax. To provide an usable interface for the attributes, you can declare the form in your configuration file. The form supports different input fields (textboxes, checkboxes, date-pickers, and so on..), validation, tabs and it uses Bootstrap.
- **Print with legend:** The print element supports the print-out of the legend on a separate page. This can be set with a checkbox.
- **Configurable layertree:** The layertree supports the usage of more than one layerset. You have to adjust the map element to define which layersets should be shown and the layertree element itself. The usage is documented [on the Layertree page](#).
- **Improved FeatureInfo dialog:** You can set a) the width and height of the FeatureInfo dialog, b) if the dialog should show the original format of the WMS and c) if it should only open if a valid entry is found (otherwise a messagebox is displayed). See the documentation of the [FeatureInfo Dialog](#).
- **Mobile template:** A new modern mobile template is provided.
- **SASS Compiler:** Architectural changes are made at the SASS compiler which leads to a more performant interface.
- **Vendor Specific Parameters:** A WMS layer instance supports the definition of Vendor Specific Parameters that are added to the WMS request. You can define hard coded values or the user or group information of the logged-in user. See the documentation of [Vendor Specific Parameters](#) for details.
- **Expanded functionality of HTML elements with a form-builder:** This approach is used in the Digitizer to provide the forms for attribute editing.
- **New button collection:** The new buttons are based on a new font, the old buttons are available under the "FontAwesome" name.
- **Starting mapbender with URL parameters:** Mapbender3 can be started with URL parameters. See the documentation of [URL parameters](#).

- New translations for Portuguese and Russian.
- Symfony updated to 2.3.30.

Changes in config.yml:

- Changes in a dbal connection:
 - **logging: false:** This options sets, that *all* SQL statements are not logged. Further information can be found here: <http://www.loremipsum.at/blog/doctrine-2-sql-profiler-in-debugleiste/>
 - **profiling: false:** This option handles the profiling of SQL statements. This option can be switched off in production environments.

If possible the options should be set this way, so that they are only active in debug mode:

logging:	"%kernel.debug%"
profiling:	"%kernel.debug%"

Known Issues

- After copying an application from Mapbender 3.0.4.x you have to set the layerset in the map/overview element.

9.33 Version 3.0.4.1

Release Datum: 23-01-2015

For details have a look at: <https://github.com/mapbender/mapbender-starter/blob/develop/CHANGELOG.md>

- option 'removelayer' added into layertree menu
- parameter 'layerRemove' removed from layertree configuration
- container accordion structure changed
- import / export from applications added (without acls)
- display layer metadata
- Frontend: Sidepane Accordeon Legend is displayed without horizontal Scrollbar
- Backend: WMS Instanz configuration - contextmenu for layers shows wrong ID (only instance ID)
- Frontend: Legend - displays WMS Information although the checkbox Show
- Frontend: Layertree - contextmenu zoomlayer does not use the layer extent
- Backend: Add Source with user/password - informations is added to field originUrl not to fields user and password
- app/console mapbender:generate:element fixed errors
- bug visiblelayers fixed
- WMS with authentication saves in table mb_wms_wmssource username and password
- no metadata for applications coming from mapbender.yml definition (no entry in context menu)
- copy an application via button on application fixed
- print template resize northarrow, overview added
- improved screenshot for application handling

- <https://github.com/mapbender/mapbender/milestones/3.0.4.1>

9.34 Version 3.0.4.0

release date: 12-09-2014

For details have a look at <https://github.com/mapbender/mapbender-starter/blob/develop/CHANGELOG.md>

- Switched to MIT license
- Symfony Update 2.3 LTS
- OpenLayers 2.13 with additional patches
- Switch Services (BaseSourceSwitcher) with menu
- added generic HTML element
- added custom CSS editor for applications
- added accordion container for SidePane
- added screenshot management to application editing
- import/export of applications/sources
- spanish translation

9.35 Version 3.0.3

release date: 17-03-2014

For details have a look at: <https://github.com/mapbender/mapbender/issues?milestone=8>

- Enhancements for Search-Router für SQL-Suchen (Selectboxes, Distinct)
- WMC Editor and Loader
- WMSLoader Enhancement add WMS via link
- i18n - Internationalisation (english / german)
- Sketch to draw temporary objects
- POI - Meetingpoint
- Imageexport to generate png or jpg
- Change WMS Collection via button (BaselayerSwitcher)
- Print with overview
- Sidepane with different elements (chnage via button)
- Layertree context menue to change opacity and to zoom to layer
- Open application with parameters (e.g. position)
- ACL for elements
- Added function for validate WMS GetCapabilities documents

9.36 Version 3.0.2

release date: 27-11-2013

For details have a look at <https://github.com/mapbender/mapbender/issues?milestone=6>

- SearchRouter
- WMC Editor and Loader
- WMSLoader enhancement to load a WMS from a link

9.37 Version 3.0.1

release date: 06-09-2013

For details have a look at <https://github.com/mapbender/mapbender/issues?milestone=5>

- Kopieren einer Anwendung mit Diensten
- Popup - draggable
- PrintClient Erweiterung Druck EPSG 4326, neue Drucklayouts, Druck A4-A0
- Catch login failures to avoid brute force login attempts
- Bug fixes

9.38 Version 3.0.0.2

Bugfix-Release Date: 19-07-2013

For details have a look at: <https://github.com/mapbender/mapbender/issues?milestone=4>

9.39 Version 3.0.0.1

Bugfix-Release Date: 07-06-2013

For details have a look at: <https://github.com/mapbender/mapbender/issues?milestone=3>

9.40 Version 3.0.0.0

release date: 29-05-2013

For details have a look at <https://github.com/mapbender/mapbender/issues?milestone=1>

- Administration Backend for Service, Application, User/Group and security administration
- Backend-/Frontend Design
- Security
- User/Group Administration
- WMS Administration

- Map
- Layertree
- Legend
- Overview Map
- Navigation Toolbar (Zoombar)
- Feature Info
- Coordinates Display
- Copyright
- Line/Area Ruler
- Scale Selector
- ScaleBar
- Spatial Reference System Selector
- GPS-Position
- Print
- Add WMS to application
- Documentation at <http://doc.mapbender.org>