Lizmap Documentation

Release 3.5

3liz

Mar 08, 2022

CONTENTS

1 Introduction **3**
- 1.1 Lizmap architecture . . . 3
- 1.2 QGIS and QGIS Server . . . 4
- 1.3 Lizmap features . . . 5

2 User guide **11**
- 2.1 The projects page . . . 11
- 2.2 Authentication . . . 12
- 2.3 A simple map . . . 14
- 2.4 Selecting a basemap . . . 20
- 2.5 Measurement . . . 21
- 2.6 Locate by layer . . . 25
- 2.7 Printing . . . 27
- 2.8 Attribute layers . . . 32
- 2.9 Editing spatial data . . . 41

3 Publisher guide **49**
- 3.1 Quick start . . . 49
- 3.2 Lizmap QGIS plugin . . . 55
- 3.3 Configuration . . . 96
- 3.4 Customization . . . 134

4 Administrator guide **143**
- 4.1 Overview of the administration interface . . . 143
- 4.2 Manage groups and users in Lizmap Web Client . . . 144
- 4.3 Lizmap Web Client configuration . . . 147
- 4.4 Cache management as administrator . . . 154
- 4.5 Log management in Lizmap . . . 156

5 Installing and upgrading Lizmap **159**
- 5.1 Requirements before installing Lizmap Web Client . . . 159
- 5.2 Installing Lizmap Web Client on Linux Debian or Ubuntu . . . 162
- 5.3 Installing Lizmap Web Client on Windows . . . 174
- 5.4 Configuration of Lizmap . . . 175
- 5.5 LDAP authentication . . . 176
- 5.6 Advanced installation configuration . . . 179
- 5.7 Upgrade Lizmap Web Client . . . 180

6 Indices and tables **185**

Author Michaël Douchin - 3Liz, René-Luc D'Hont - 3Liz

Date 2014-2021

Contact info@3liz.com

Organization 3Liz

Version 3.5

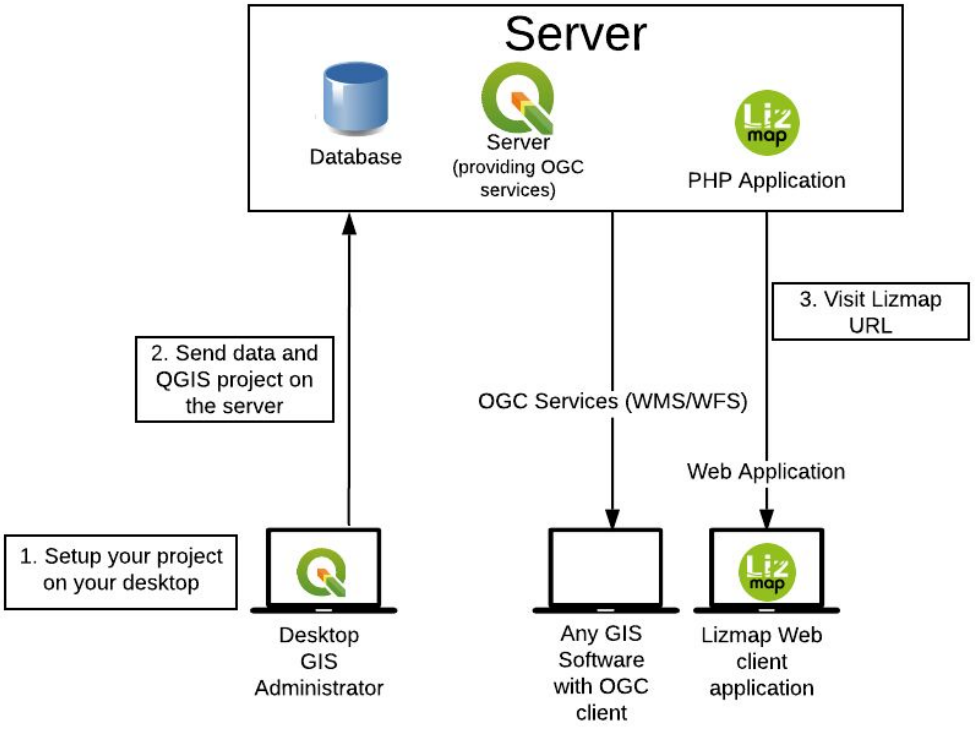
Abstract This document contains the complete documentation of Lizmap, an application to publish online maps: the QGIS plugin and Lizmap Web Client application

CHAPTER

ONE

INTRODUCTION

1.1 Lizmap architecture



Data preparation

The map's construction work is carried on his own computer with **QGIS Desktop**:

- loading layers,
- map creation,
- organising and preparing data.

Map configuration & publication

To be able to put the map online, the publisher configures publishing options (scales, base layers, metadata) with **Lizmap plugin** in QGIS Desktop. Then he synchronizes his working folder with his server in intra- or extranet.

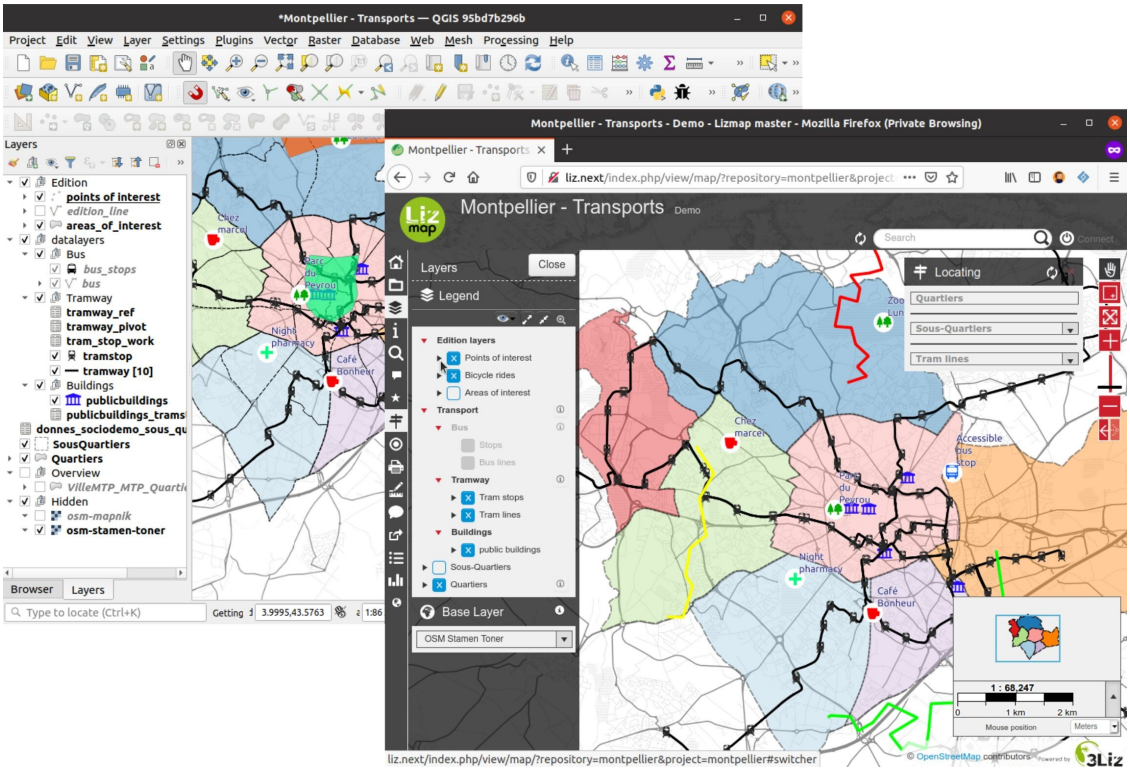
Data vizualisation

When synchronization is complete, The QGIS map can then be accessed on the Internet at the same. It is available on the application **Lizmap Web Client** with a Web browser (Firefox, Safari, Chrome, Internet Explorer).

For more information on versions (Web-browsers, QGIS Server, PostgreSQL etc) which are supported, it's written on the Lizmap Web Client wiki page : https://github.com/3liz/lizmap-web-client/wiki/Versions

1.2 QGIS and QGIS Server

- QGIS, a complete Desktop GIS software
  - Management of a wide range of data format for raster (GeoTIFF, ASC, NetCDF, etc.) and vector (SHP, TAB, MIF/MID, GML, GeoJSON, etc)
  - Management of key geographic database (PostgreSQL/PostGIS, Oracle, MSSQL, etc.)
  - Management of raster and vector layers, symbology, printing, processing
  - Performance and functional richness of a desktop GIS Software
- Geographical standardized Web Services:
  - *Open Geospatial Consortium*, OGC, consortium for standardization of geographic information exchange through Internet
  - *Web Mapping Service*, WMS: standard for publishing map as image
  - *Web Feature Service*, WFS: standard for publishing geographic vector data
  - *Web Coverage Service*, WCS: standard for publishing geographic raster data
- QGIS Server allows you to easily publish 3 types of standardized Web Services Geographic:
  - Install QGIS Server on a server
  - Configure properties of Geographic Web Services
  - Copy and paste the QGIS project and data files
  - QGIS Server and QGIS Desktop share **the same rendering engine**



1.3 Lizmap features

1.3.1 Open Source

Lizmap and the modules are opensource projects edited and managed by https://3liz.com

The source code is currently available on https://github.com/3liz/ Help is available through :

- https://gis.stackexchange.com/
- Github tickets
- Our mailing-list : https://lists.osgeo.org/pipermail/lizmap/
- On IRC on the #lizmap channel of https://libera.chat
- Commercial support through 3Liz, contact mailto:info@3liz.com?subject=CommercialSupportRequest

1.3.2 Default features

By default, Lizmap offers web maps with the following features:

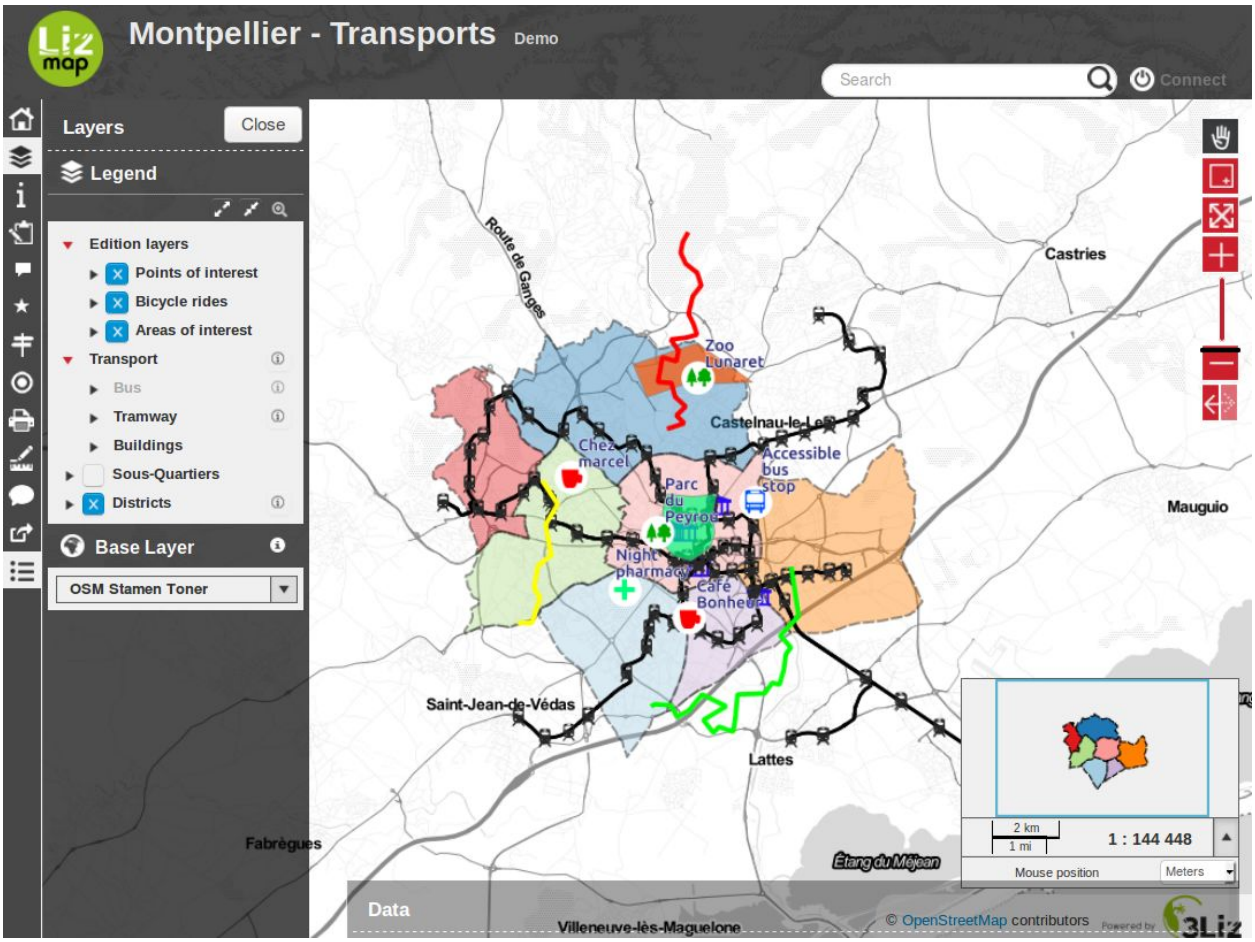
- pan
- zoom on area drawn by the user
- zoom more
- selecting a zoom level with a level bar
- zoom less

- displaying the scale as a scale bar and digitally

The displayed legend respects the order and organization defined in QGIS.

The interface allows the user to hide the legend and take advantage of the map in the entire width of the screen. It also lets you view the information on the map (description, extent, contact).

Finally Lizmap is a proxy to the web services used to construct maps. It is therefore possible to use WMS or WFS requests through Lizmap. It is possible to secure access to web services for access in QGIS or ArcGIS. It is also possible to share these addresses with web services systems cataloging.



1.3.3 Additional Lizmap modules

Some modules can be added to Lizmap :

- **AltiProfil** Display a profil with altitude according to a line and a DEM (Digital Elevation Model)

  https://github.com/arno974/lizmap-altiProfil

- **Map builder** To combine many Lizmap projects into a single map viewer

  https://github.com/3liz/lizmap-mapbuilder-module

- **PgMetadata** For displaying metadata stored with the QGIS PgMetadata plugin. It is also providing a DCAT XML catalog, to be harvested by external services

  https://github.com/3liz/lizmap-pgmetadata-module

- **PgRouting** For displaying itinerary between two points using the PgRouting extension for PostGIS.

  https://github.com/3liz/lizmap-pgrouting-module

- **Web Processing Service** To execute QGIS Processing algorithms or models on a server and view the results within Lizmap

  https://github.com/3liz/lizmap-wps-web-client-module

Specific to France use-case :

- **French adresse** For managing addresses

  https://github.com/3liz/lizmap-adresse-module

- **French cadastre** For managing cadastre data

  https://github.com/3liz/lizmap-cadastre-module

- **Naturaliz** For managing species and observation data

  https://github.com/3liz/lizmap-naturaliz-module

1.3.4 Configuration with QGIS

Map's configuration is done in QGIS.

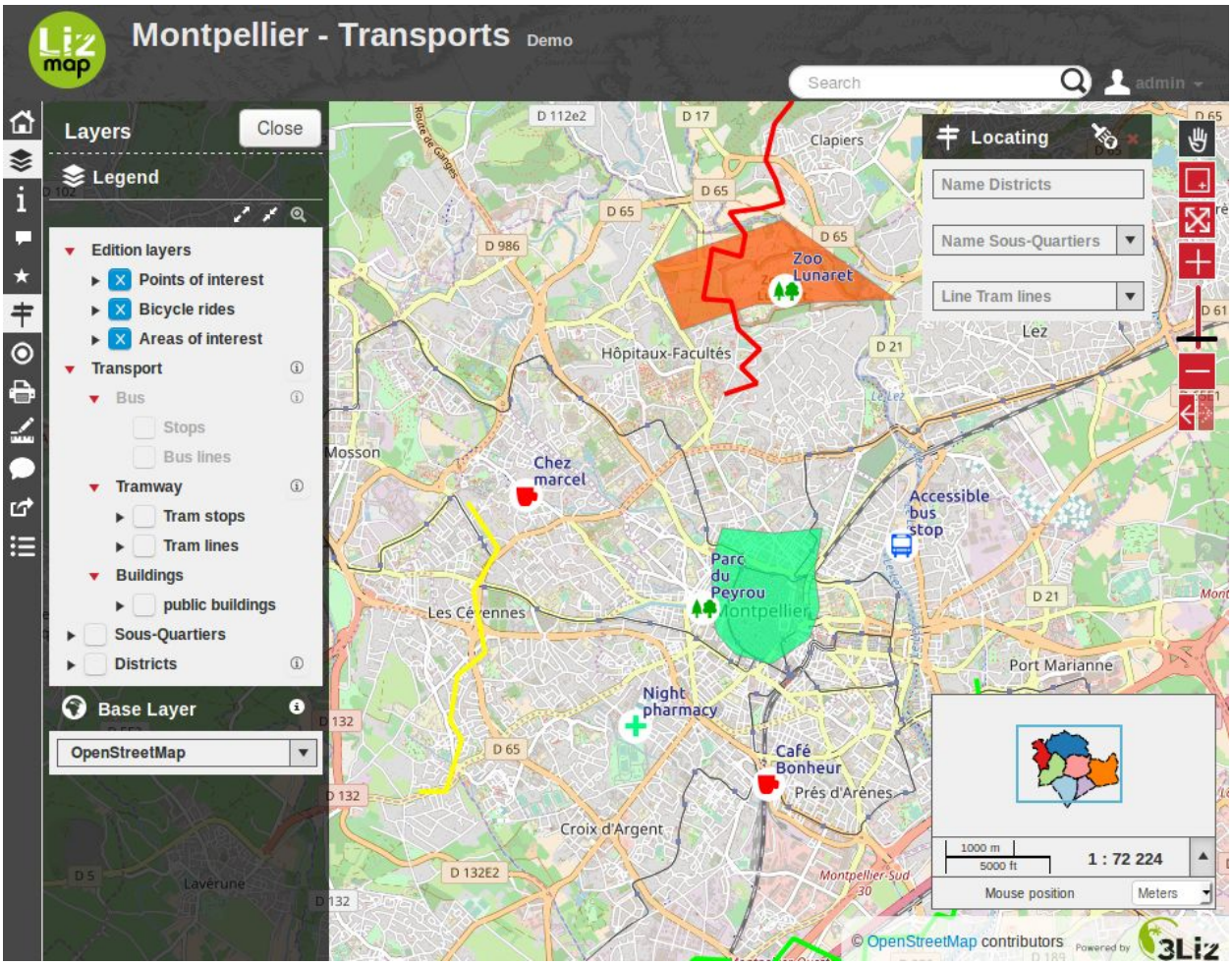
One part comes directly from the **QGIS project**:

- legend order and structure
- layer symbology
- location map with a group Overview in the legend
- configure print templates via composers

The rest is done with the **Lizmap plugin**. This plugin allows to:

- specify the layers and groups name in the Web interface
- group the layers into a single WMS layer
- set some layers as baselayer
- set the display layer system: alone image or tiles
- set the lifetime of the images in the cache
- set the lifetime of the images in the network
- set the popups for querying layer
- set a link to the metadata
- set the map scales
- set the tools available with the map:
  - printing
  - measuring length, area and perimeter
  - using user geolocation (GPS)
  - address search in OpenStreetMap or Google
- set external baselayers:

  - Google Maps
  - Bing Maps
  - OpenStreetMap
  - Géoportail IGN
- set layers for navigation shortcut:
  - search in a layer field
  - zoom to the selected item
  - cascaded search within 2 fields of the same layer or between joined layers



1.3.5 Configuring with the Web back-office

The Lizmap back-office defines:

- the directory list containing maps
- users' rights by map directory
- using subdomain to improve the responsiveness of the interface.

1.3.6 In a nutshell

- HTML5 interface compatible mobiles and tablets
- Grouping maps by directory: application multi-directories and multi-projects
- Management of access rights per directory for user groups
- Automatic map cache management
- Multilingual interface: English, French, Italian, Spanish, Portuguese, Greek
- Using external baselayers: OSM, Google, Bing
- Printing
- Online edition: adding, spatial and filed changing, deleting
- Zoom with localisation
- Address search: Nominatim (OpenStreetMap engine), Google
- Access logs

1.3.7 Requirements

Version of Lizmap Web Client requires:

- Firefox ≥ 63
- Edge ≥ 79
- Chromium ≥ 54

CHAPTER

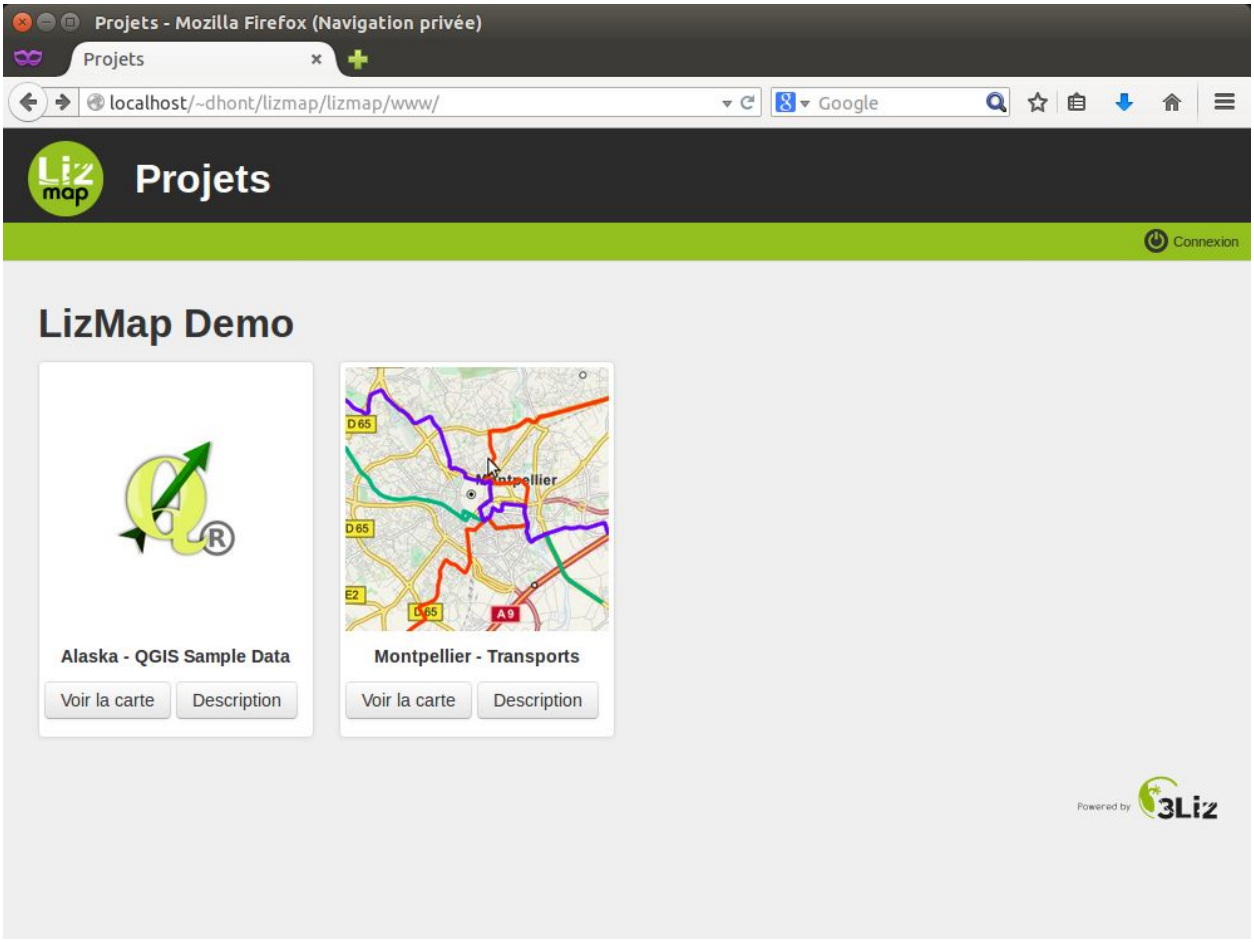
TWO

USER GUIDE

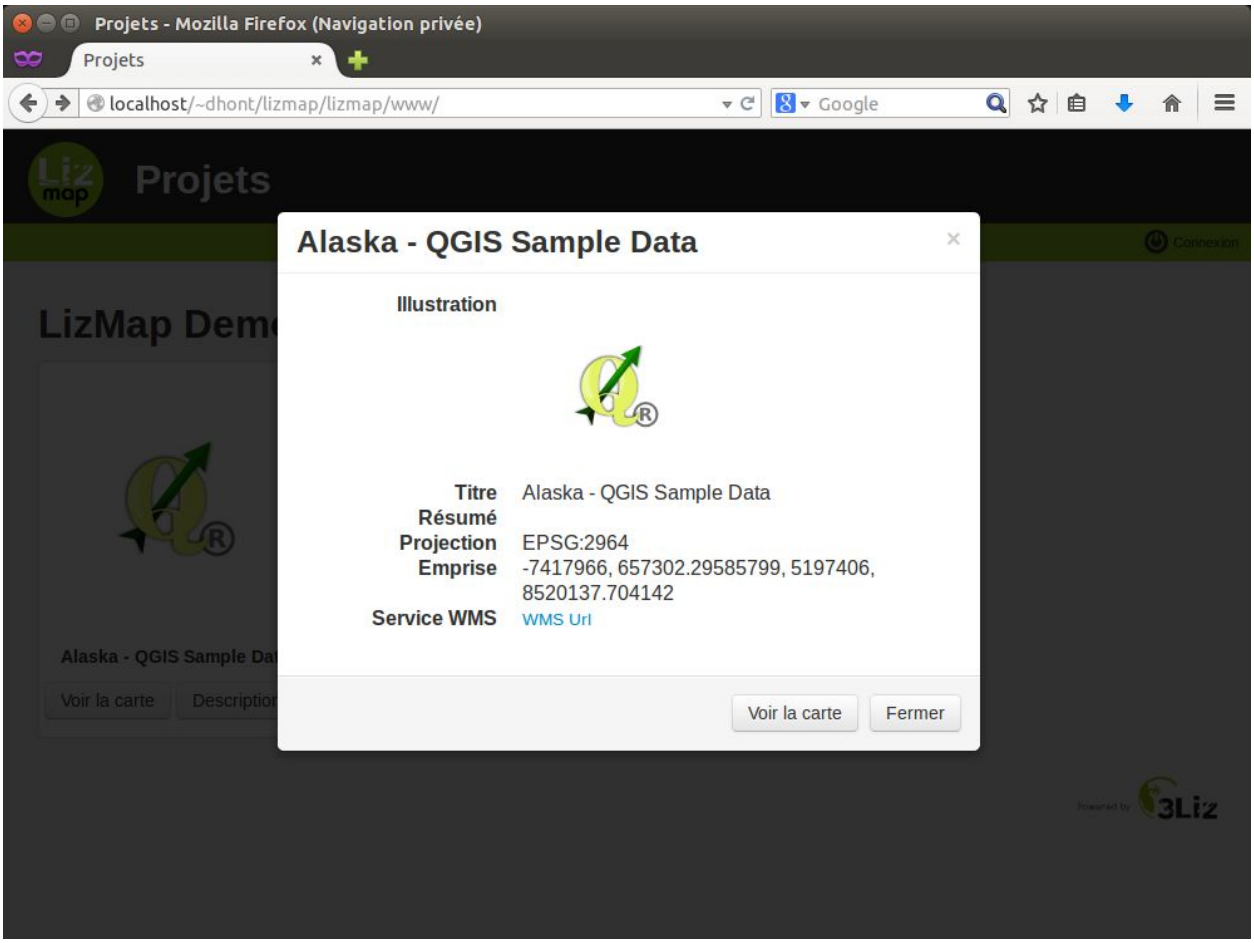
This guide is for the end user, when using the web client.

2.1 The projects page

The default lizmap page offers a list maps organized by folder.



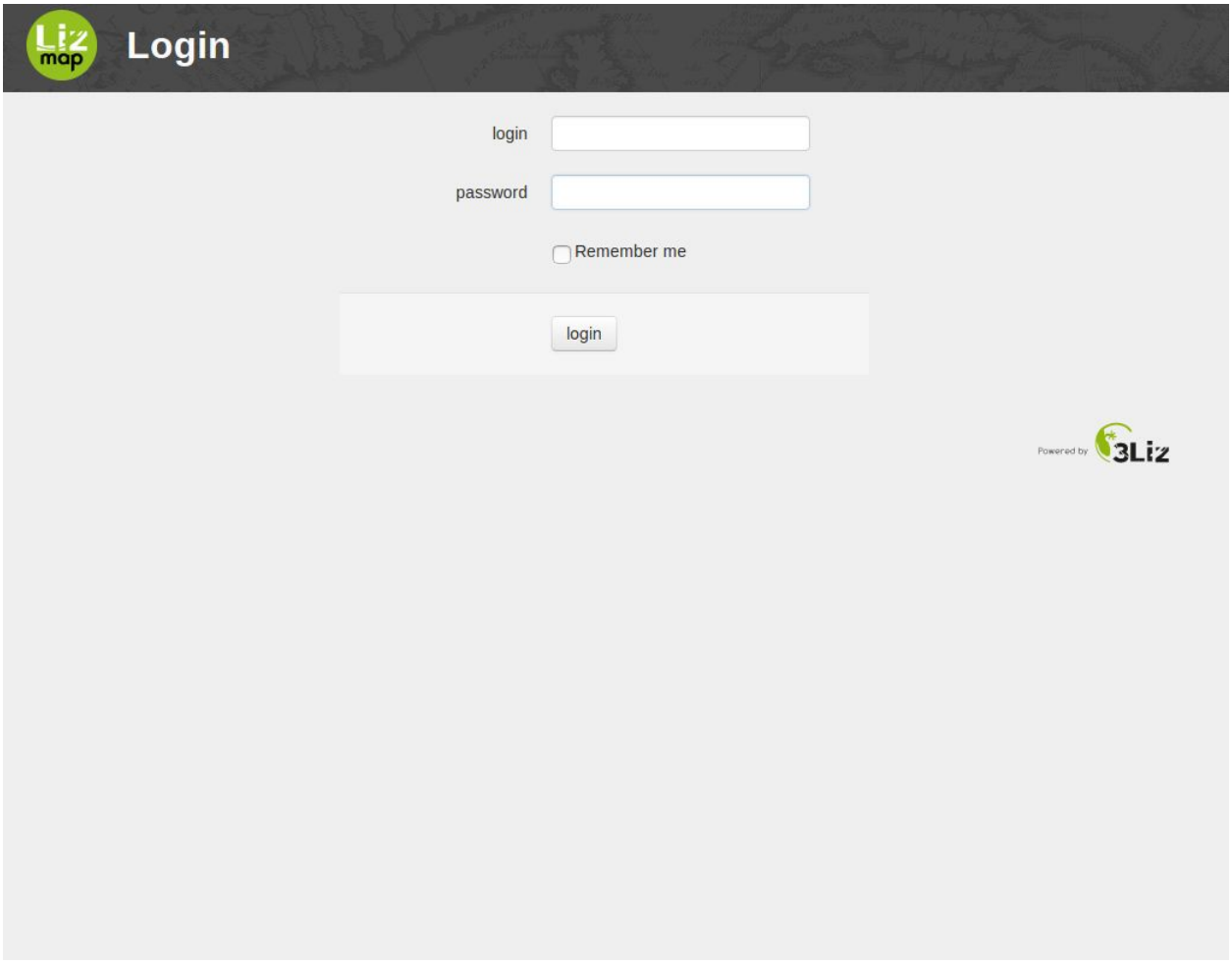
You can access to the map information sheet. This informations come directly from QGIS project.



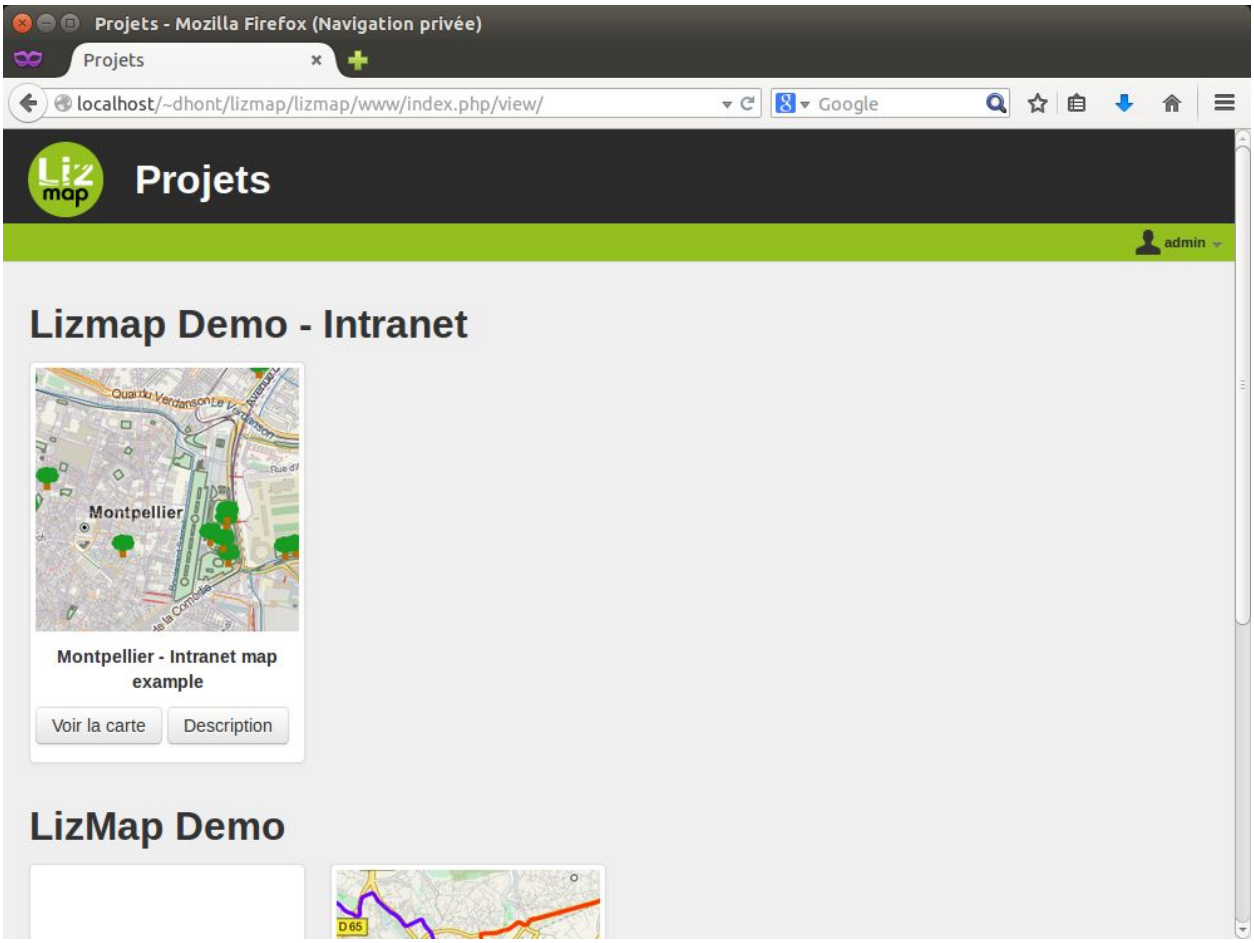
Access to the map is done either by clicking the **Load the map** buttons or the image that accompanies the project.

2.2 Authentication

The admin can restrict access to certain map groups. To access these maps, you must authenticate. Authentication is accessible through the button *Connect* at the top left.



Once authentication is enabled, depending on your rights, you should have access to new maps.



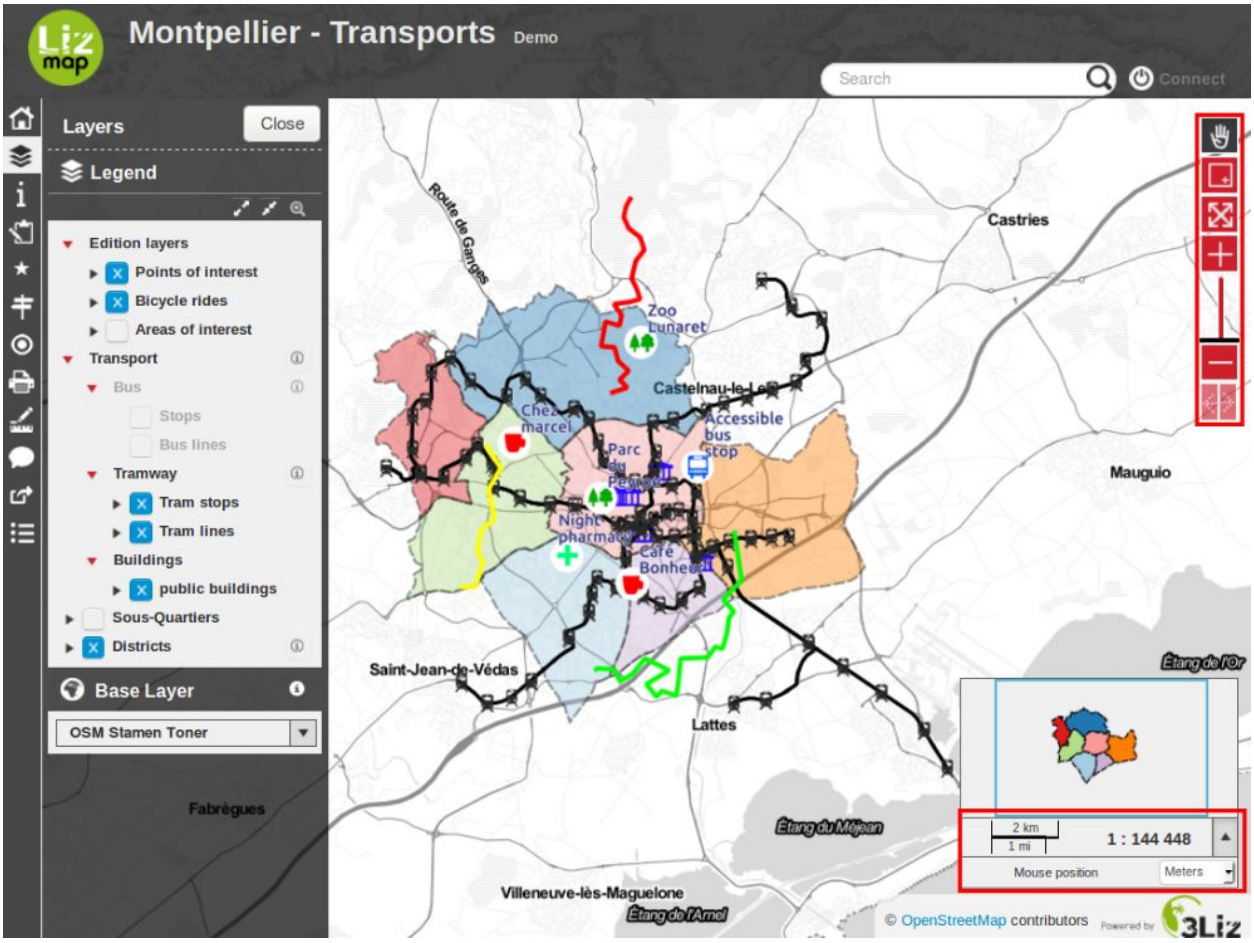
You can log out and edit your user information.

2.3 A simple map

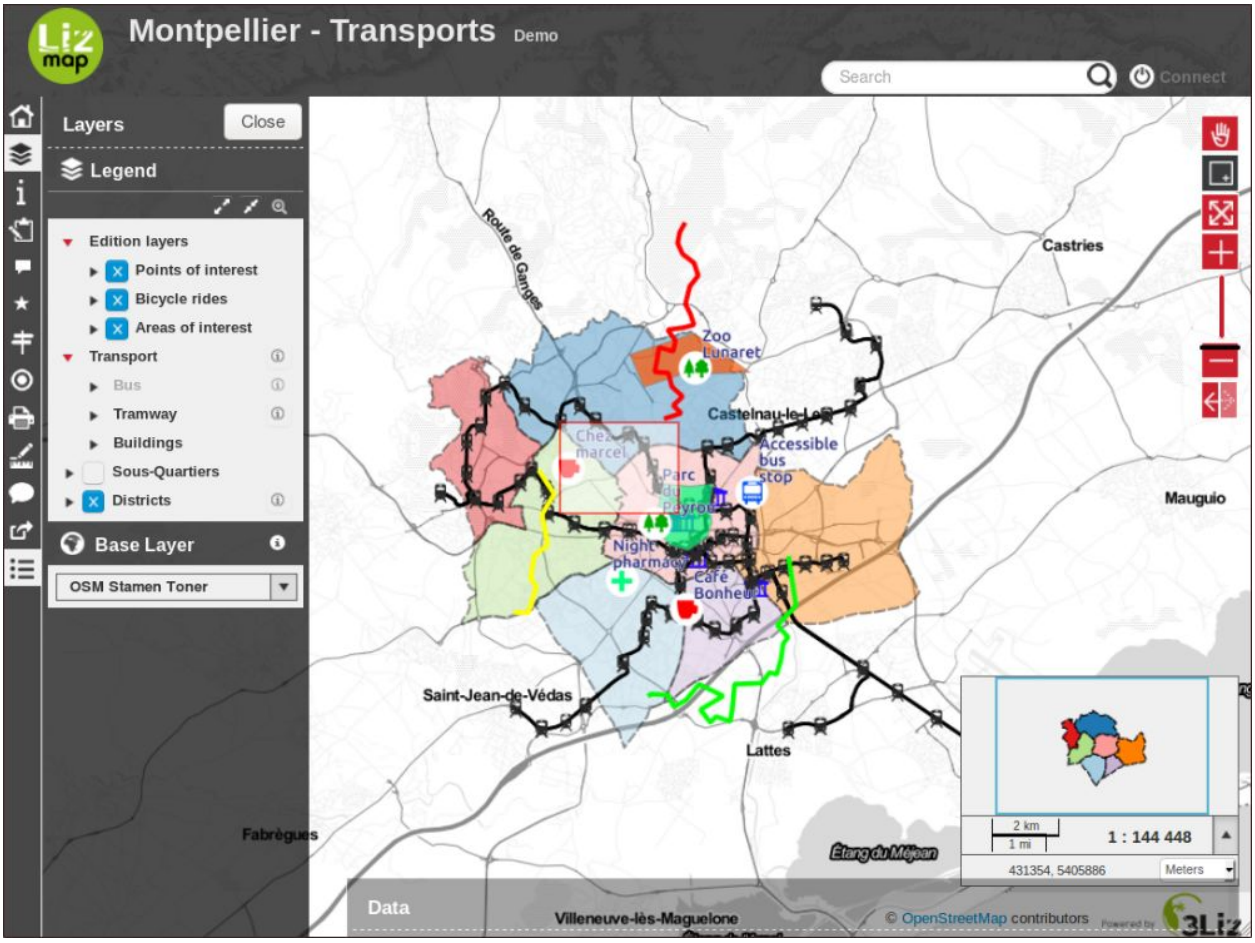
Lizmap offers default web maps with the following features:

- pan
- zoom on area drawn by the user
- zoom more
- selecting a zoom level with a scale bar
- zoom less
- scale display as a scale bar and numerically

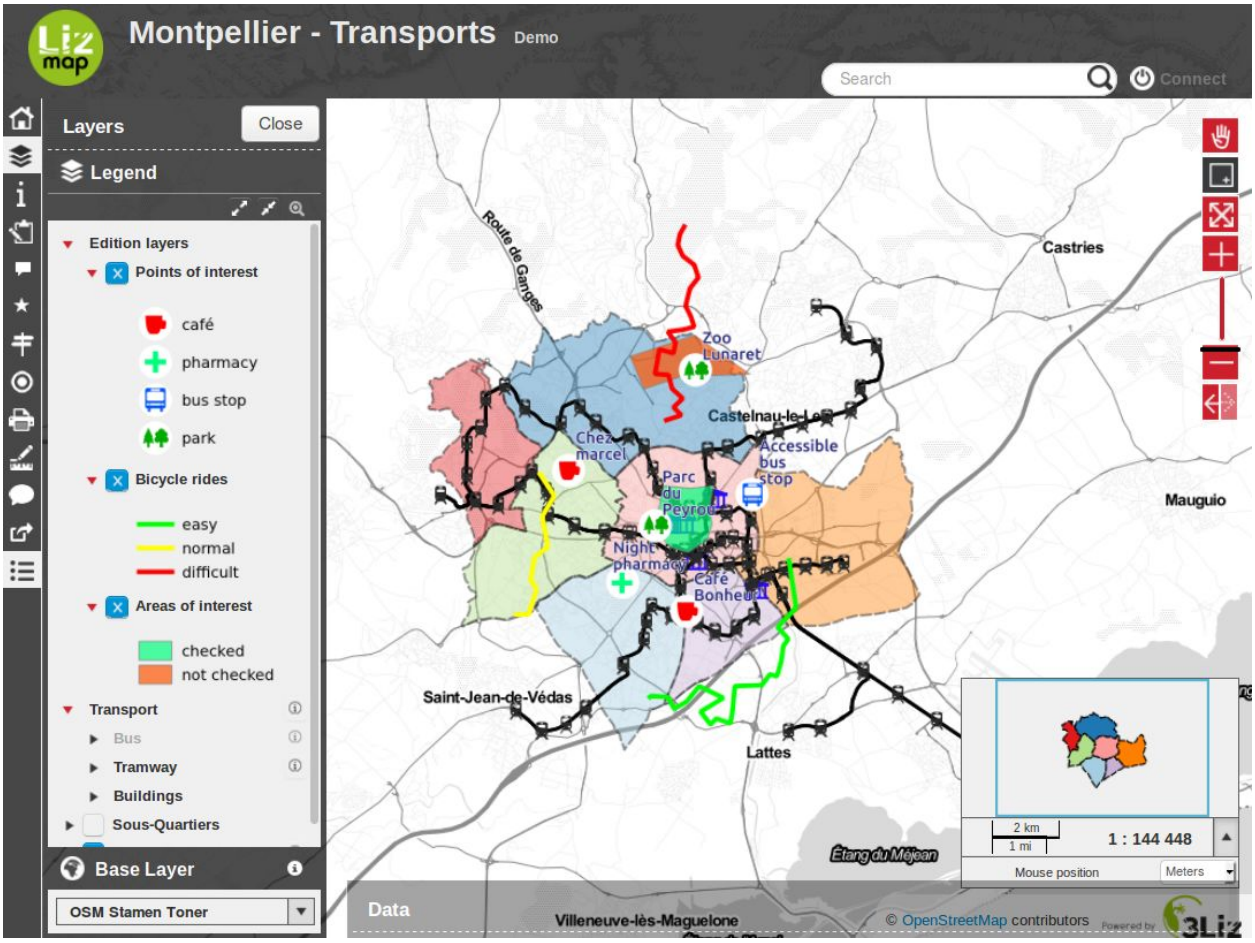
The order and organization of layers in the layers panel management meet those defined in the QGIS by the map publisher.



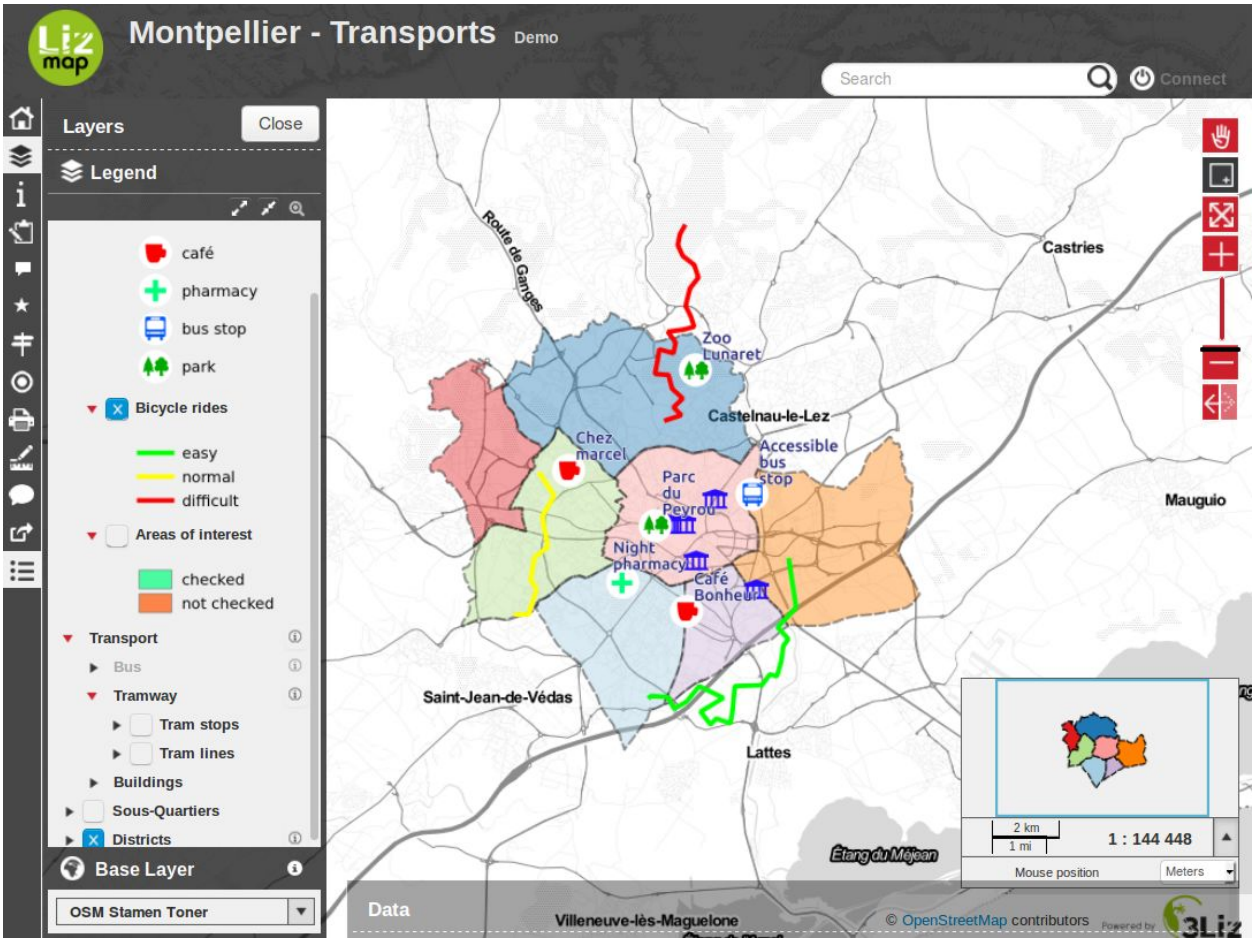
Zooming and paning are available on the right of the map. To zoom on an area, you need to select the *zoom by rectangle* and drag to draw a rectangle defining the area to reach.



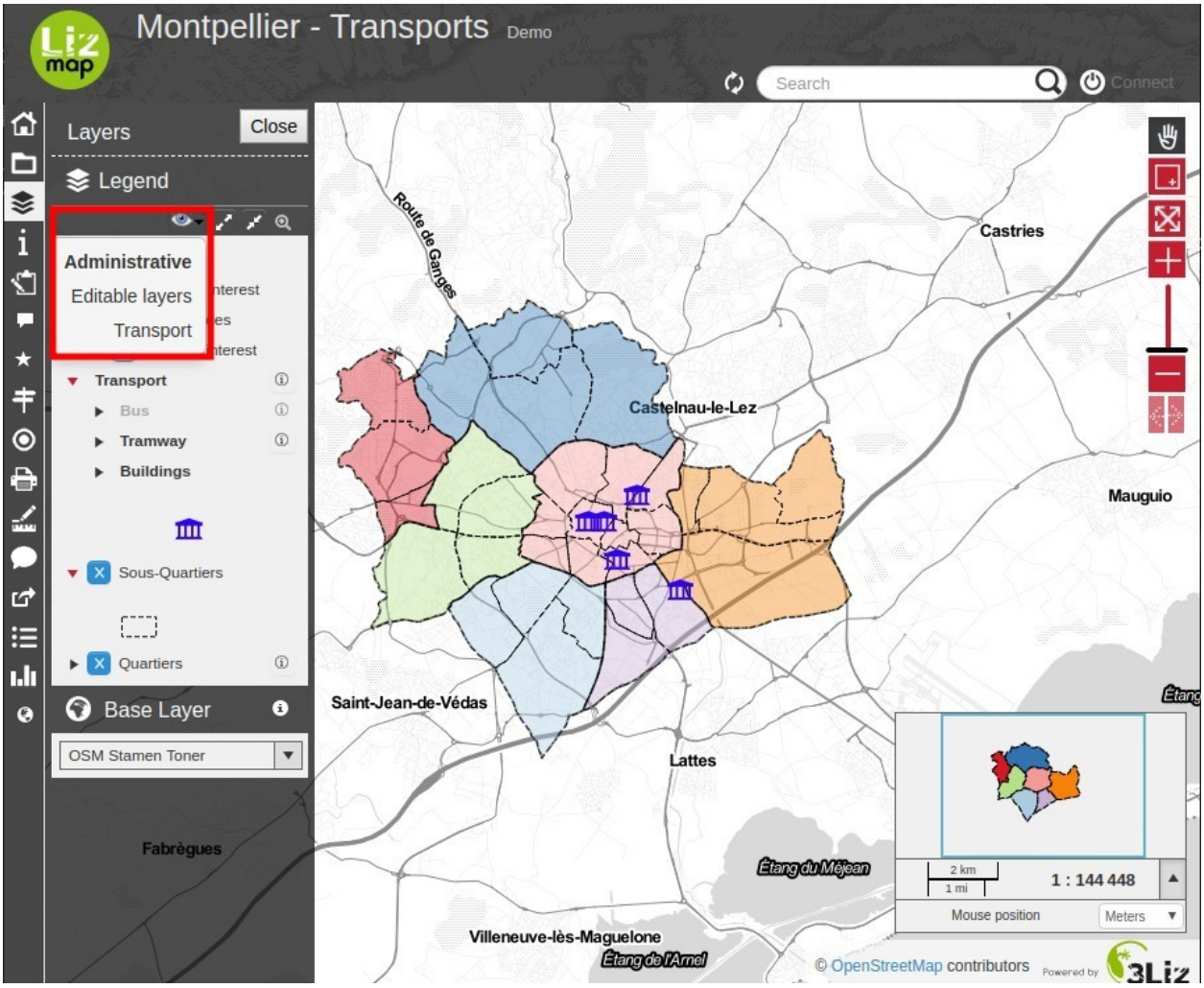
You can using the *triangle*, left the layers title, bring up the legend of the layer.



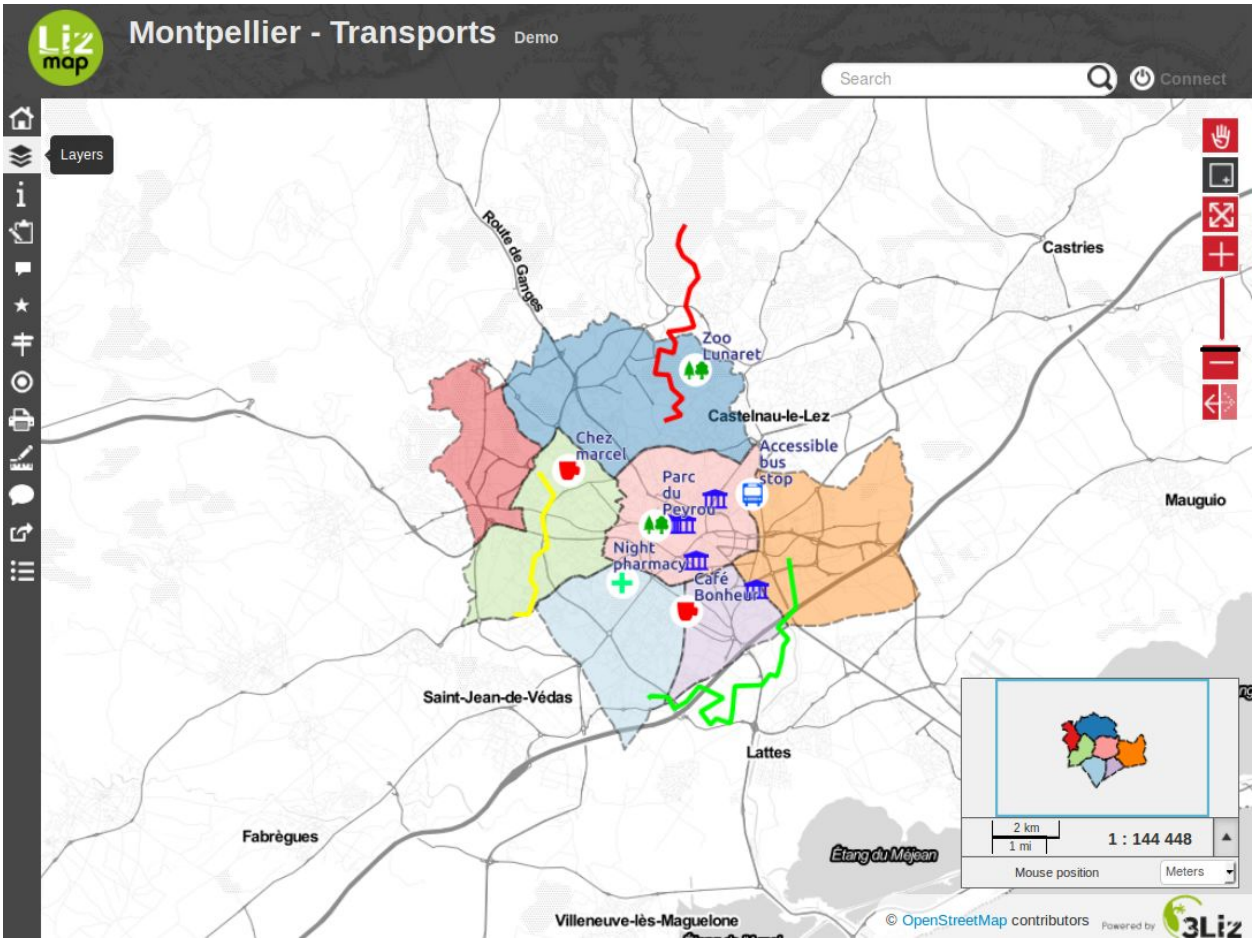
Selection boxes allow you to hide and display the proposed layers.



Themes allow you to change layers visibility following a context.



To take full advantage of the map, you can hide the panel management layers.

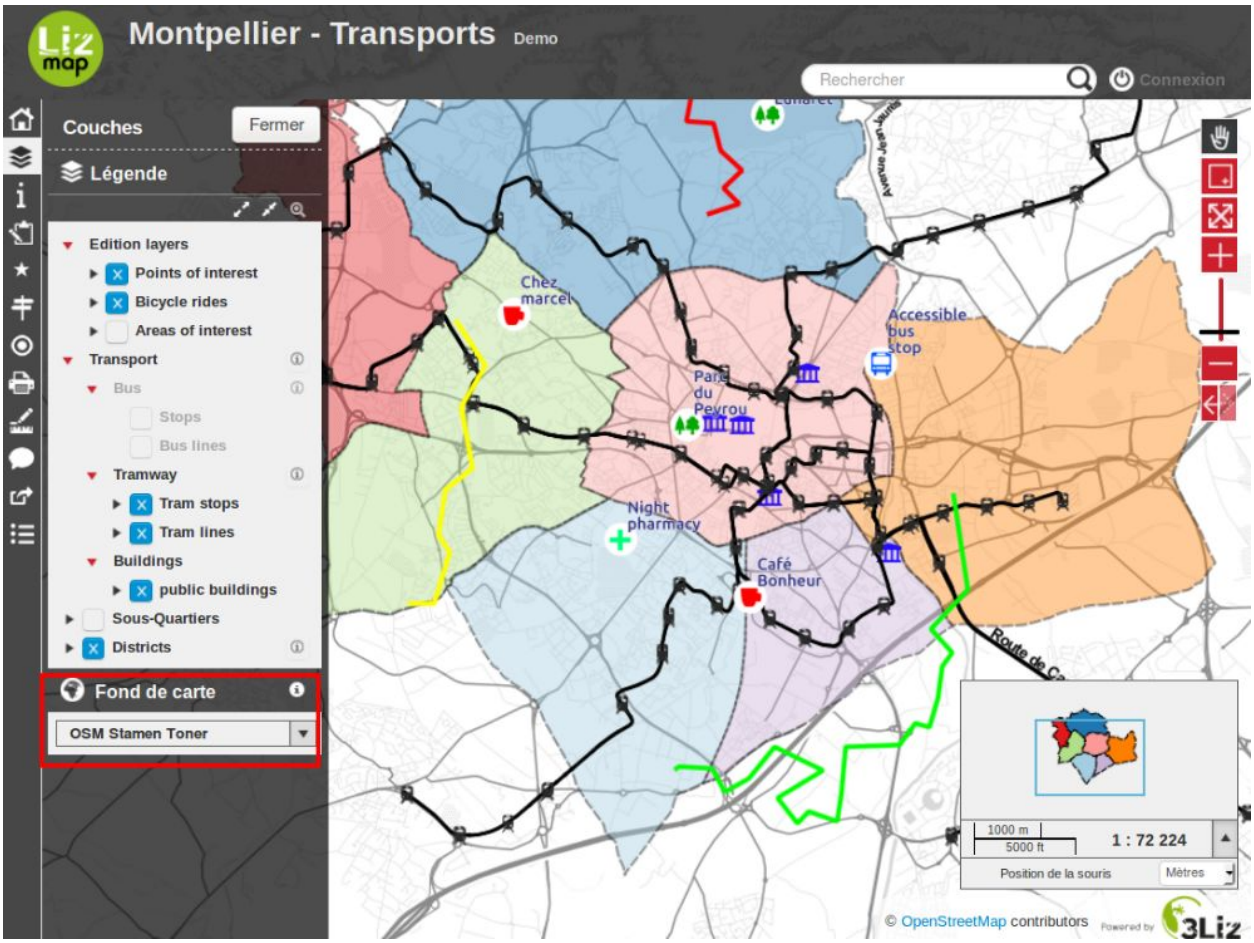


Finally you can:

- go back to the projects page
- display the map information sheet

2.4 Selecting a basemap

Using the Lizmap plugin, the editor may have added external base maps or an empty base layer to the map. These base maps are available in the layer management panel as a list.

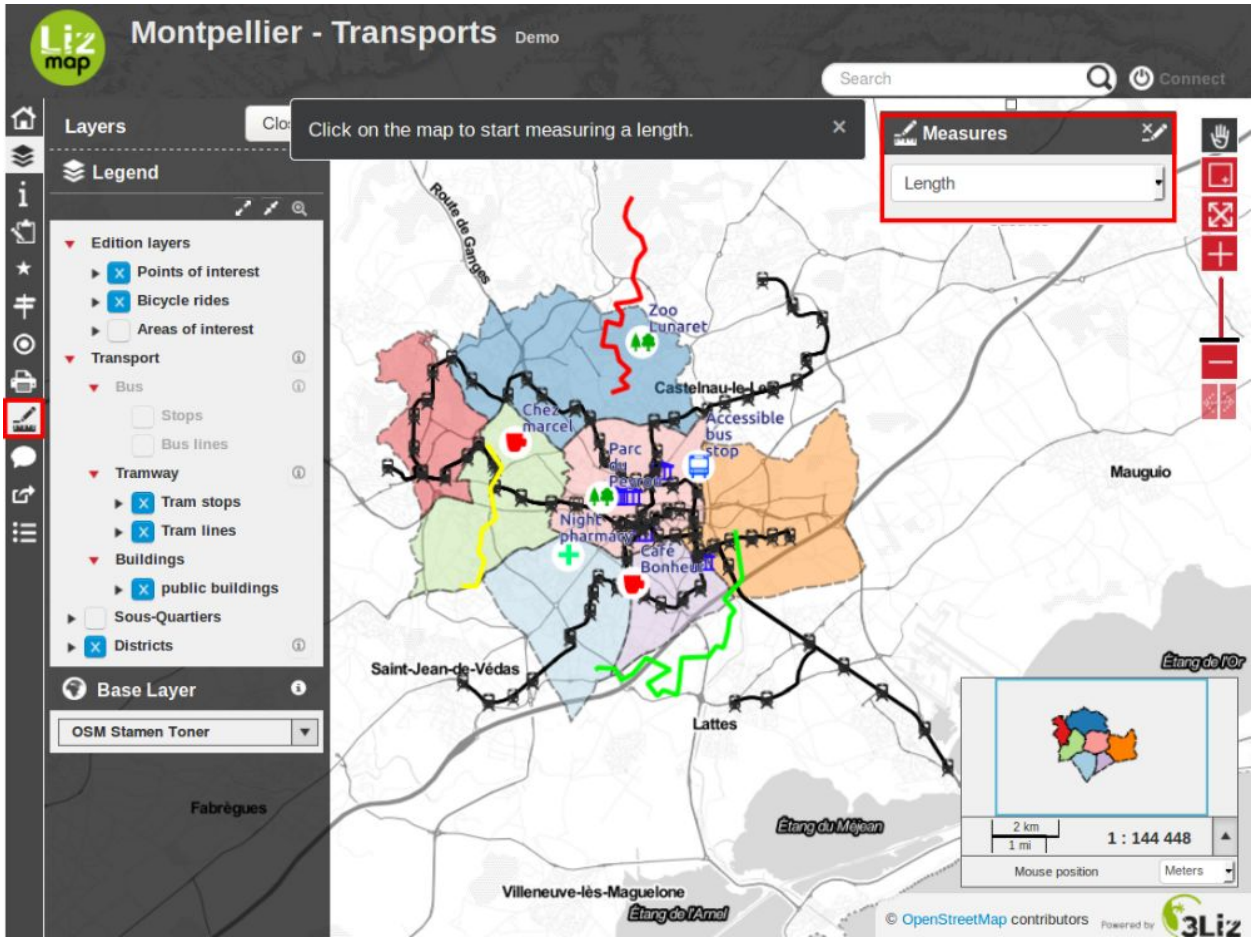


2.5 Measurement

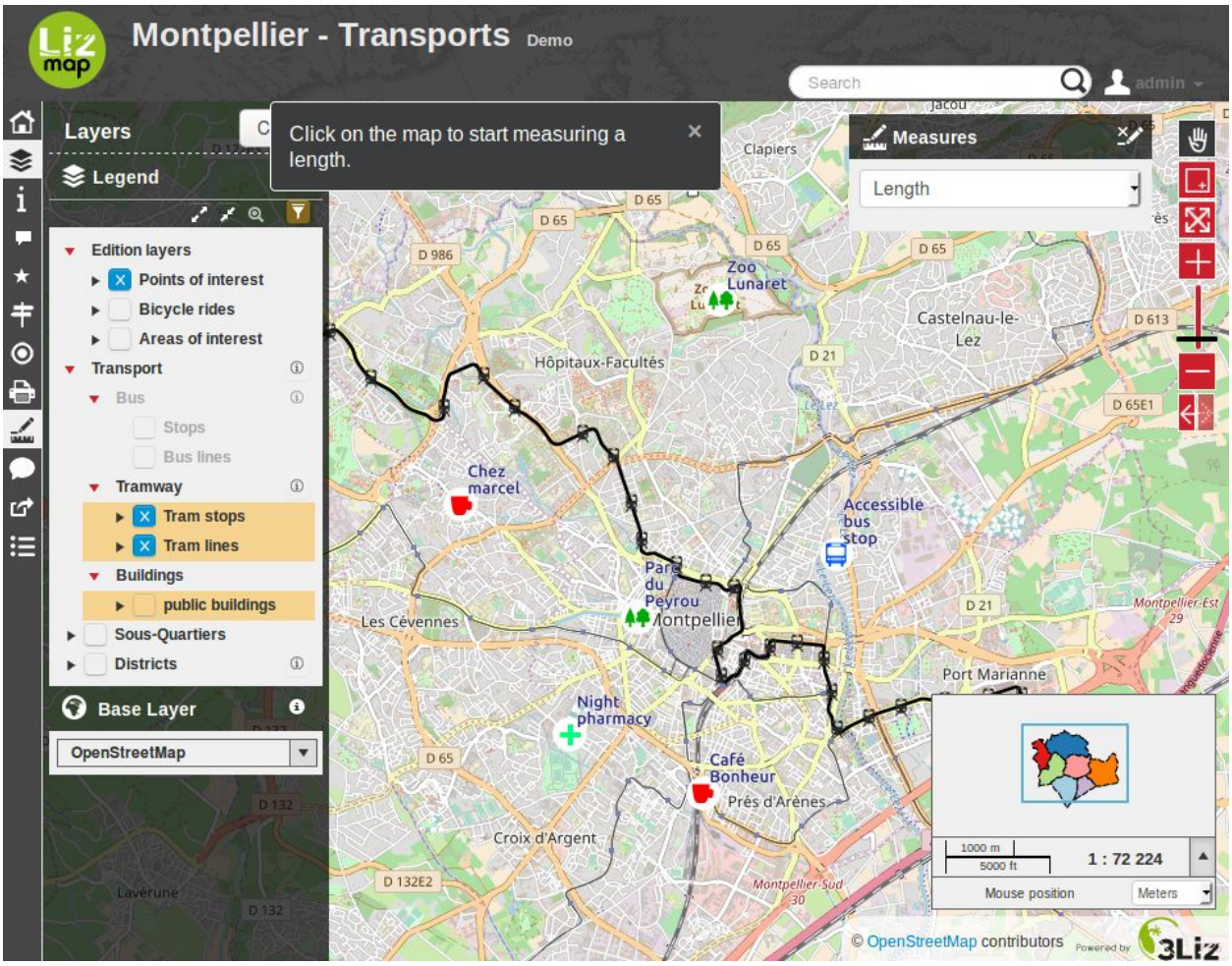
Measurement feature give you the ability to calculate:

- a distance
- an area
- a perimeter
- an angle

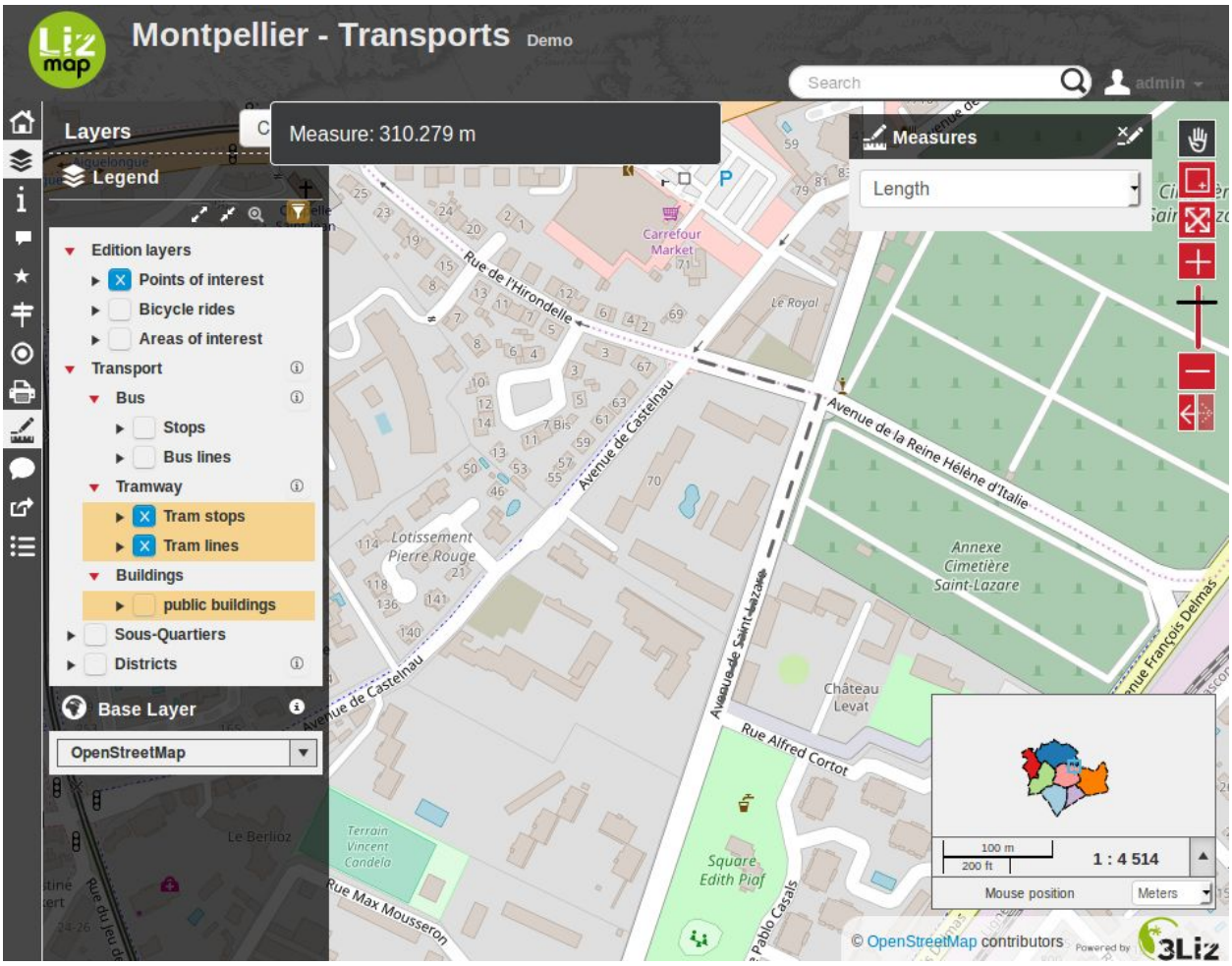
It is available in the Lizmap menu bar.



The tool is activated by selecting the measurement type. Once activated, a message tells you what to do.

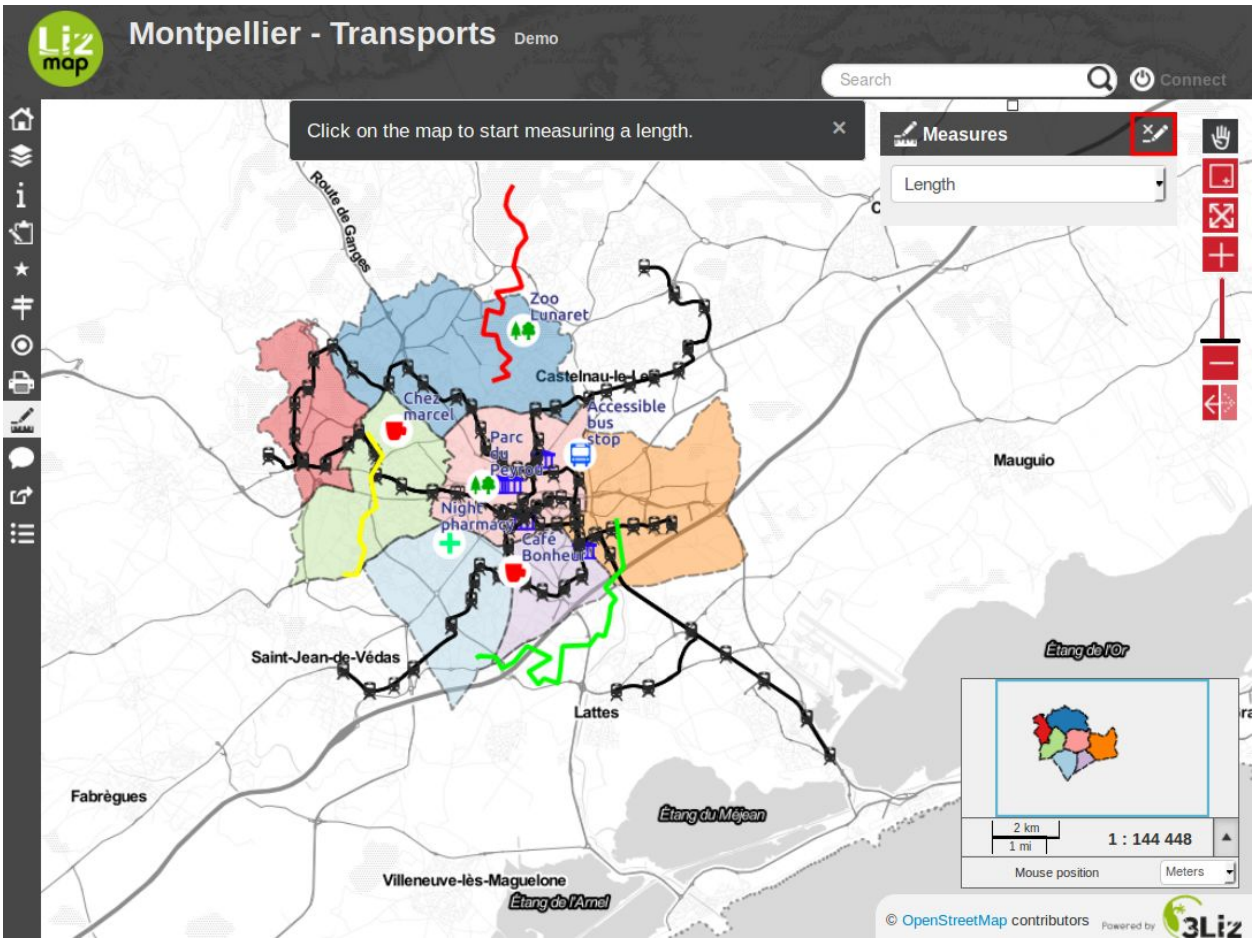


The measurement is displayed in the status bar.



By double-clicking on the map, the measurement is fixed. To start over, you can click on the map and restart the measurement calculation.

In the feature bar, the button on the right allows to stop using it.

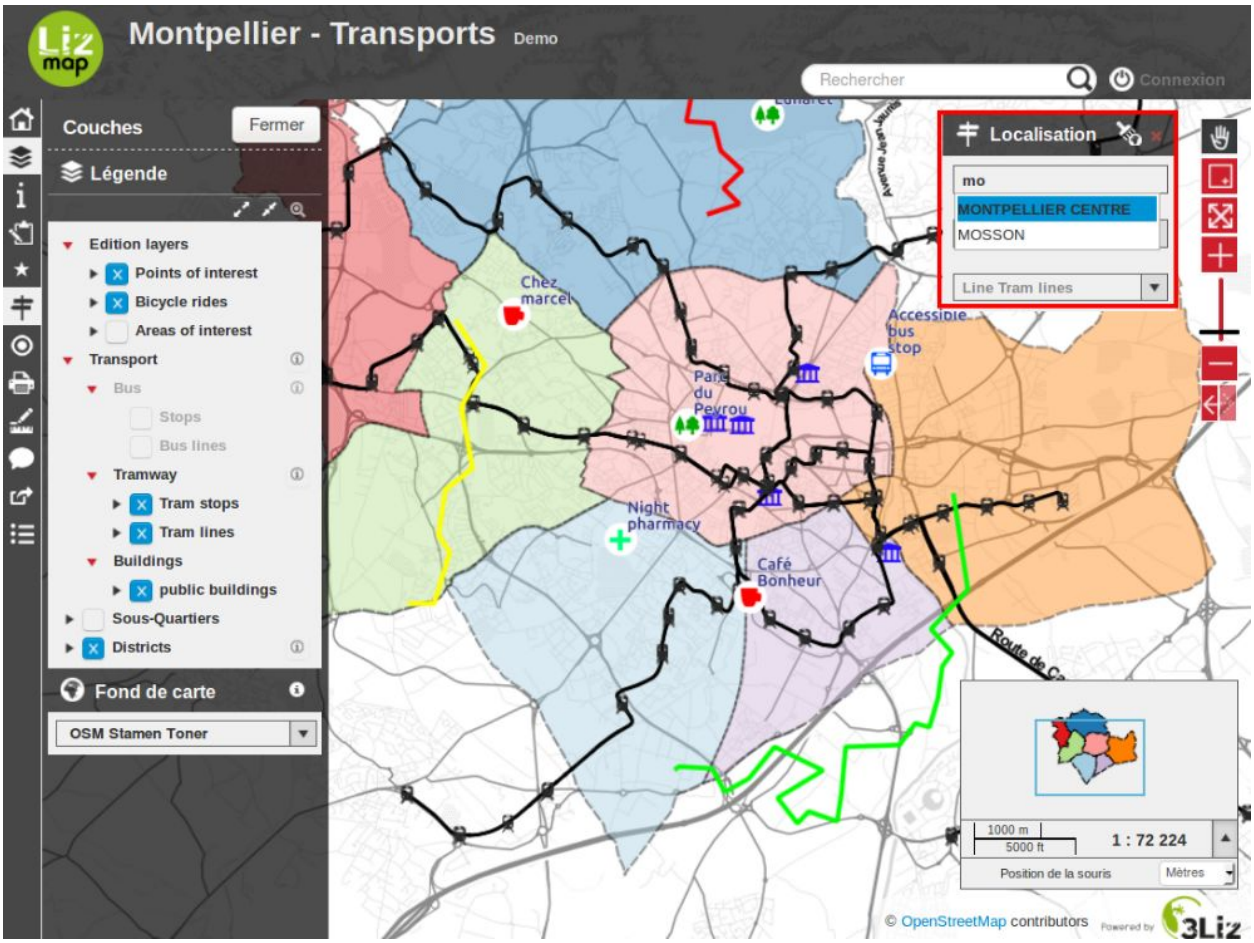


It is also possible to change the measurement functionality without having to stop the feature.

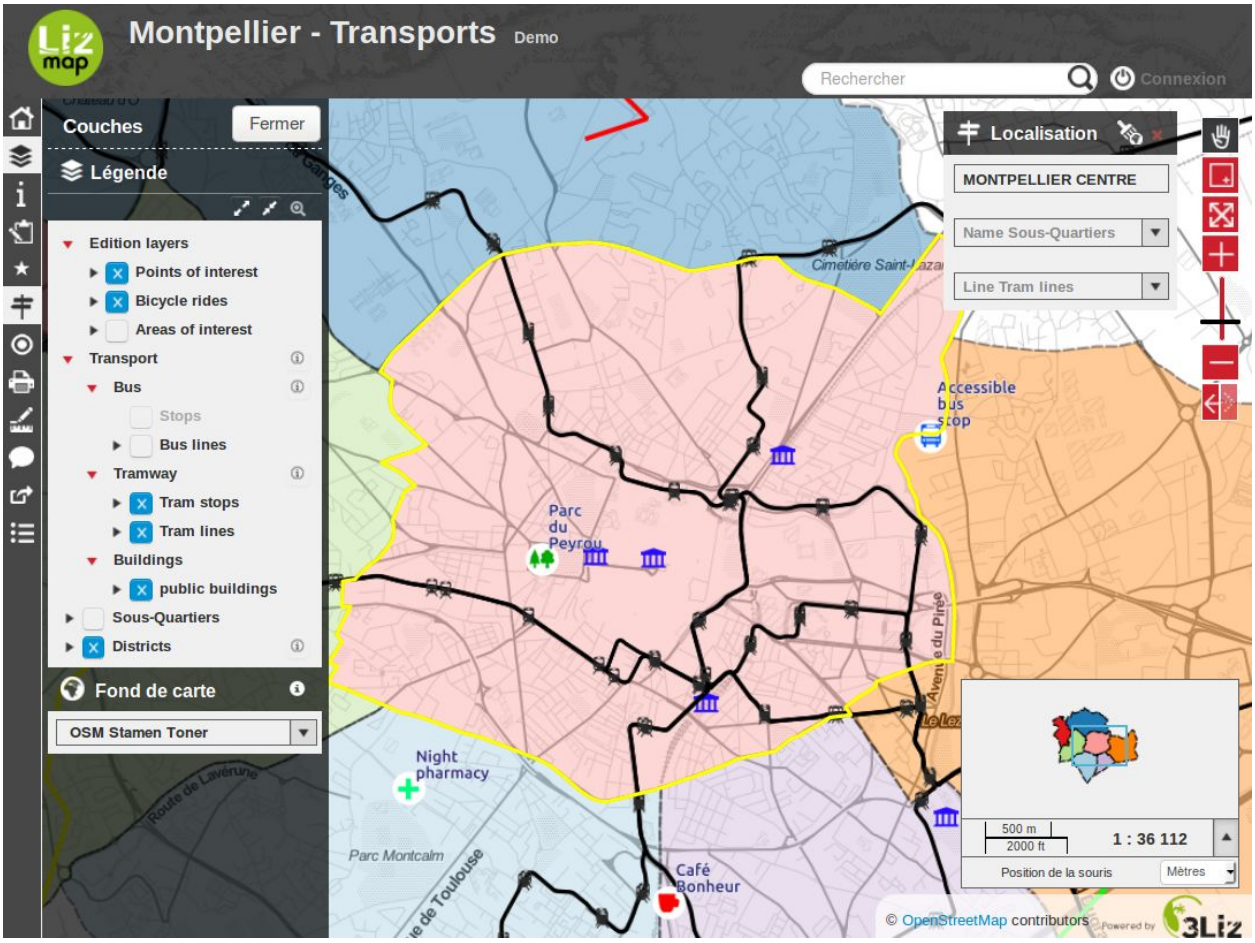
2.6 Locate by layer

This feature is displayed by default if it has been activated by the map publisher.

It is located above the layer management panel and is in the form of lists. Some lists need to enter a few characters before proposing locations.

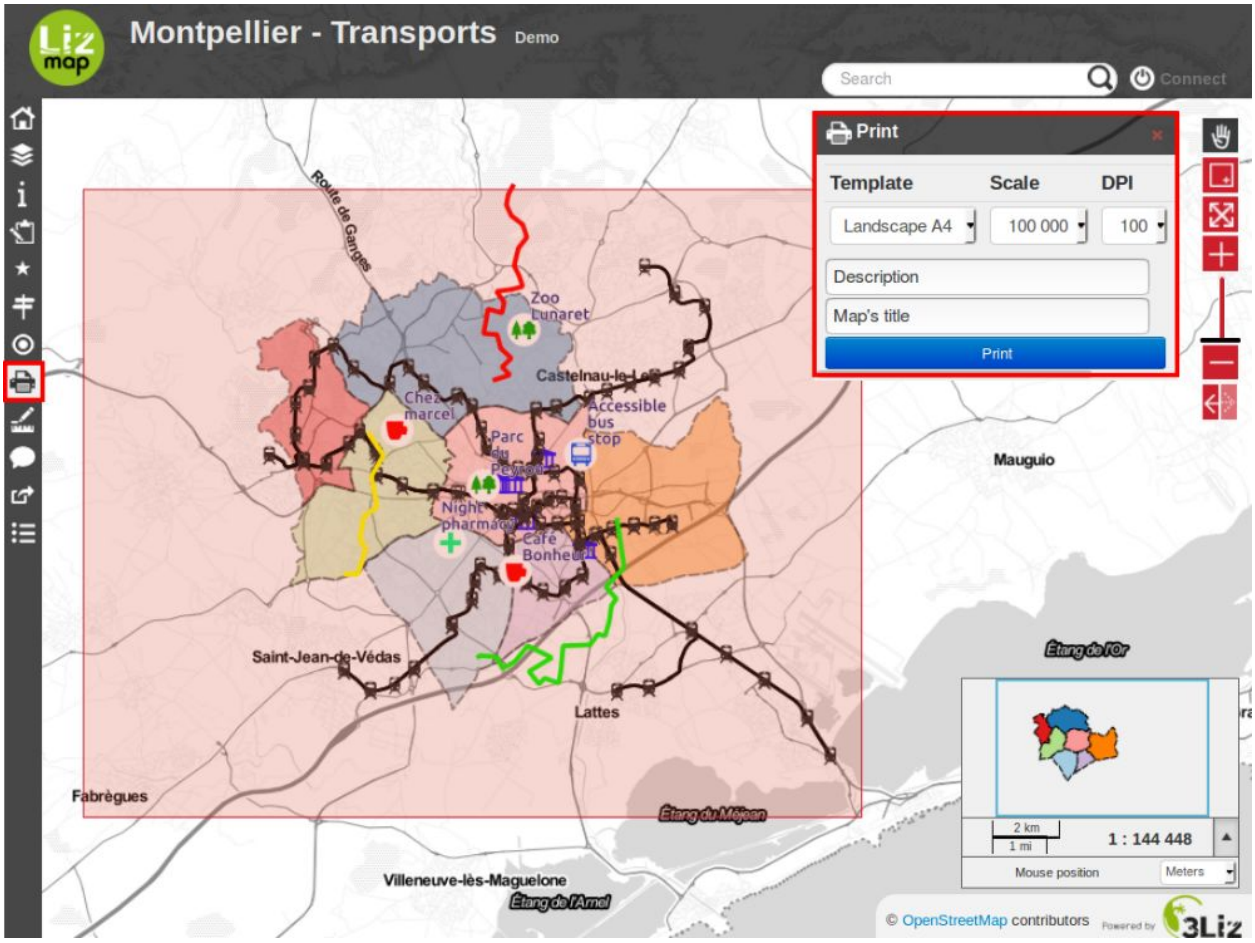


Simply select a location from the list to zoom to the item.

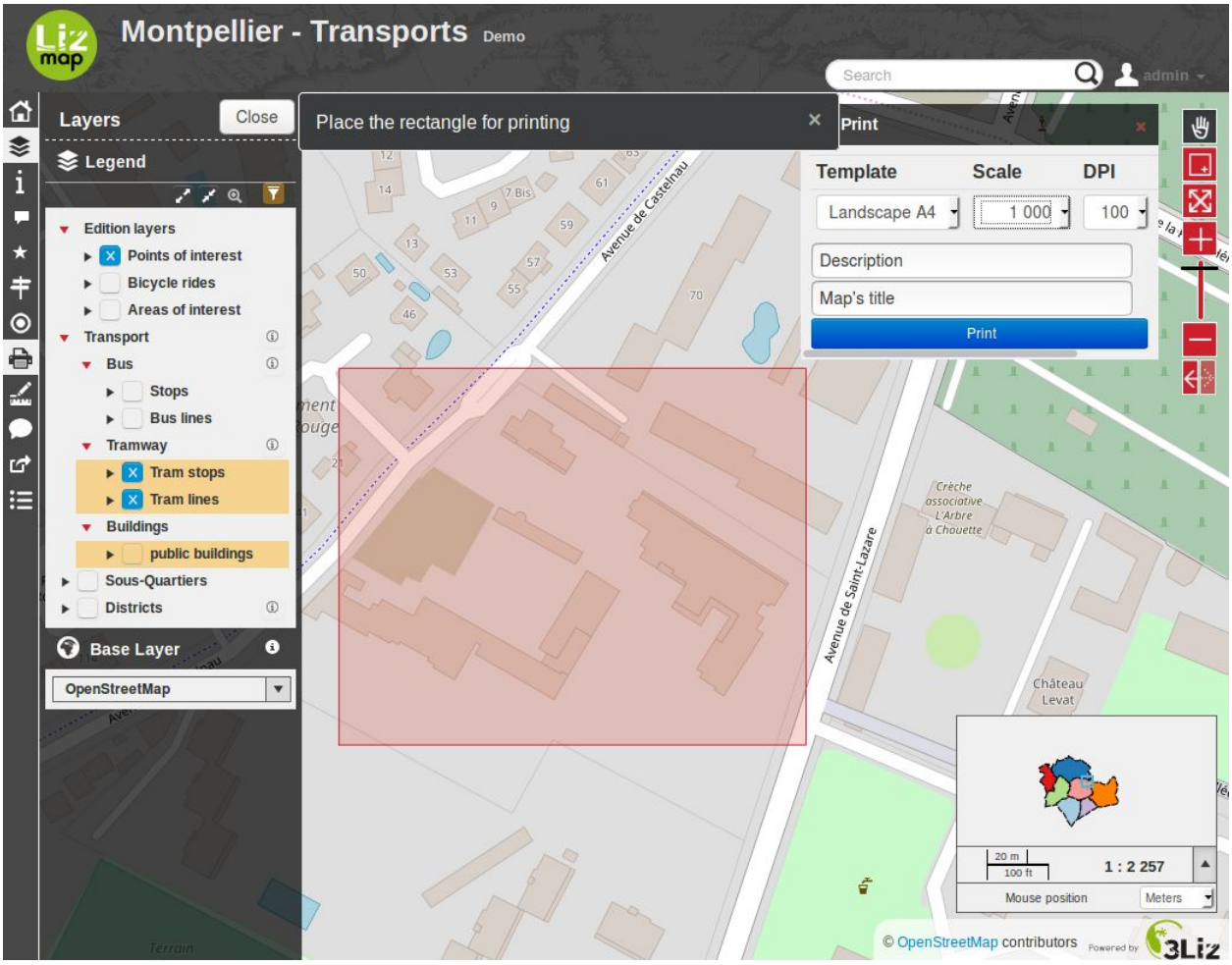


2.7 Printing

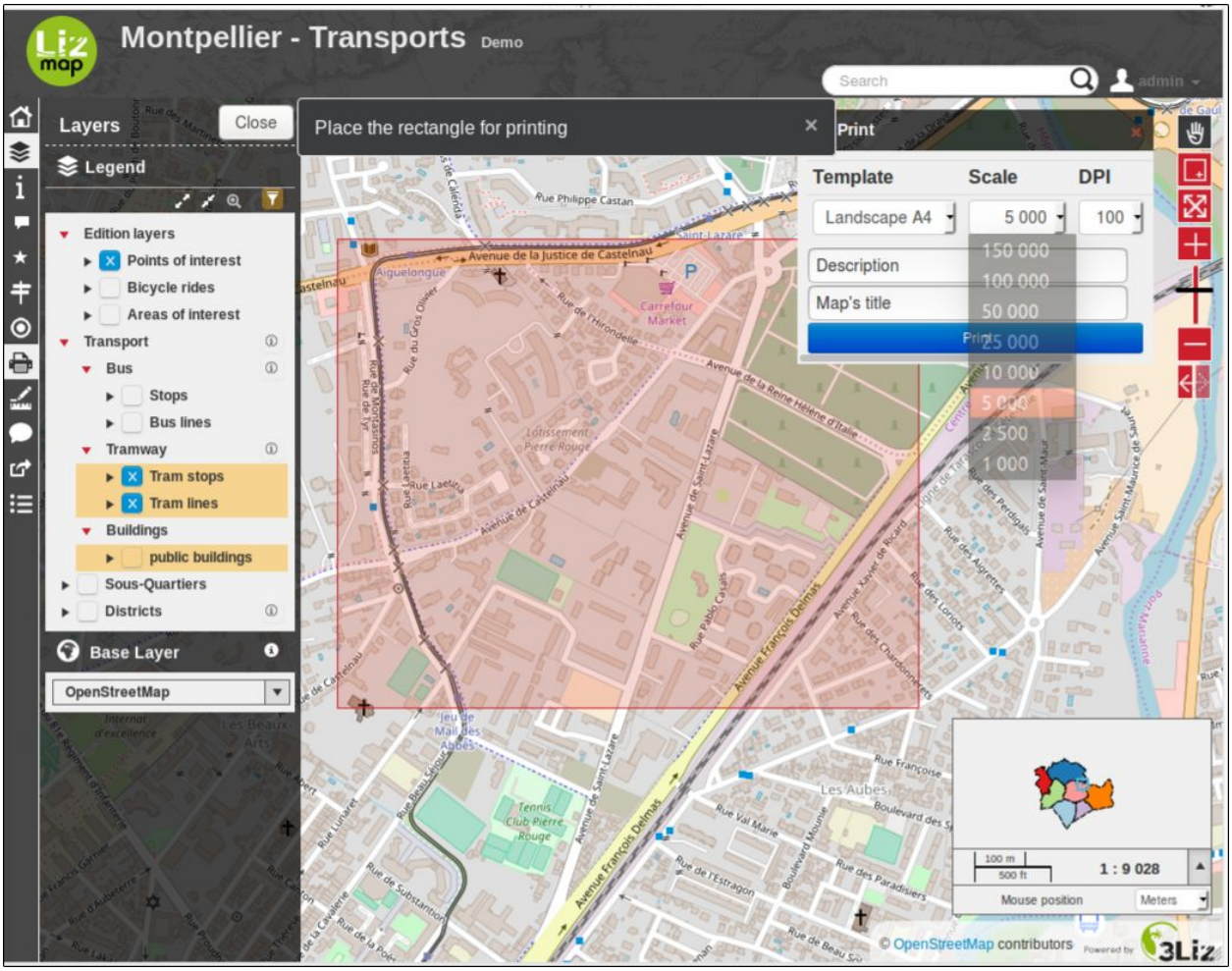
The print feature is available in the Lizmap menu bar. It depends on the number of print compositions published by the map publisher.



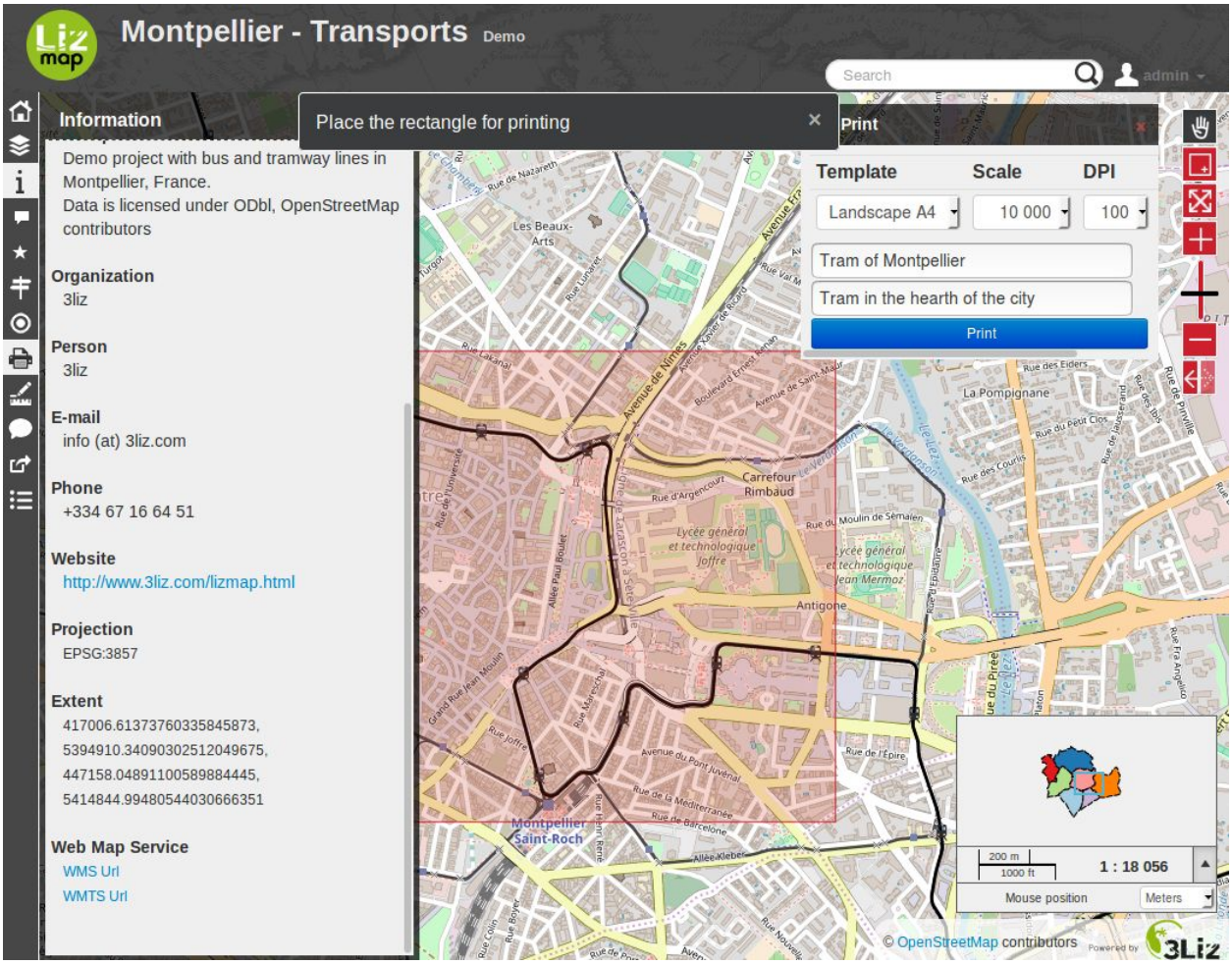
Once the feature activated, a print area with the form of the composition is displayed on the map. Use this area to define the area to be printed. You can move it.



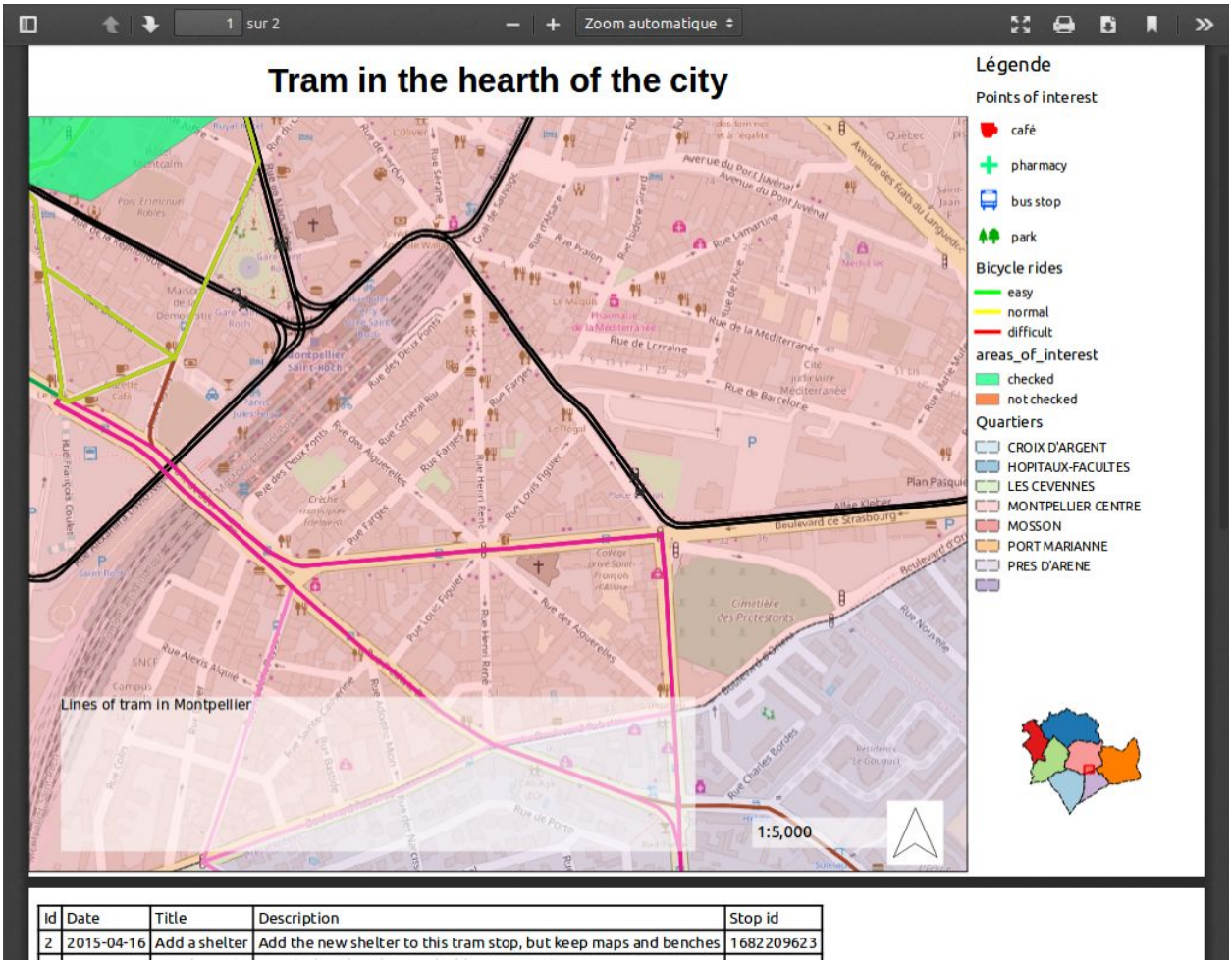
On the left, above the panel layer management, you can select the scale of printing.



Depending on the configuration of the printing composition, you can have the possibility to enter your own text.



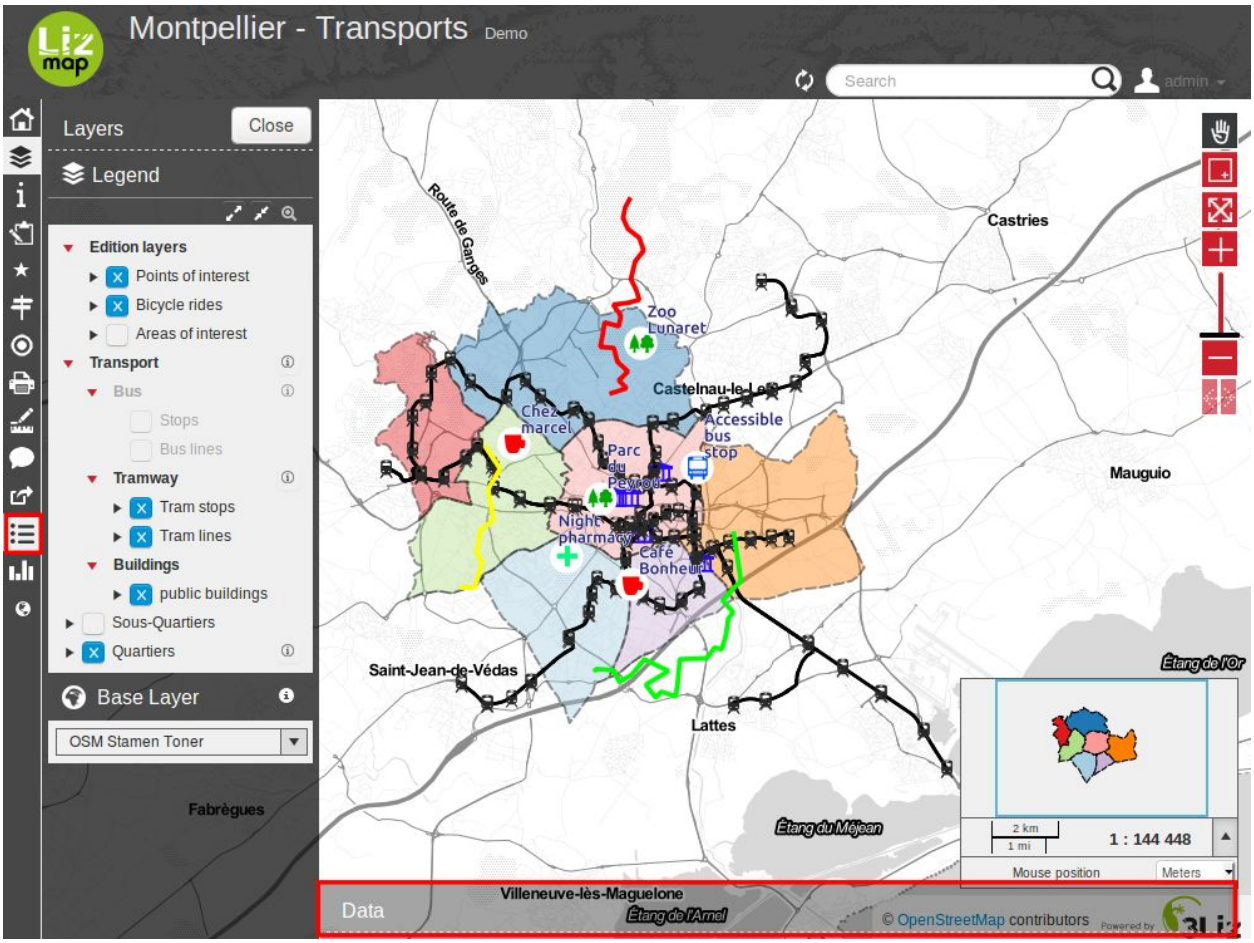
To begin building the print file, you can click on *Print*. You will get a PDF file with the layout defined by the map publisher.



2.8 Attribute layers

When this feature has been enabled by the map publisher for one or many vector layers, a new menu entry will be visible at the bottom of the menu bar, labelled as **Data**. Another entry labelled **Selection** with a star icon will also be displayed for a layer for which the attribute table has been activated.

Clicking on this icon will open a new panel situated at the bottom of the interface, containing one single **Data** tab with a list of published vector layers.



2.8.1 Attribute table panel behaviour

Since this panel is situated above the map, some default behaviours have been proposed to ease the consultation of the data in the map and in the table at the same time.

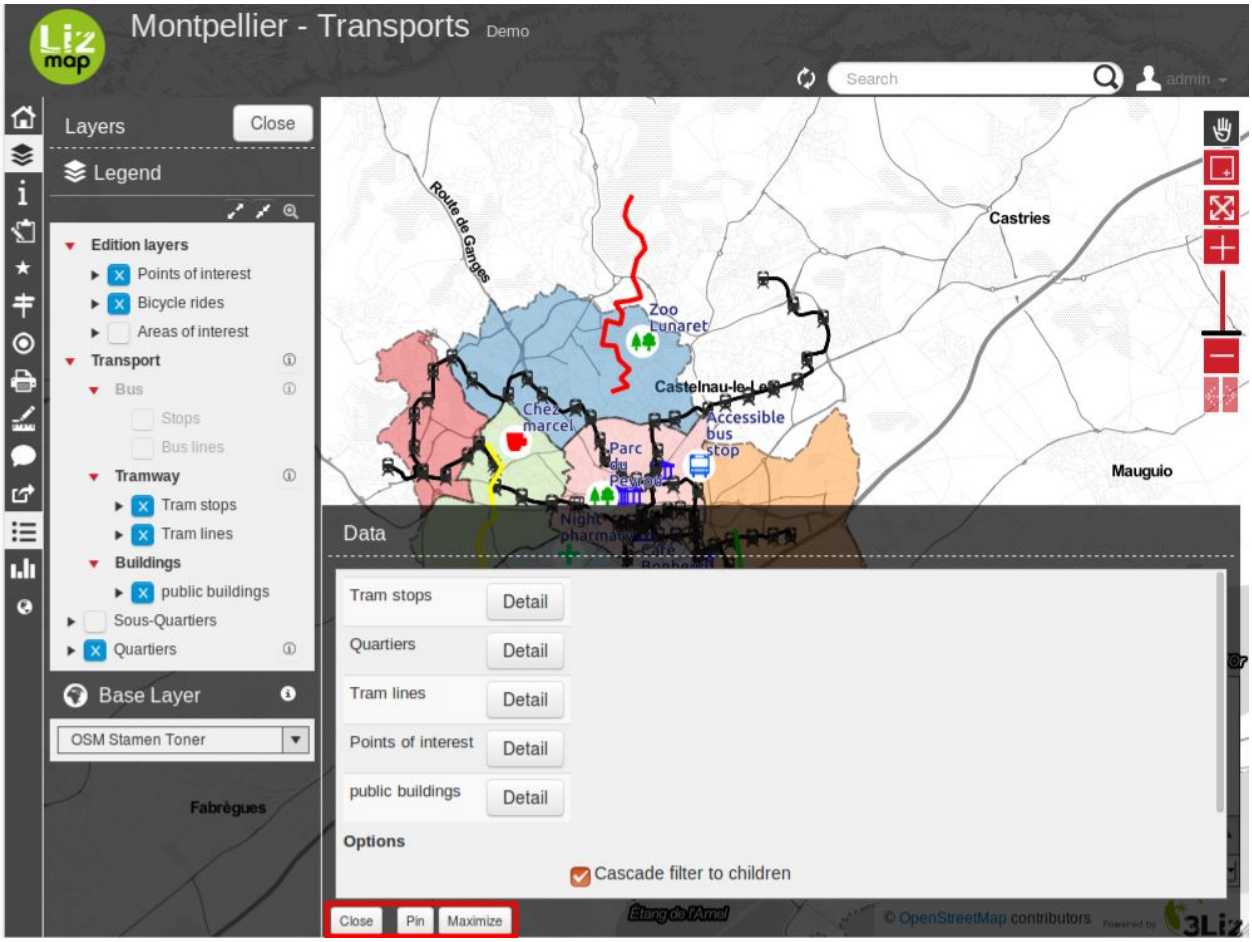
- The attribute table panel takes half the size of the screen
- It is automatically reduced at the bottom when the mouse is out of it, and automatically displayed back when the user moves the mouse over the bottom of the map.

You can change the default behaviour by using the buttons displayed at the bottom of the attribute table panel

- The **Pin** button deactivates the automatic reduction of the panel when the user mouse leaves it. But in this case, the whole panel will be displayed half transparent to allow the user to see the map underneath.
- The **Maximise** button expands the panel so that it occupies the total place left between the left panel and the map header. This is handy when you need to have a confortable view of the layer data. In this configuration, the attribute table panel is also **pinned**, and won't be reduced when the mouse leaves the panel, for example when hovering the left panel with the legend. But in this case, only the opacity of the attribute table panel will change so that you can see the map underneath.

You can click back on any of these 2 **Pin** and **Maximise** button to get back to the default behaviour.

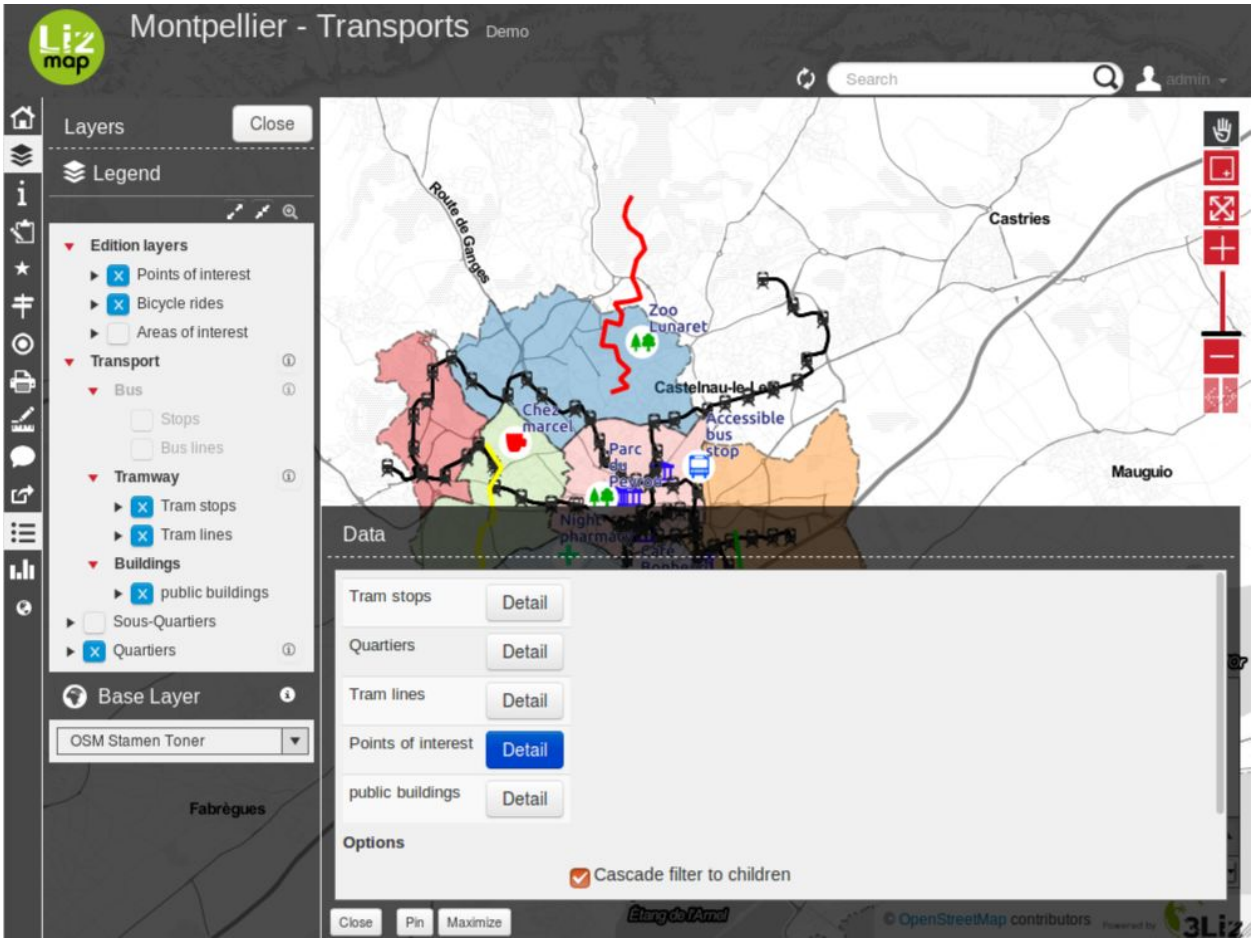
The **Close** button closes the attribute table tool completely (as if you clicked in the menu bar icon).



2.8.2 Open an attribute table for a layer

Once the attribute table panel is displayed, you can open the attribute table of each vector layer by clicking on the button situated right to the layer name. This will open a new tab labelled with the layer title which will show the attribute table for the chose layer.

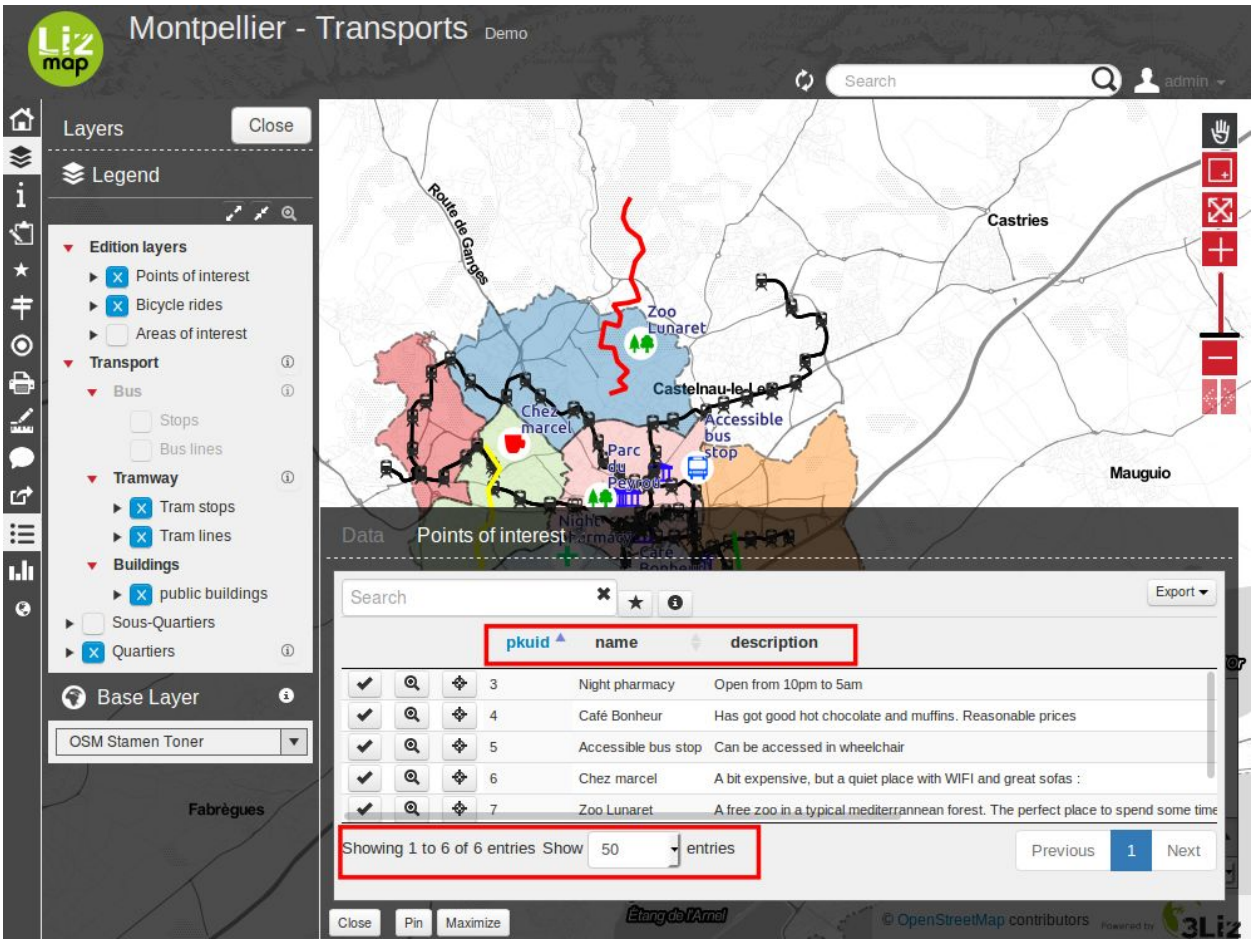
Opening an attribute table can take some time depending on the data size and complexity.



2.8.3 Description of the attribute table functionality

The layer data is shown in a **paginated table**. By default, the table shows 100 lines at a time. You can change this behaviour by using the list situated in the bottom left side of the tab content.

The table of data can be **ordered by a field** by clicking on the column corresponding to . If you click again on the column, the order will be reversed.



Actions on lines

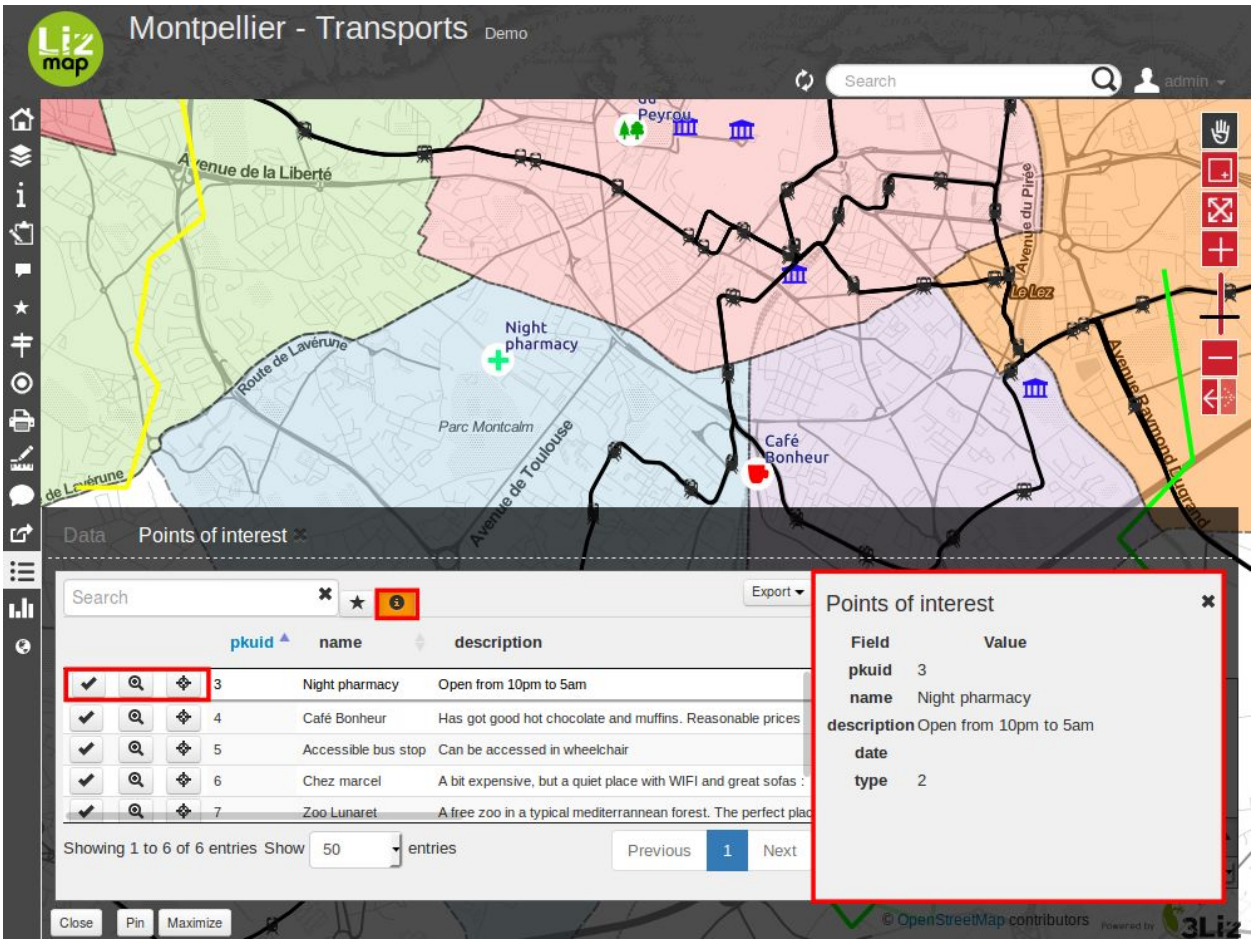
When you **click on a table line**, the line will be displayed with a border to help viewing which line is **highlighted**.

If the publisher has enabled the **popup** for the layer, you will be able to see the content of the popup for the highlighted feature by clicking on the (i) button (labelled with *Display info* when hovering the mouse on it). This will open a panel right to the table, which will display the detailed information on the highlighted table line. Furthermore:

- Clicking on another line will refresh the right panel content with the new highlighted feature.
- Clicking back on the (i) button or on the cross situated at the top right side of the information panel will hide the panel. To have the (i) button, you need to have activated the popup for the current layer.

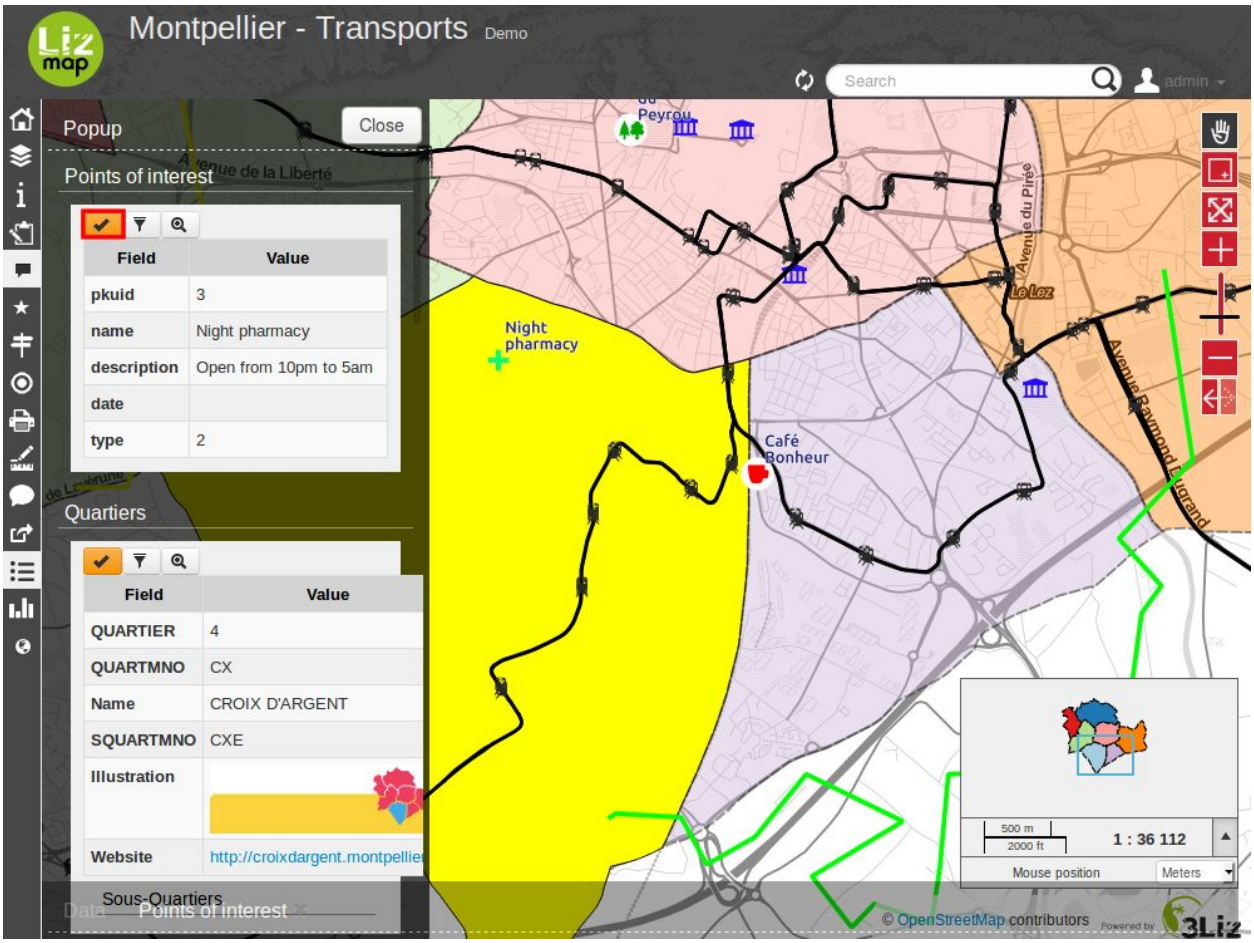
Each line of the table shows some buttons at the left side:

- **Select** button : when clicked, the corresponding feature is selected. You can select many lines by using this button on different lines. The selected features will be displayed in a different style on the map, usually with a Yellow color. You can click again on the "Select" to unselect a selected feature.
- **Zoom** button : clicking on this button will zoom to the corresponding geometry in the map. The scale will be chosen so that the feature geometry uses most of the available space.
- **Center** button : clicking on this button will just pan the map to the corresponding feature, without changing scale.



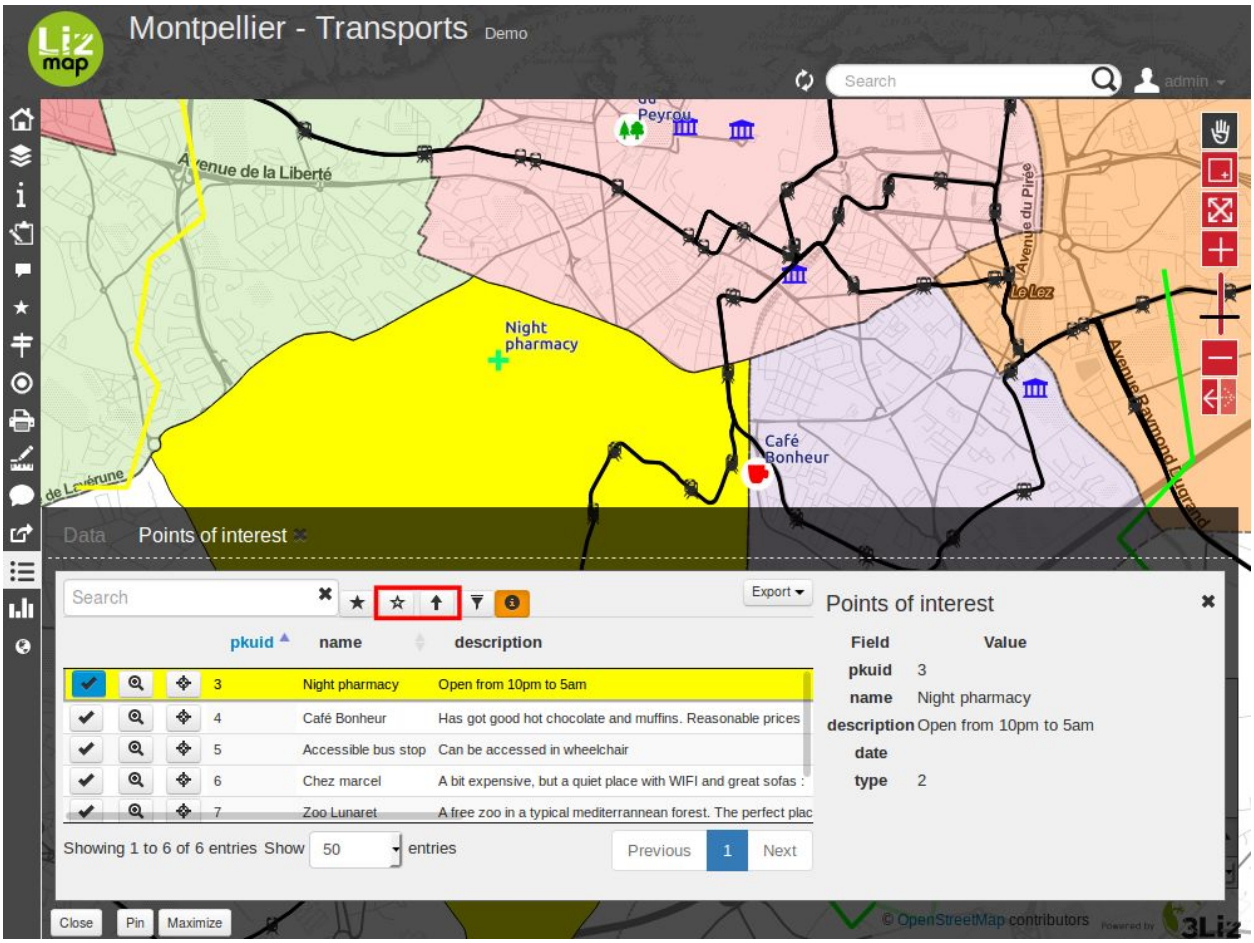
More about selection tools

You can also select a layer object by **displaying the popup** for this object (if the map publisher has enabled the popup for the layer, a click on the map will show a popup window containing detailed information on the clicked feature). Inside the popup, if you can see the select button, you can use it to select only this object. Previous selection will be replaced by only this object.



When one or more lines have been selected in the attribute table (they become yellow), you can use the black “arrow up” button situated above the table to **move the selected lines at the top of the table**.

You can **unselect all the selected objects** by clicking on the “white star” button situated above the table.



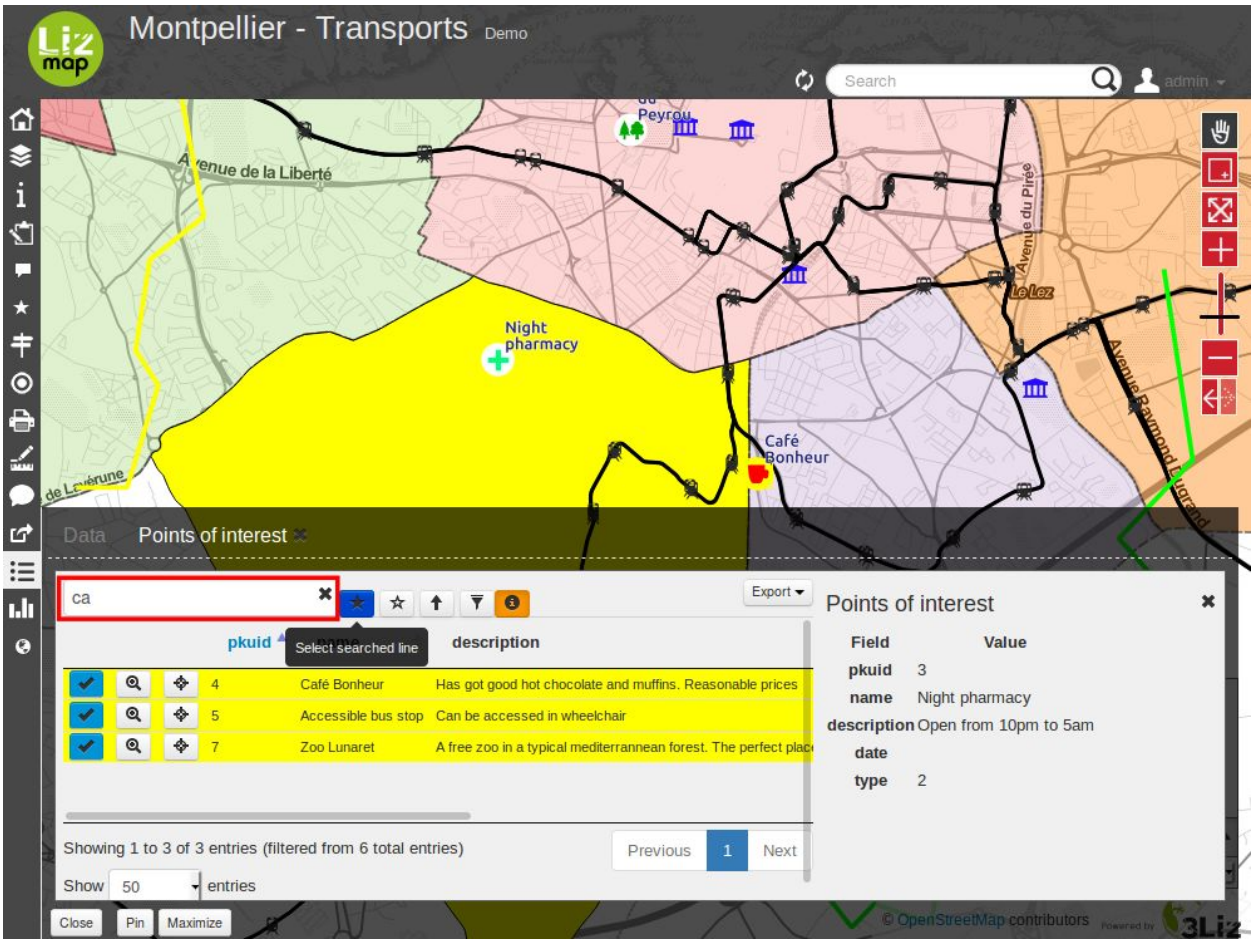
Quickly search through data

You can **filter the lines displayed** in the table by entering some letters in the **Search field** situated at the top left side of the tab content. If you want to see all the features again, just erase the search field content manually or by clicking on the cross button situated inside the field.

The text entered in the search field launches a search among the data for **all the fields of the table.**

Note that only the content of the table will be restricted to the lines matching your textual search. **The objects on the map will not be filtered dynamically** (but you could use select and filter to do so, see below)

Once you have filtered some data in the attribute table by entering some text in it, **you can easily select them all** by clicking on the "black star" button labelled "Select searched lines". This will select all the corresponding objects in the attribute table (display them in yellow) and also change their color in the map (usually in yellow too, depending on the configuration done by the map publisher)



Filter data

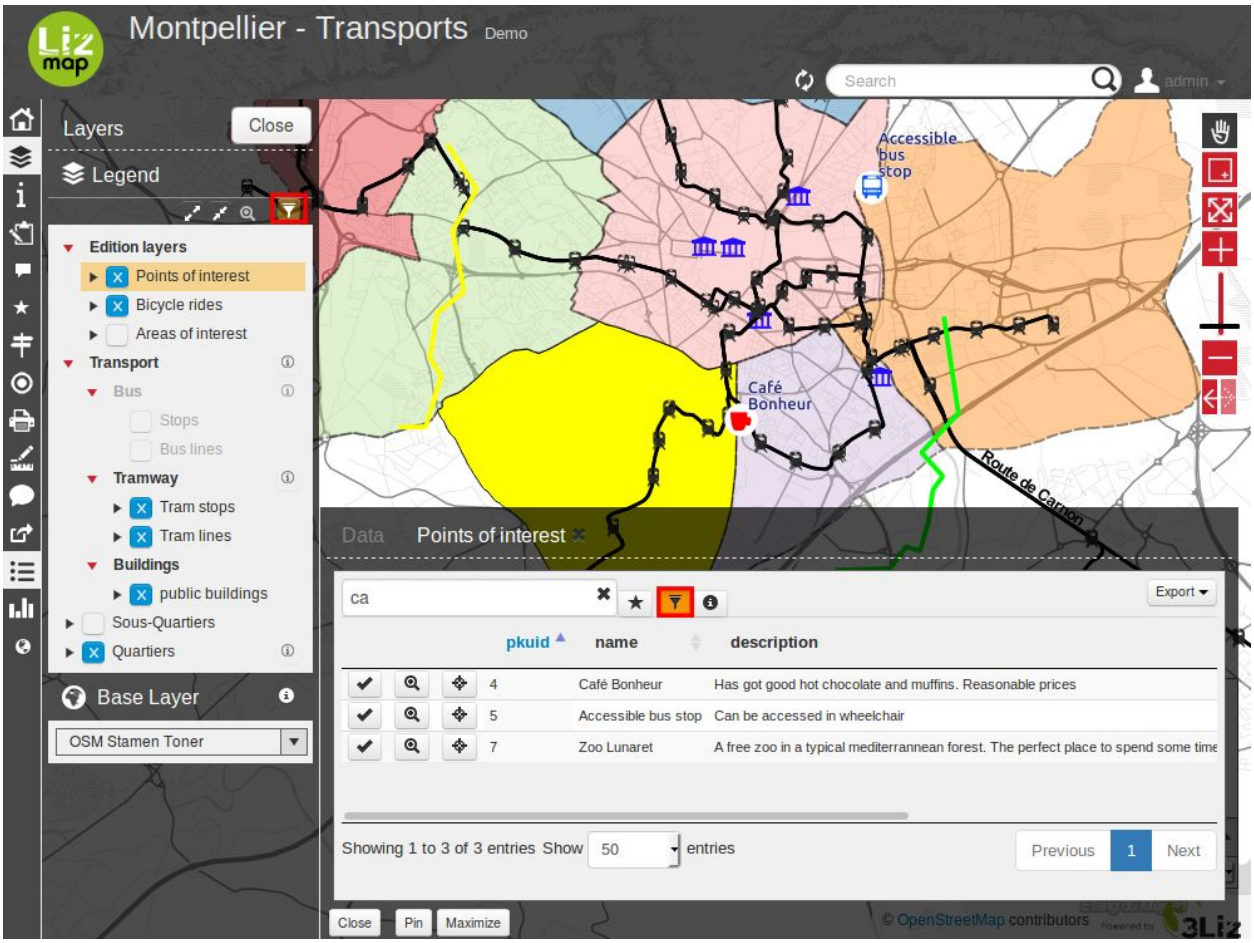
When you have selected one or more objects in the layer attribute table, you can then **filter the data displayed in Lizmap** for this layer. To do so, just click on the "Funnel" button labelled "Filter" situated above the table (only available if some the selection contains at least one object).

Filtering will have the following consequences:

- The attribute table will **show only the filtered data**
- The Search input field will allow to **search only among filtered data**
- The map will show **only the filtered objects**
- The child layers linked with relations (and also published in the attribute table tool) will be filtered too. We call it "cascading filtering". For example, the bus stops could be filtered automatically if you have filtered one bus line, to show only the ones served by the filtered line.
- The filtered layers will be marked in the left panel legend with an orange background, and a new orange "Funnel" button will be displayed above the legend.

You can cancel the filter to go back to previous state:

- by clicking on the orange "Funnel" button at the top of the legend in Lizmap left panel
- by clicking back on the filter button just above the attribute table concerned by the filter



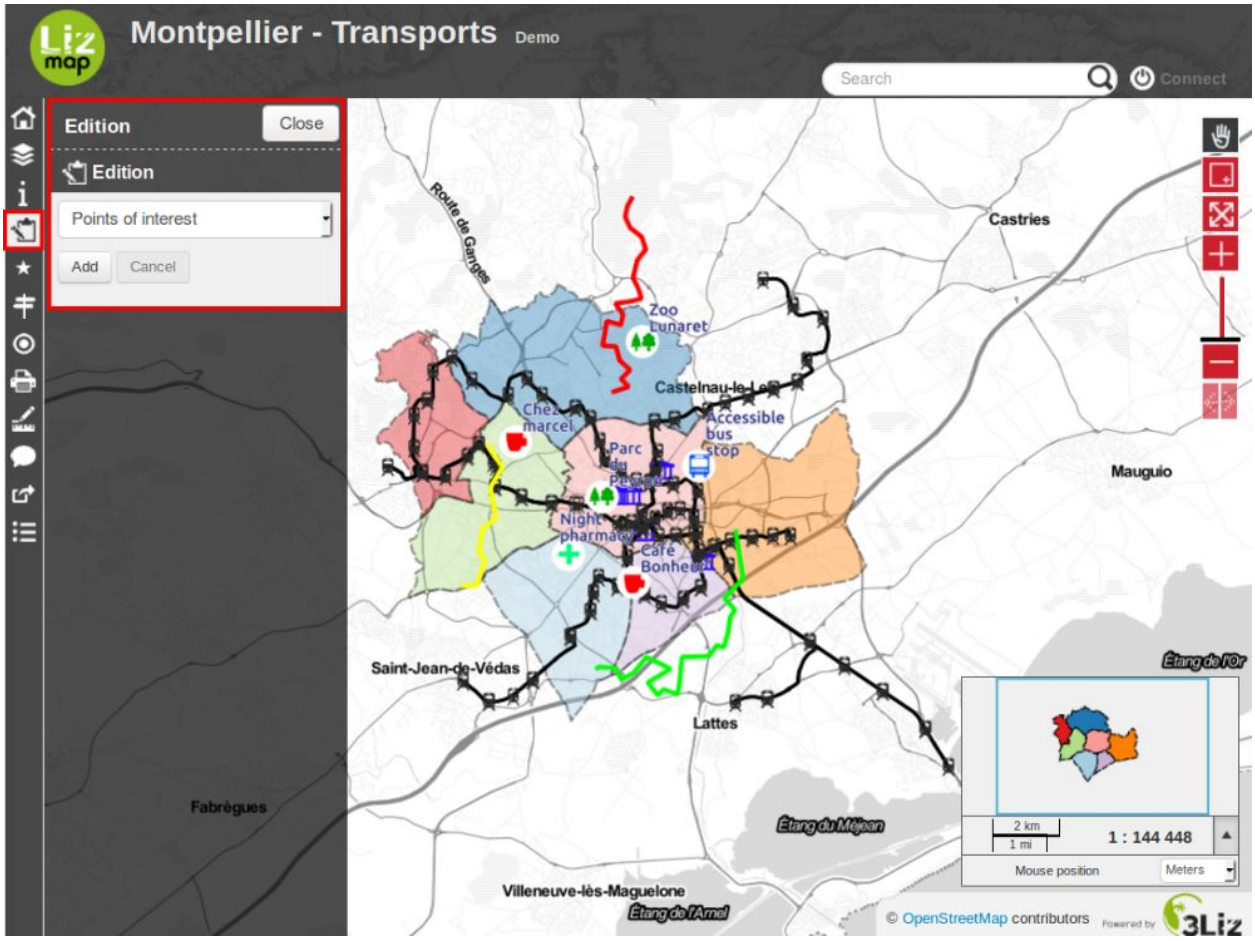
When exporting the map view with the permalink tool (situated in the menu bar), **the filter will be activated** in the linked map and the users won't be able to easily unfilter the data : the unfilter button will not be displayed in Lizmap interface The only way would be to remove the filter parameters from the permalink URL. **This is not a safe way to protect some data, but a way to focus on some data only**.

2.9 Editing spatial data

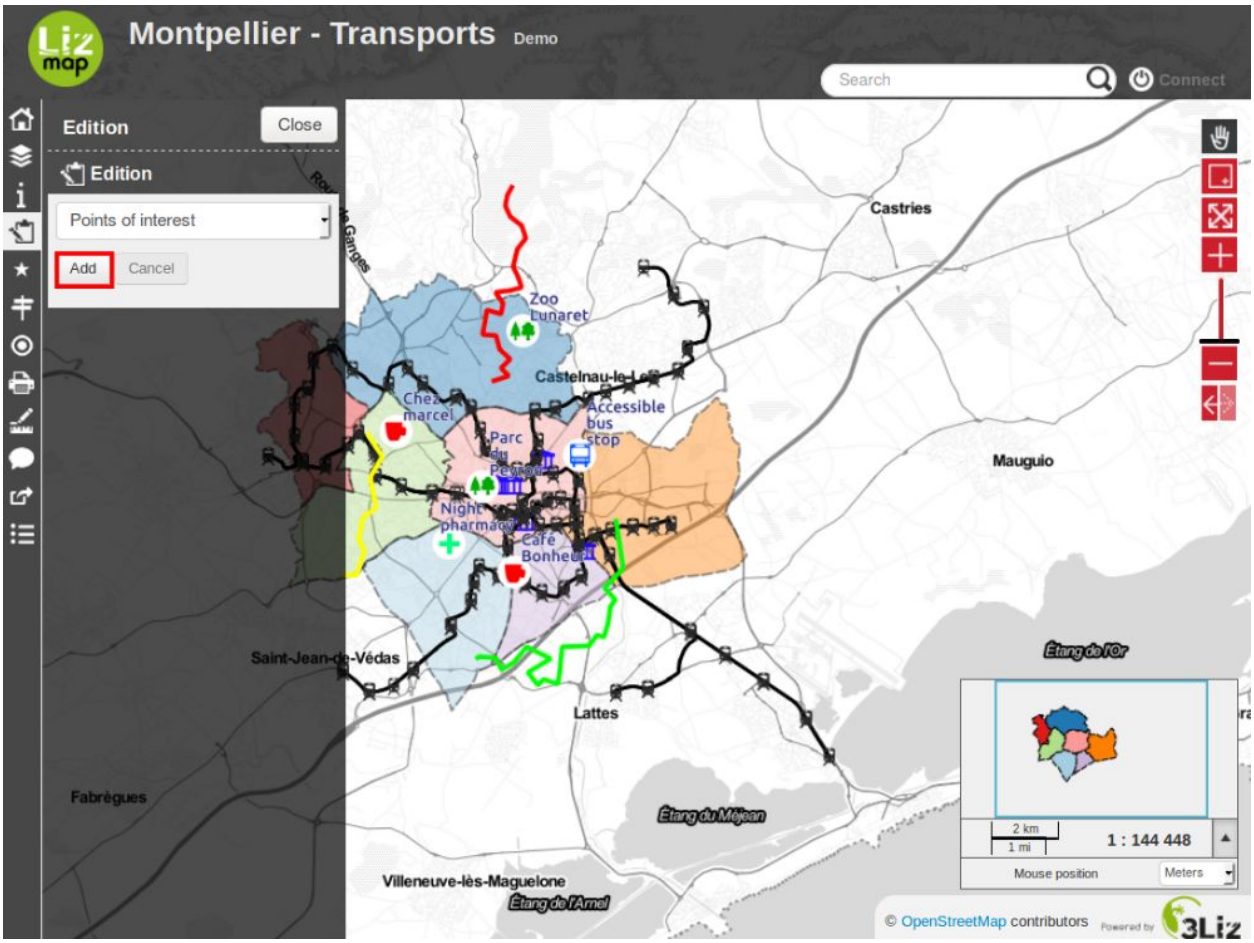
The map publisher can allow users to edit certain data. It also has the ability to limit possible changes:

- adding spatial object
- geometric modification
- fields modification
- deleting spatial object

The feature is available in the Lizmap menu bar. The edit menu allows you to select the data you want to update.



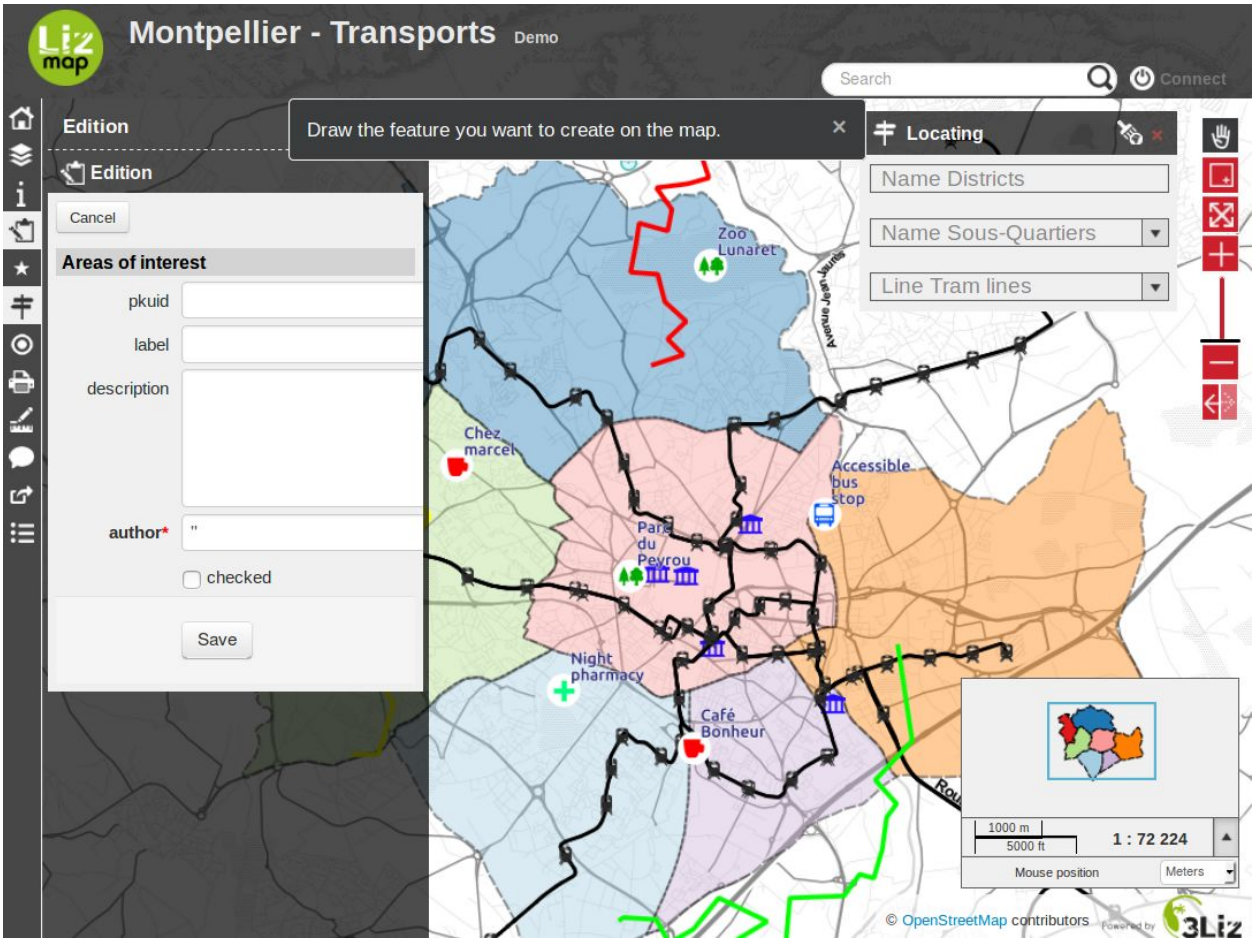
Once the layer selected, the edit pannel appears. This varies depending on the configuration desired by the map publisher. If any changes are available you have to choose between *Add* a new object or *Select* one.



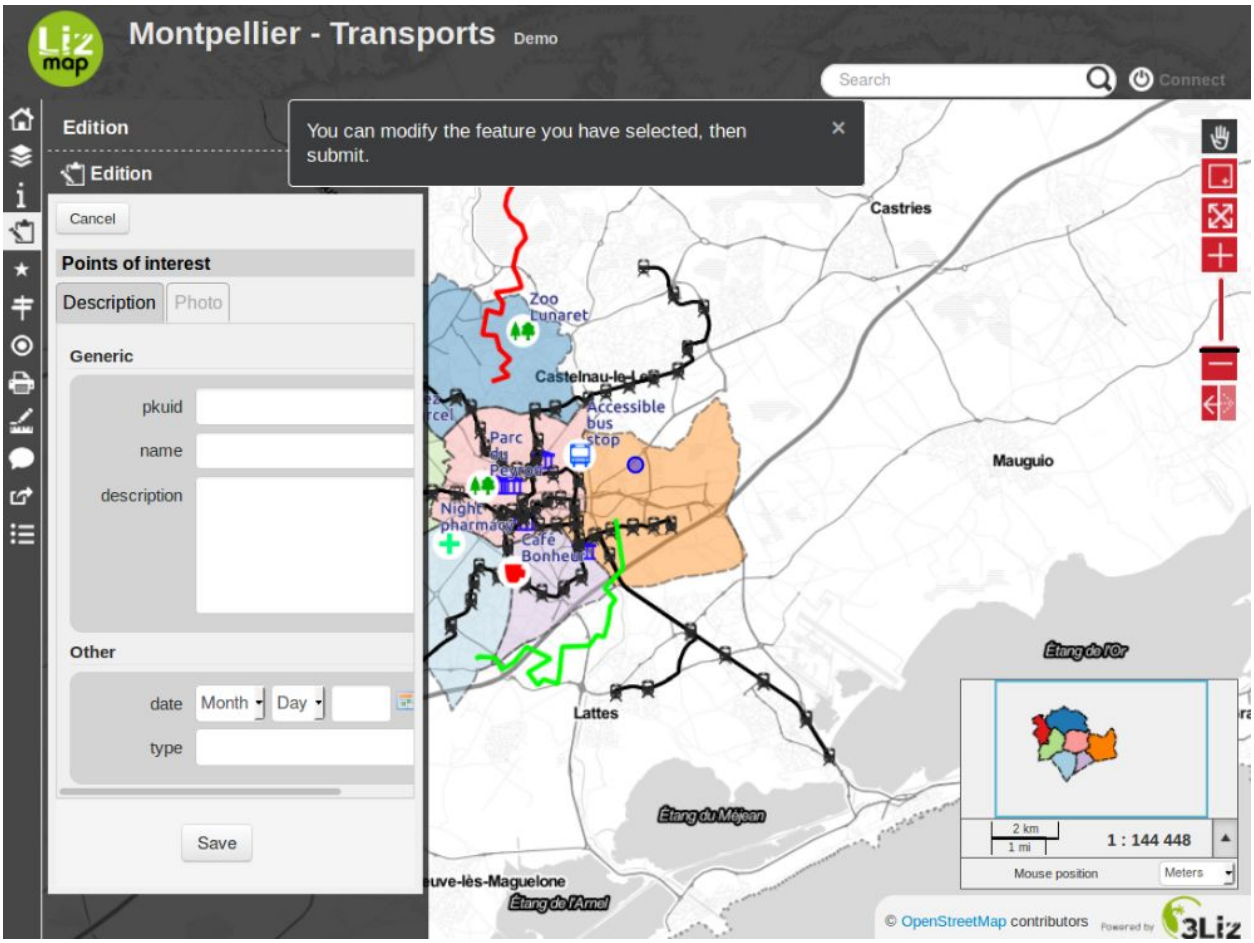
If you have selected *Add*, you will be asked to draw a simple form that depends on the selected data layer:

- point
- line
- polygon

In the case of line and polygon, you need to click several times to draw the shape you want.



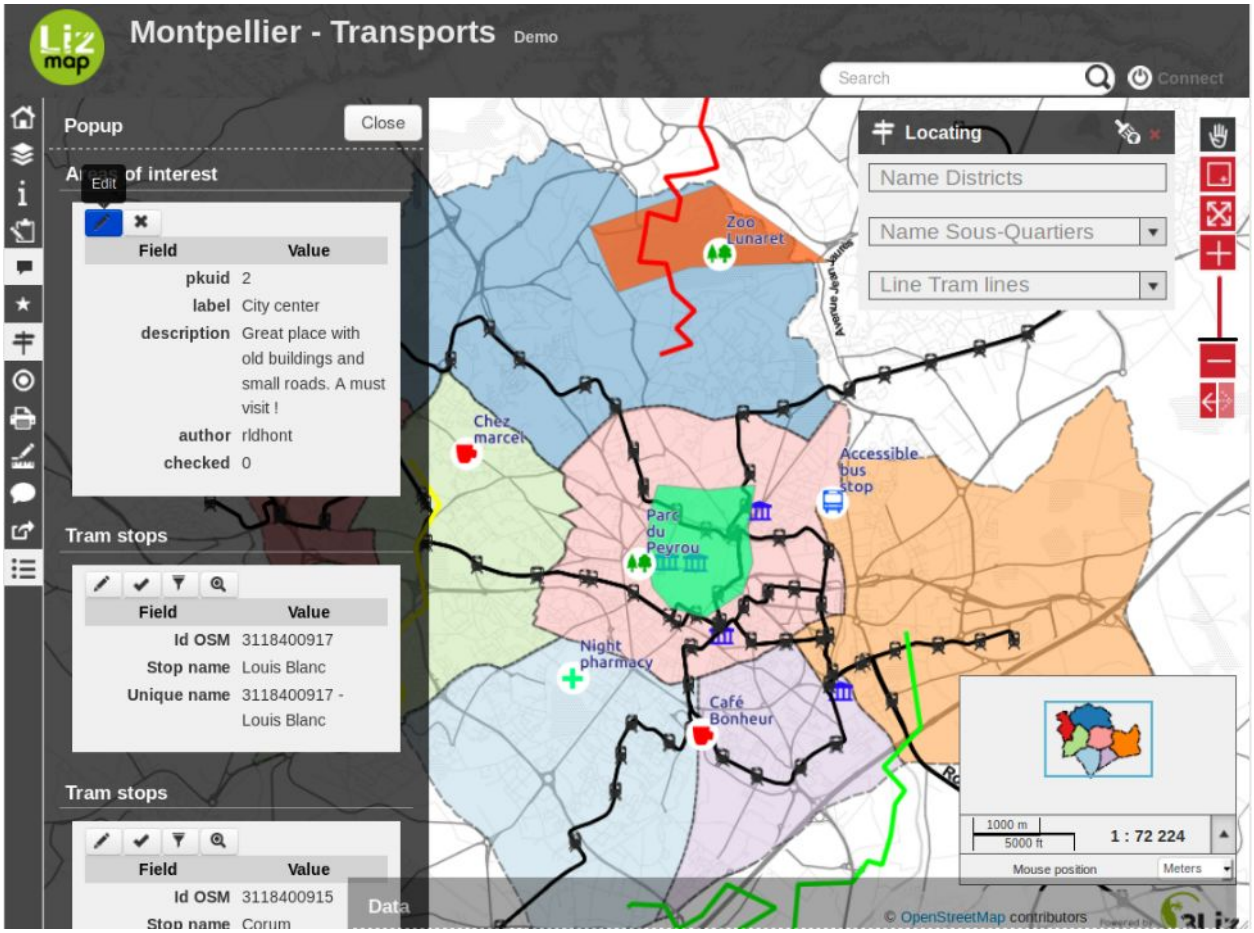
To finish your line or your polygon you must add the last point by double-clicking the desired location. Once drawing finished, an editing form for fields will be displayed.



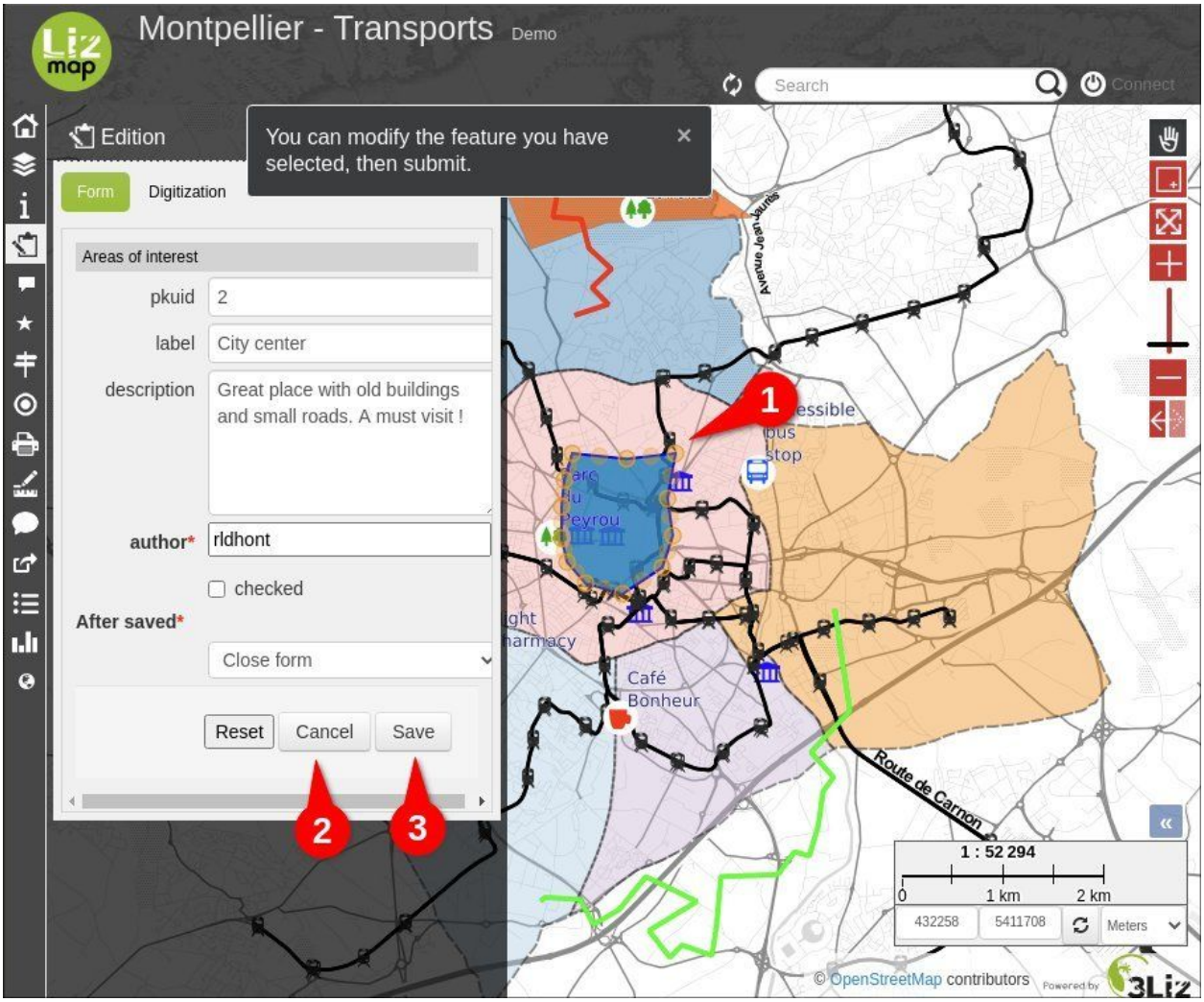
If you want to restart drawing the geometry, you should click *Cancel*.

If the geometry is right for you and you have entered the required information, you can *Save*. The new object will be added. You will be able to update it by selecting it.

To select an object to update, you will need to click on it on the map then click on the button *Edit*.



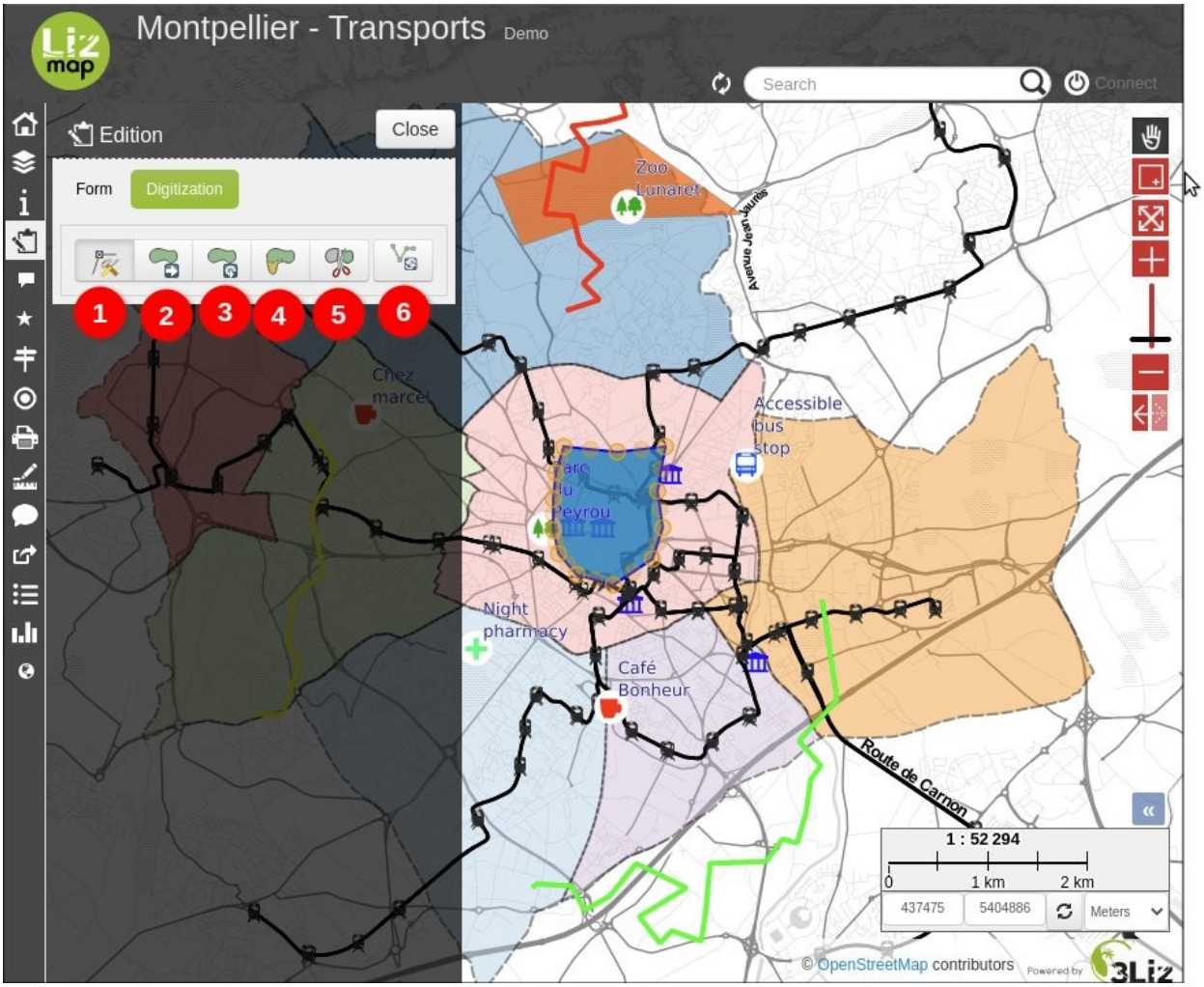
1. The selected object appears on the map and its geometry may be changed immediatly.
2. You can undo geometry changes and stop edition using the *Cancel* button.
3. To validate your geometry modifications, you must click *Save*.



If you want to remove a point on a geometry, you must hover it and type *Del* on your keyboard.

Digitizing button is displayed with line and polygon geometries, by clicking it, you'll access those tools:

1. Node tool (default). Move or delete the geom nodes (vertices) and also create new ones by dragging virtual nodes in the middle of the segments.
2. Drag tool. Translate a geometry by dragging the displayed point at the center of the geometry.
3. Rotate tool. Rotate a geometry by dragging the displayed point at bottom right of the geometry.
4. Reshape tool.
5. Split tool.
6. Reverse geom (To be used with the node tool). Reverse nodes (vertices) order. Useful for streets when their circulation orientation is reversed for example.


Montpellier - Transports
Demo
Search
Connect
Edition
Close
Form
Digitization
1
2
3
4
5
6
Zoo
Lunaret
Avenue Jean-Lanes
Chez
marcel
Accessible
bus
stop
Parc
du
Peyrou
Night
pharmacy
Café
Bonheur
Route de Carnon
1 : 52 294
0
1 km
2 km
437475
5404886
Meters
© OpenStreetMap contributors
Powered by
3Liz

CHAPTER

THREE

PUBLISHER GUIDE

This guide is for the GIS technician who want to publish some data in Lizmap web client.

3.1 Quick start

This is quick start guide to help you how to publish your first dataset on Lizmap with a few knowledge.

3.1.1 Prepare a QGIS project for Web

Contents

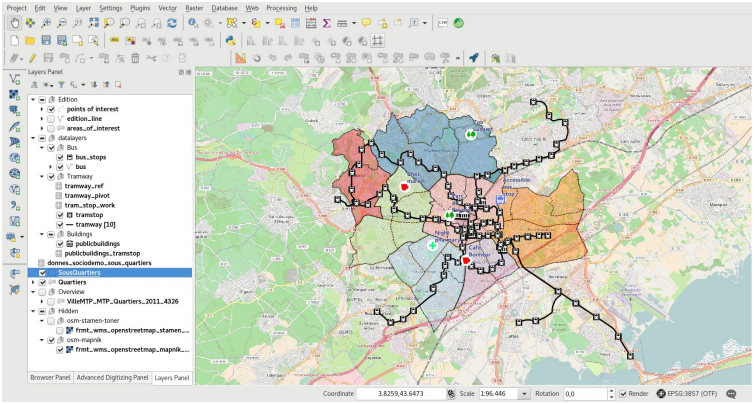
- *Prepare a QGIS project for Web*
  - *Create your project*
  - *Set up your project for Web*
  - *Configure your layers for the Web*
  - *Save your QGIS project*

Create your project

Add your data:

- Vector geographic data files
  - ESRI Shapefile
  - MapInfo TAB and MIF/MID
  - GeoJSON
  - etc
- RASTER geographic data files
  - GeoTIFF
  - Arc/Info ASCII Grid
  - netCDF
  - etc

- Geographic data base
  - PostgreSQL / PostGIS
    - When you create your connection to PostGIS, use the checkbox *Use estimated metadata*. Be careful, changing this checkbox after you have already added a layer doesn't change layers already loaded.
  - MSSQL spatial
  - Oracle locator / spatial



Organize and manipulate the layers in the legend:

- *Add groups* with a right click in the empty part of the legend: *Add a new group*
- *Move* layers and groups with *drag-and-drop*
- *Rename* layers and groups with the `F2` or the layer properties window
- Manipulate the rendering order:
  - with the *legend layer order*: the upper layers are rendered above the others.
  - by specifying *layer order* with the menu *View* ▸ *Panels* ▸ *Layer order*

Add a title to your project and save it in your working directory.

Note: If your layer has more than one style, the user will be able to switch between them through the button *Change layer style* at the top of the legend.

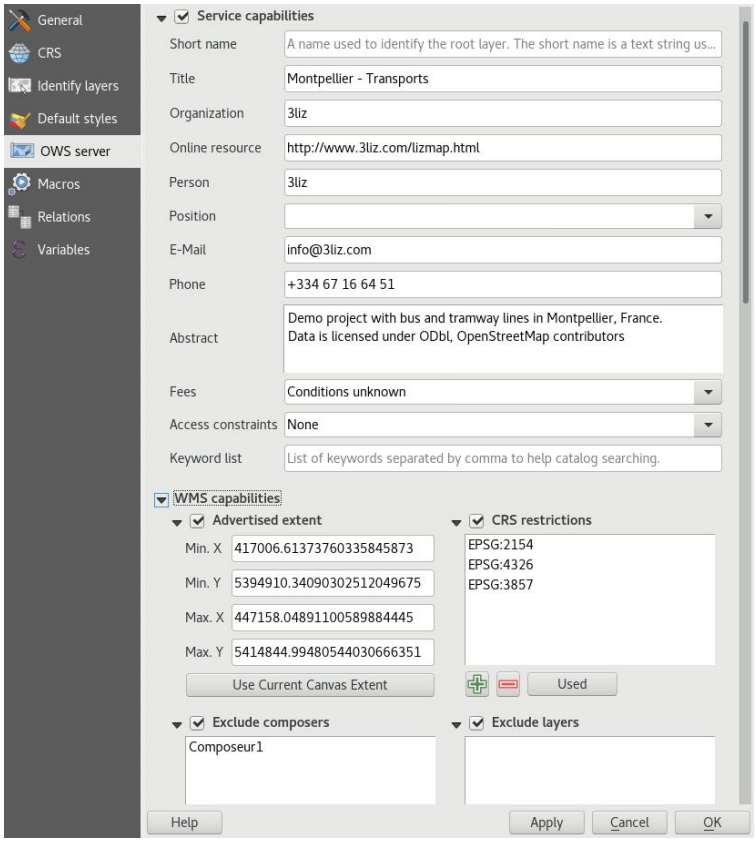
Set up your project for Web

Configure the coordinates reference system, CRS, of your project:

- Select the CRS of your Web map:
  - EPSG:3857 for Google Mercator
  - EPSG:2154 for Lambert 93
  - etc
- QGIS can reproject raster and vector data.

Configure the Web Geographics Services parameters with the *QGIS Server* tab:

- Set the title of your Web Geographics Services. This title is used in the Lizmap landing page.
- Add informations like your organization, the owner of the publication, the abstract, etc
- Set the maximum extent of your WMS service
- Restrict the CRSs list of your WMS service:
  - at least select the one used in your project
  - you can use the button *Used* to get all the layer CRS and the project one
- Exclude compositions and layers if data cannot be published in WMS
- Enable the layers you want to publish WFS and WCS

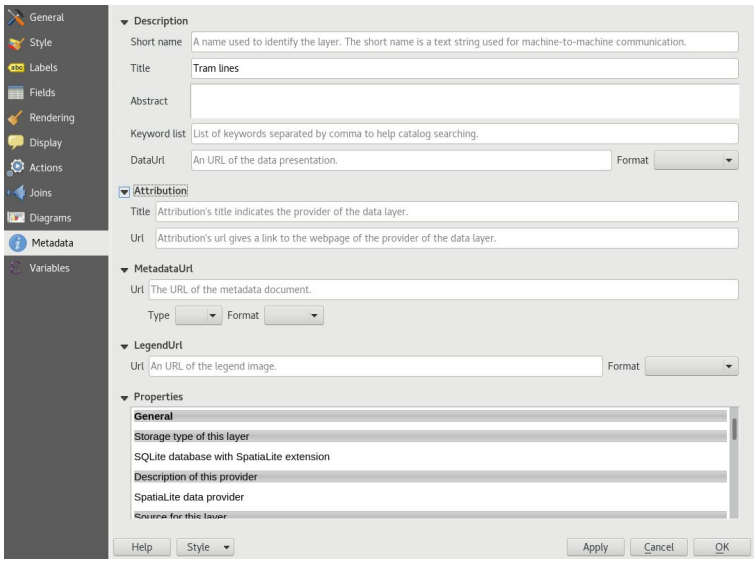


Check that the paths are saved *relative* in the general tab of the project properties window, access it with the menu *Project* ‣ *Project Properties* or `CTRL+SHIFT+P`.

Configure your layers for the Web

In the window *Layer properties ▸ QGIS Server* allows you to configure a lot of information for Web Geographic Services:

- Provide a title, a description and keywords
- Specify the attribution to respect the data license
- Add the metadata record URL if it's available



In the window *Rendering* tab, enable the *Simplify geometry* and the *Simplify on the provider side if possible* checkbox too. Note this can be changed in your global settings for layers added later. Read *Server side simplification*.

Save your QGIS project

Warning: In **QGIS 3**, it's possible to save your project in `QGZ` format or in database (PostGIS/Geopackage). Lizmap does not support these formats. You must save your project as `QGS` extension by doing *Project ▸ Save as… ▸ QGIS file* and not choosing `QGZ`.

You should save your project as `QGS` file on your filesystem before opening the Lizmap plugin in the next step.

3.1.2 Set up your project for Lizmap

Contents

- *Set up your project for Lizmap*
  - *Install the Lizmap plugin*

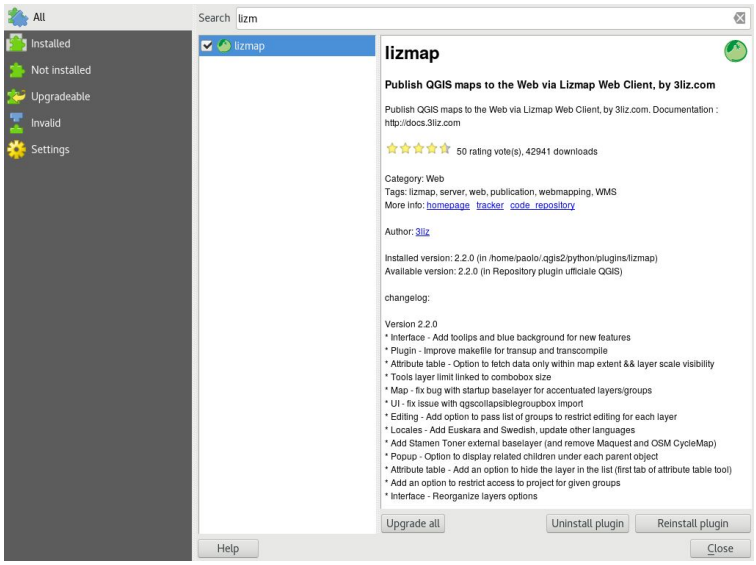
Install the Lizmap plugin

The Lizmap plugin is available through the official QGIS project repository: https://plugins.qgis.org/plugins/lizmap/

To install it, just do it like any QGIS plugin:

- *Menu ‣ Plugins ‣ Manage and Install Plugins*
- Search `Lizmap`
- Install the plugin
- It's available in the menu and toolbar *Web*

Warning: Lizmap QGIS plugin is regularly updated. To enjoy new features or before reporting a bug, be sure to update your plugin. In *Menu ‣ Plugins ‣ Manage and Install Plugins ‣ Settings*, check that QGIS will check for updates on a regular basis automatically.



The plugin is organized in 13 tabs:

- *Information*: some information about the Lizmap project and your servers
- *Map options*: the general options of the map
- *Layers*: the options of each layer
- *Baselayers*: the baselayers used on the Web
- *Locate by layer*: the locating tool
- *Attribute table*: configure the attribute table and the vector selection
- *Layer editing*: which layer can be edited with Lizmap
- *Tooltip layers*: configure the tooltip for some layers
- *Form filter*: Make some filtering based on attributes
- *Filter layer by user*: setup some filtering based on the current logged user
- *Dataviz*: add some charts and dataviz
- *Time manager*: play an animation based on date or datetime field

- *Atlas*: setup an atlas for the project
- *Log*: displays information of performed actions

And it has 5 action buttons:

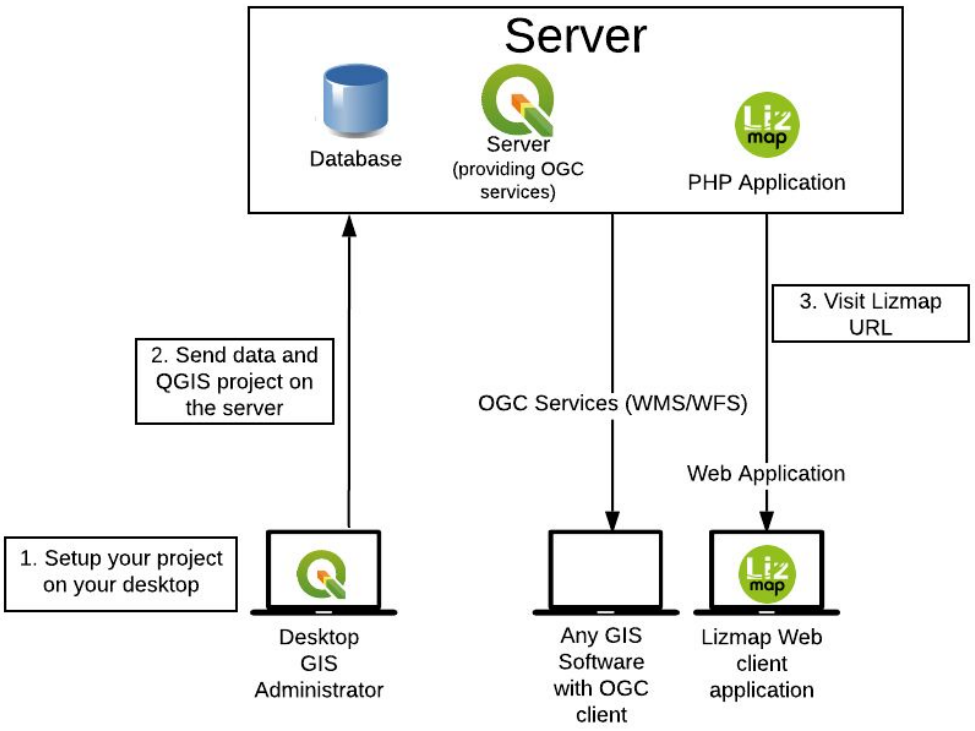
- *Auto-save* : When ever you click on *Apply* or *Ok* if Lizmap should save the QGIS project too at the same time
- *Help* : open the help in the webbrowser
- *Apply* : write the configuration in the Lizmap file and keep the dialog open
- *Ok* : write the configuration in the Lizmap file and close the dialog
- *Close* : close the dialog without writing the configuration

3.1.3 Publish the map by FTP

Contents

- *Publish the map by FTP*
  - *Reminder of Lizmap architecture*
  - *Use an FTP client*

Reminder of Lizmap architecture



Lizmap is based on repositories system. To publish a map in Lizmap, it is sufficient to ensure that the contents of the local directory containing the data and QGIS projects **be reproduced exactly** identical in the corresponding server repository.

For this, it is necessary **to synchronize the local directory with that of the server** each time you update the QGIS project, modify the Lizmap configuration with the plugin, or add files in the local directory.

Note: If you are working locally, as Lizmap Web Client is installed on the same machine you use for QGIS, you do not need to *synchronize* your files with FTP. This configuration should only exist for testing.

Note: You can use any tool and synchronization protocol (FTP, FTPS, SFTP, rsync, unison, etc), if you can master the tool and have access to the Lizmap server configuration.

Use an FTP client

FTP allows you to access files from a server, retrieve it and add documents and/or folders. It can therefore be used to synchronize your local directory with the server one Where Lizmap Web Client is on. This protocol is a Web standard that can be exploited through many FTP clients.

You can use the following client or one you usually use:

- *FireFTP*: Firefox add-on
- *Filezilla*: Free cross-platform software (Windows, MacOS, Linux)
- *WinSCP*: Free software for Windows

You can use these tools to make manual changes to the remote directory:

- **made a backup**
- **remove contents**
- **overwrite files manually**: QGIS project `*.qgs` and Lizmap configuration `*.qgs.cfg`.

3.2 Lizmap QGIS plugin

This section is listing panels available in the Lizmap QGIS plugin.

3.2.1 Information - Get some news about the project and your server

Contents

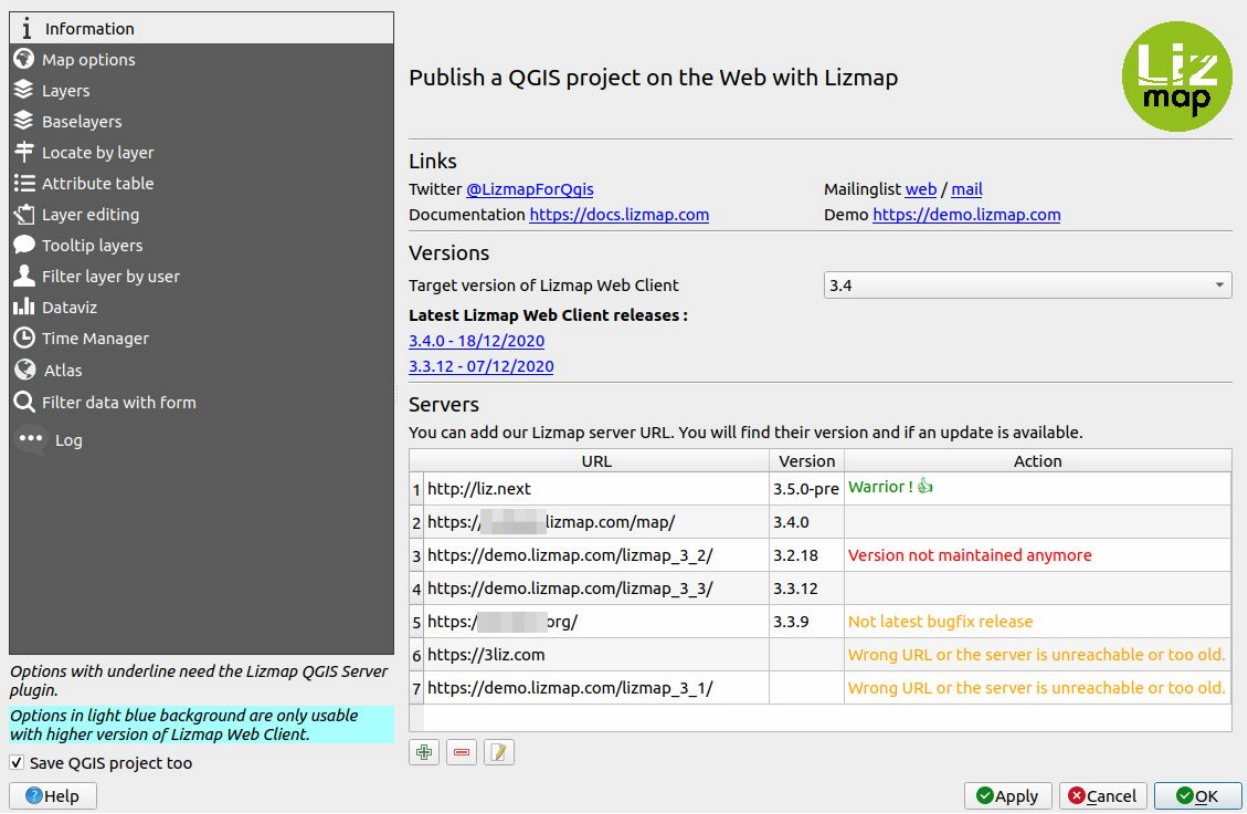
- *Information - Get some news about the project and your server*
  - *Principle*
  - *Configuring the tool*

Principle

In this panel, we can :

- find some links about Lizmap social networks
- be aware of a new release regarding Lizmap Web Client. Only two branches are maintained simultaneously. If you branch is not listed, it's not maintained anymore, except if you are using the *master* branch.
- set the target Lizmap Web Client version. You will be aware which features are available or not on your server.
- check your Lizmap Web Client server's version

Configuring the tool



- For adding a new server :
  1. Click
  2. Write the URL of the Lizmap Web Client server. The URL should be the main Lizmap landing page.

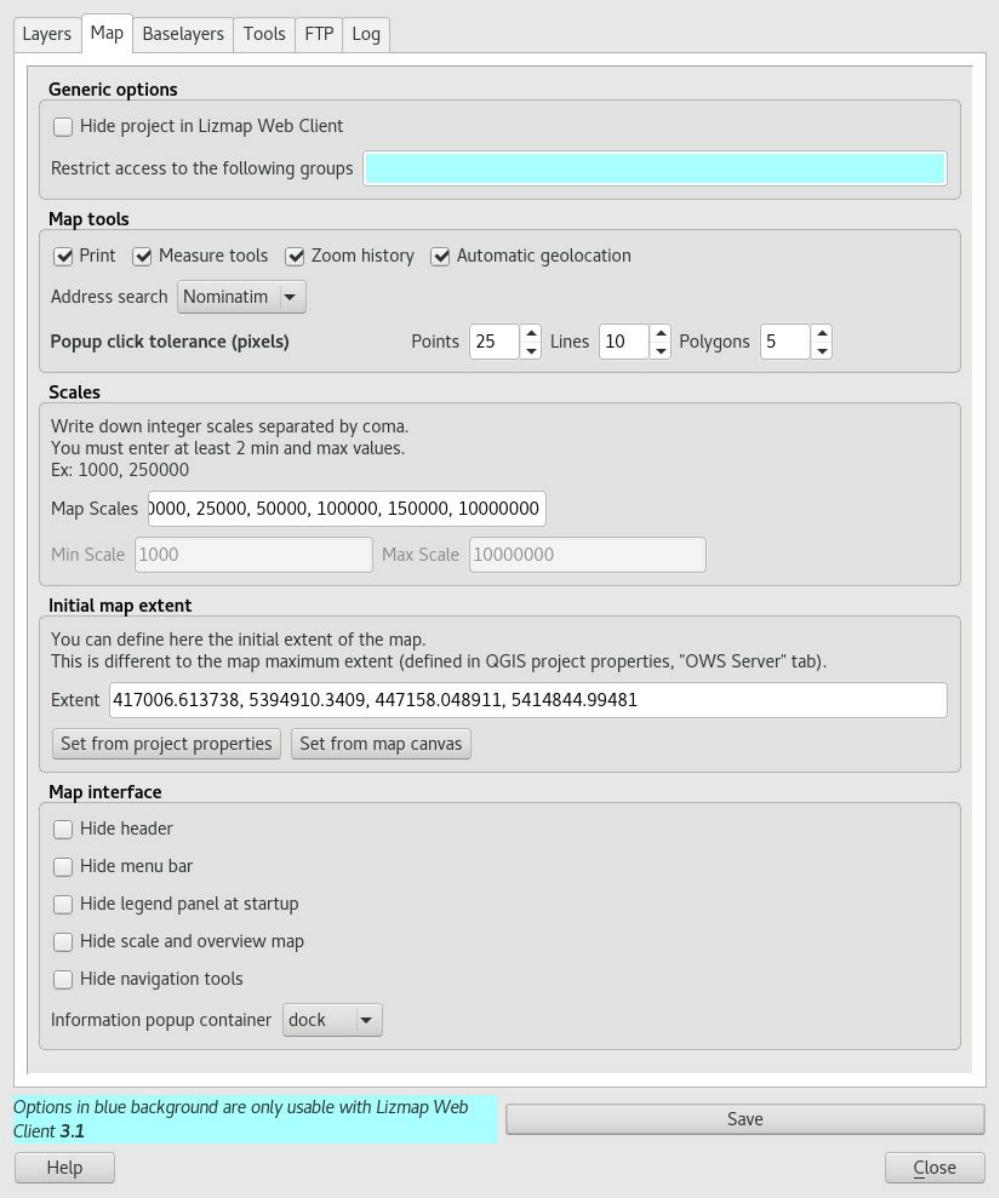
Warning: Do not use the any URL redirection. For instance, https://demo.lizmap.com/ is a redirection to https://demo.lizmap.com/lizmap/. Only the second one will work.

3.2.2 Map options

Contents

- *Map options*

The tab *Map options* allows you to enable or disable basic Lizmap tools, choosing scales and the initial extent.



The generic options:

- hide the project in Lizmap Web Client:
  - if this option is checked, the project will be hidden in the Lizmap home page that shows thumbnails for all directories and project of the application. You can use this option to hide the project
  - the project will still be accessible for WMS or WFS clients based on directories rights

  - this feature is interesting in the case of using this project as an external project for other ones.

The map tools:

- *Draw*: **Lizmap 3.4**, enables some drawing tools.
- *Print*: enables the use of QGIS compositions for PDF generation map
- *Measure tools*: enables the measurement tools in the map (length, area, perimeter)
- *Zoom history*: enables the navigation buttons in the history of zoom and move in the map
- *Automatic geolocation*: enables the functions to use the HTML5 geolocation based on Wifi and/or GPS
- *Address search*: to add an address search engine that is based on one of these services:
  - Nominatim (OpenStreetMap)
  - Google
  - IGN France (Géoportail)
  - BAN (France)

The scales :

- a list of integer values separated by commas (and optional whitespace), eg: `250000, 100000, 50000`.
- Lizmap also used these scales to restrict the application display between the minimum and maximum of these scales. This is why **it is mandatory to enter at least 2 scales** in the list.

Warning: As soon as there is an external basemap published in **EPSG:3857** such as OSM, Google Maps…, the application will only use the minimum and maximum **scale** to the minimum and maximum **zoom level**. Intermediate **scale** that you might have defined won't be read, because there isn't a strict match between your **scale** and **zoom level** provided by external base map in **EPSG:3857**.

The initial map extent:

- a list of coordinates in the Reference Coordinate System map in the format: `xmin, ymin, xmax, ymax`, setting the initial map extent
- the maximal map extent is specified in the *OWS server* tab of *Project Properties* window. The data will not be displayed if they are outside it
- by default, the initial extent is the maximal one.

3.2.3 Layers - settings for each layer

Contents

- *Layers - settings for each layer*
  - *Principle*
  - *Configuring the tool*

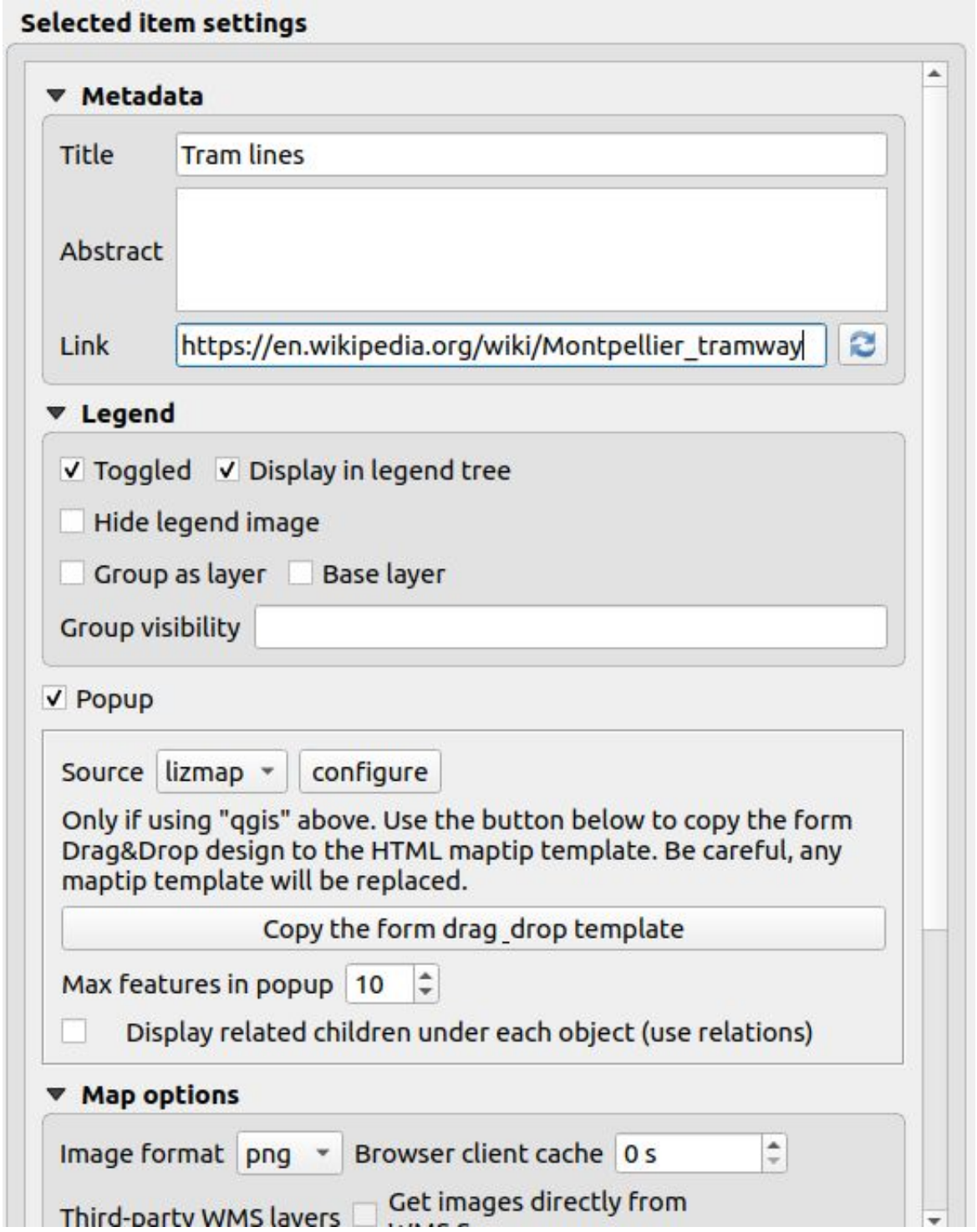
Principle

To set parameters for each layer in the project, which are related to the server or the behavior in Lizmap Web Client.

Configuring the tool

The layer configuration is done in the *Layers* tab.

This tab shows the layers tree of the project with the same organization that is defined in the Layers panel. You can select one of the tree elements, a layer or group, and then configure the options for the selected group or layer.



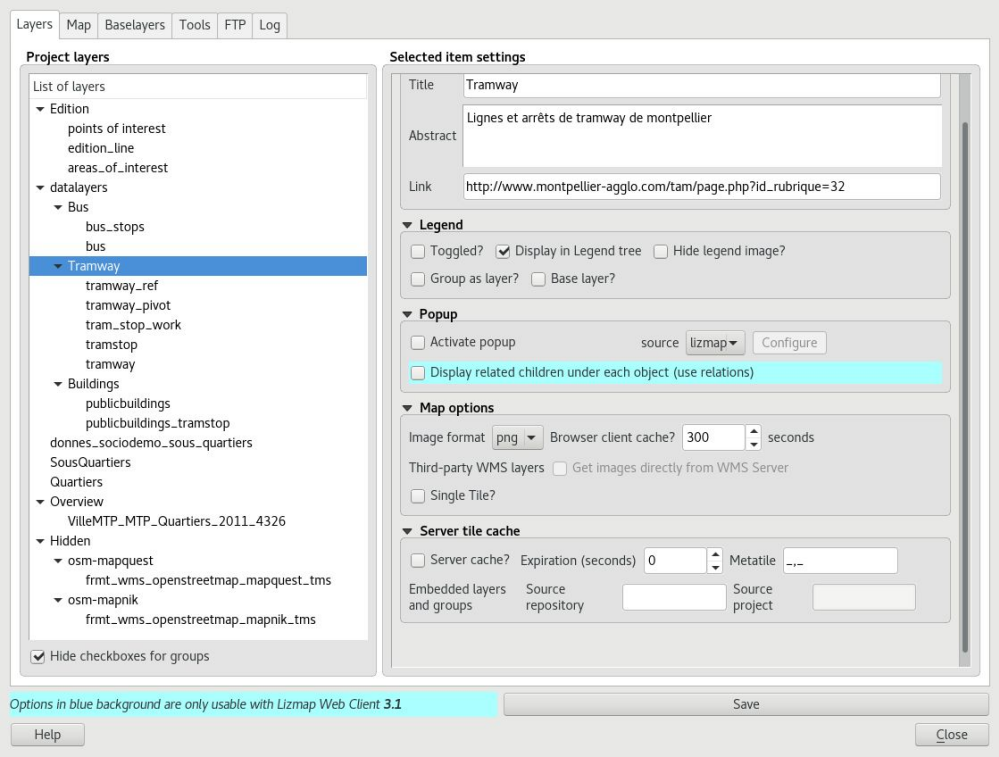
Information about groups and layers:

- *Title*: It will be used in the Web layers tree instead of the name. For layers, the *Title* field is linked to that of the *QGIS Server* tab in the layer properties window
- *Abstract*: It allows to describe the layer or group. It is displayed in the mouseover. For layers, the *Abstract* Field is linked to that of the *QGIS Server* tab in the layer properties window
- *Link*: The URL of a document or a web page describing the layer or group.

An icon appears in the Lizmap legend if the link has been set. You can use the *Media* folder to build an internal link. The button will pick the value set in the *Layer Properties* ‣ *QGIS Server* ‣ *DataUrl*.

Layers options:

- *Toggled*: allows you to specify whether a layer is displayed by default
- *Display in legend tree*: toggles visibility of the layer in the layers tree; when toggled off, it does not allow the user to manage the display of it
- *Hide legend image*: allows you to hide the legend of the layer in the web interface
- *Group as layer*: Option to use on a group in the legend to display it as a single layer, read below
- *Base layer*: sets the layer as a base map. This will be accessible via the list of basemap
- *Activate popup*: enables information popups on map's interrogation by clicking. See *Popup*
- *Single Tile*: selects the layer display mode. It can be displayed as multiple images, tiles, or a single image generated by the server
- *Image format*:
  - `png`: full image format, full range of colors with transparency
  - `png; mode=16bit`: lighter image format, panel color restricted with transparency
  - `png; mode=8bit`: very light image format, color panel restricts to the maximum with transparency, possible degradation of the image
  - `jpeg`: light image format without transparency with loss of quality



If the layer is provided by a WMS service and that it supports the Web map Coordinate Reference System, it is possible to *Directly Request images to the WMS server*. This reduces the QGIS-Server load and optimize Lizmap. This option is available in the *Third Party WMS layers* group.

The group options:

- *Group as layer*:
  - transforms a group in a single layer in the web interface
  - used to group layers with scale dependent visibility
  - simplifies the interface to the users of the Web map
  - to create a base map from multiple layers
- if the group is a layer other options apply to it.
- Server tile cache

3.2.4 Base layers

Contents

- *Base layers*
  - *Principle*
  - *Configuring the tool*
  - *The available base layers*
  - *Scale with external base layer*
  - *External Lizmap layers*

Principle

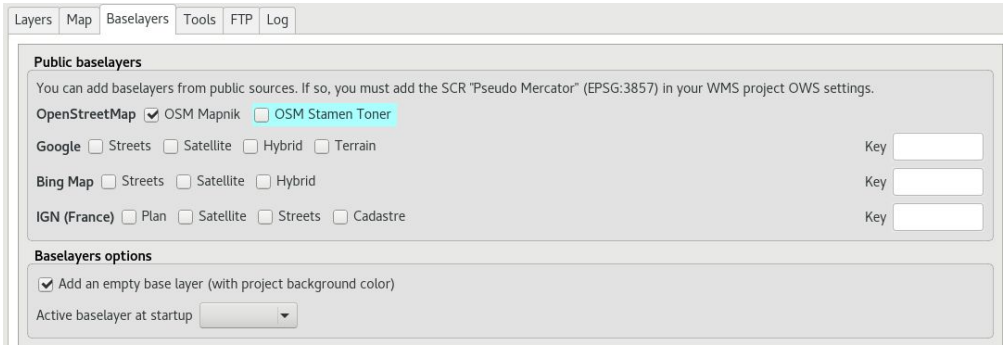
It is often useful to separate base layers as a reference and thematic layers in a Web map. In Lizmap, you can use groups or layers as base layers. It is also possible to use external services in the Web map.

The base layers are not part of the legend and are presented as a list.

Note: If a single base layer is configured (project layer, external service or empty base layer), then Lizmap Web Client interface does not show the box *Base layers*, but the layer will be however visible below the other layers.

Configuring the tool

The *Base layers* tab lets you add external services as base layer and an empty base layer. The empty base layer will display thematic layers over the project background color.



The available base layers

- *OpenStreetMap*, mapping project under free and open licenses:
  - OSM Mapnik: service available on openstreetmap.org
- *ThunderForest*, company using OSM data and providing tiles: * Open Cycle Map: OpenStreetMap cycling data including altitude information
- *Google*, requires compliance to use licence agreement:
  - Streets: the default Google Maps background layer
  - Satellite: the background map incorporating aerial and satellite images
  - Hybrid: the background map mixing streets and satellite
  - Terrain
- *Bing Map*, requires compliance with the Microsoft license agreement and therefore a key:
  - Streets: the default Bing Map background layer
  - Satellite: the background map incorporating aerial and satellite images
  - Hybrid: the background map mixing streets and satellite
- *IGN Géoportail*, requires compliance with the IGN license agreement and therefore a key:
  - Plan: The IGN rendering for the Web
  - Satellite: the background map incorporating IGN aerial and satellite images
  - Scan: the background map mixing the various IGN scan
  - Cadastre

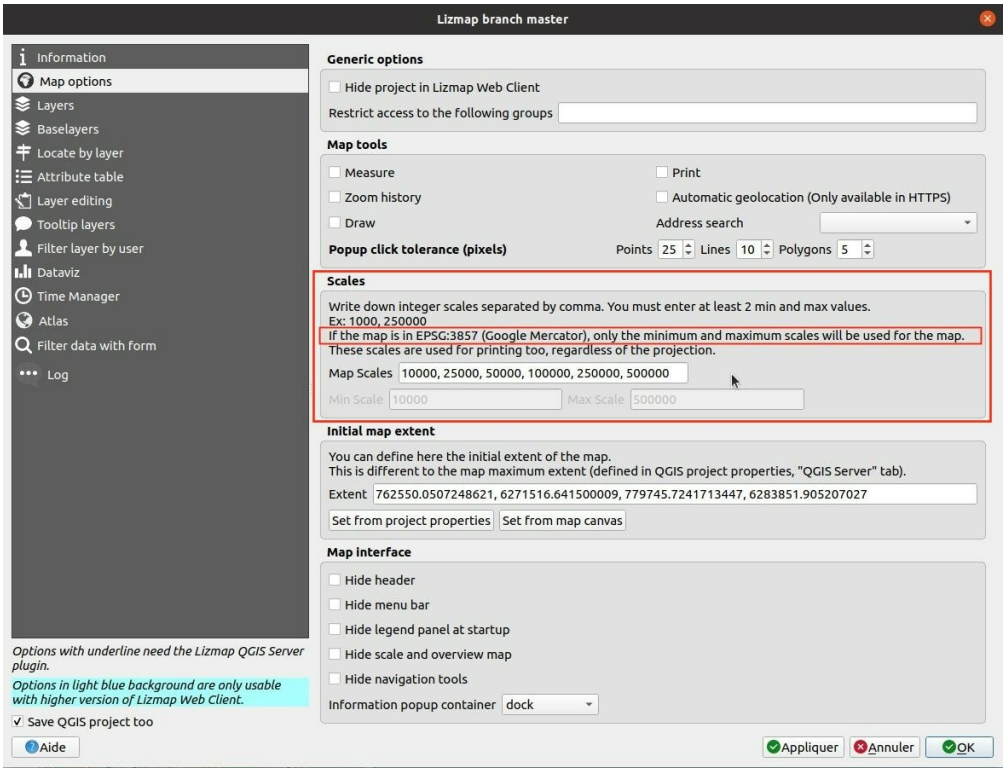
The licenses are available at the following URLs:

- *OpenStreetMap*: https://wiki.openstreetmap.org/wiki/Tile_usage_policy
- *Google*: https://cloud.google.com/maps-platform/terms/
- *Bing*: https://www.microsoft.com/en-us/maps/product
- *IGN*: https://depot.ign.fr/geoportail/api/develop/tech-docs-js/fr/license.html

Scale with external base layer

All these external base layer are provided only in **EPSG:3857 / Pseudo Mercator** from the provider. The scale of such layer are **fixed** by the projection EPSG:3857.

Warning: As a result, if you choose an external base layer, the project **will be displayed** in Google Mercator EPSG:3857. Therefore, if you have specified multiple scales in the Lizmap configuration, these intermediate scales **won't** be used. **Only** the **minimum** and **maximum** scale are used in this case. Read below about approximate scales.



QGIS Server will perform **on the fly reprojection** for your data. It is therefore necessary to prepare the QGIS project accordingly.

The *Google Mercator* coordinate reference system must be added in the Web Services CRSs list with the *Project properties ▸ QGIS Server ▸ WMS*.

This is a list of a **very approximate** integer scale for a given zoom level for the **EPSG:3857**:

```
0   500 000 000
1   250 000 000
2   150 000 000
3   70 000 000
4   35 000 000
5   15 000 000
6   10 000 000
7   4 000 000
8   2 000 000
9   1 000 000
10  500 000
11  250 000
```

(continues on next page)

(continued from previous page)

```
12  150 000
13  70 000
14  35 000
15  15 000
16  8 000
17  4 000
18  2 000
19  1 000
20  500
```

Warning: These scales are just a hint of the scale. It is not possible to have rounded scale like this with EPSG:3857.

Visit the OpenStreetMap website at the country level and check how the scale is changing while panning/zooming.

https://www.openstreetmap.org/#map=6/48.995/4.856

External Lizmap layers

This feature has been removed. It's replaced by the possibility of using the menu *Layer ▸ Embed Layers and Groups*, and in the plugin *Layers* tab declare the parent project and the Lizmap repository for the embed layers and groups. See *Centralizing the cache with the integration of groups and layers from a master project*

3.2.5 Locate by layer

Contents

- *Locate by layer*
  - *Principle*
  - *Use case*
  - *Prerequisites*
  - *Configuring the tool*
  - *Hierarchical Lists*

Principle

The idea of this tool is to present to the Lizmap Web Client user a drop down list that gives the ability to zoom on one or more spatial objects of the layer.

Use case

Consider a spatial vector layer **districts** contained in the QGIS project. We choose to add these districts in the tool *Locate by layer*, to allow Lizmap Web Client users to quickly position on one of the districts.

Once this layer added in the tool *Locate by layer*, a drop down list of the districts appears on the Lizmap Web interface.

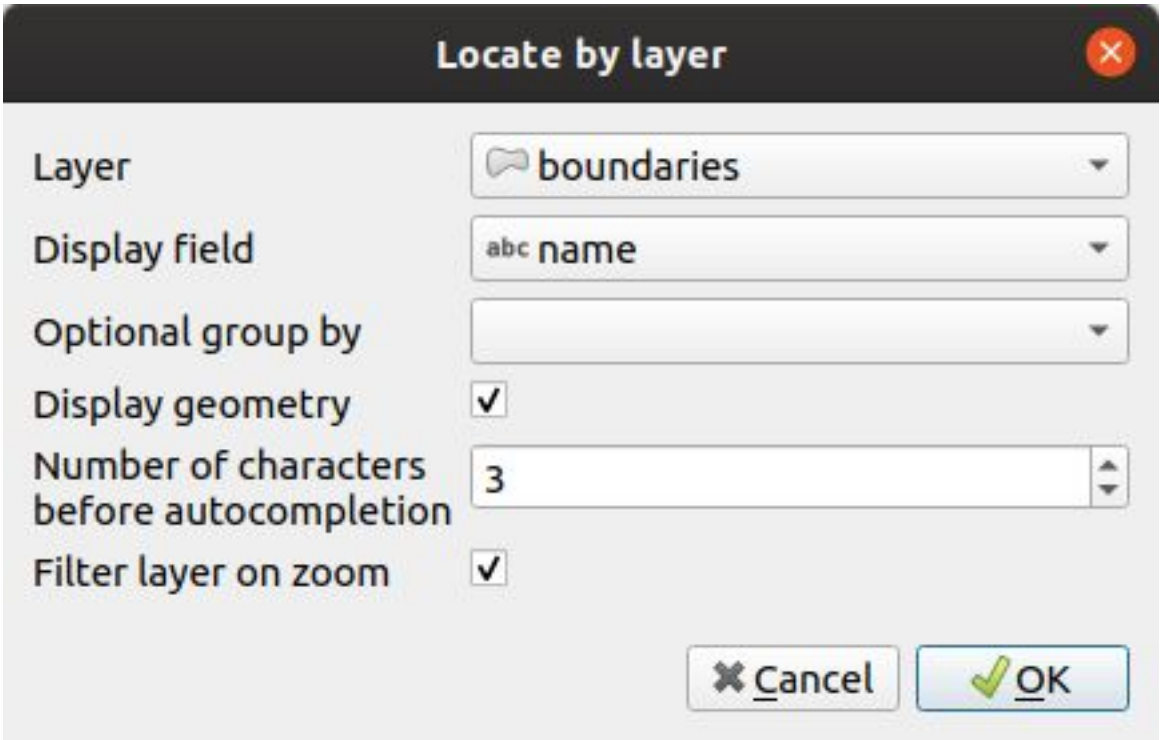
When the Web map user selects one name in this list, the map will automatically refocuses on the selected district and the district's geometry is displayed (optional).

Prerequisites

The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

Warning: This tool is not designed for layers having a lot of features or a lot of complex geometries. The web browser will indeed have to download **all** features with vector geometries so it will make the loading the application slower. To reduce the loading time, you can reduce the decimals in the WFS settings for instance if your layer is in meters. For better performance if your layer is stored in PostGIS, use *PostgreSQL search*.

Configuring the tool



To add a layer to this tool:

1. Click the button.
2. **choose the layer** with the first dropdown from the list of the project vector layers
3. then **the column that contains the label** you want to display in the dropdown list

4. if you want to add pre-filter your data if a optional group by, use the *Optional group by* field.
5. if you want the geometry of the related objects is also displayed on the map when the user selects an item from the list, then check the option *Display the geometry*
6. If you set a value above 0, autocompletion will be used after this amount of characters while the user types. The classical combobox will be replaced by a editable text input.
7. If Lizmap must trigger the filter on the layer. Only the selected feature will be visible on the map.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.

Hierarchical Lists

If we take the example of districts, it may be interesting to also provide to the user a *sub-districts* dropdown. We wish that when the user chooses a district, the dropdown of sub-districts is automatically filtered to display only the sub-districts of the chosen district.

For this, there are 2 methods:

- you either have **2 separate vector layers**: one for districts and for sub-districts. So you have to use a **field join** between the two layers to enable automatic filtering lists in Lizmap
- either we have **only 1 layer for sub-districts**, and then you can specify with the plugin a **group field**. Two dropdowns will be created instead of one in the Web application.

Note: Don't forget to check if your configuration of QGIS server is the right one (in Lizmap : My account -> Lizmap configuration -> check if the qgis server version is the right one, if not, you can use the button 'modify' below). If the configuration is not right the location will be wrong!

3.2.6 Attribute table

Contents

- *Attribute table*
  - *Principle*
  - *Prerequisites*
  - *Configuring the tool*
    - *At the layer level*
    - *At the project level*
  - *Order of fields*
  - *Download data*
  - *Using with relations*

Principle

Lizmap is designed to show spatial data in the main map, and you can propose users to see an object data through the "popup" feature (a small popup containing the objects data is shown whenever the user clicks on the map). See *Popup*

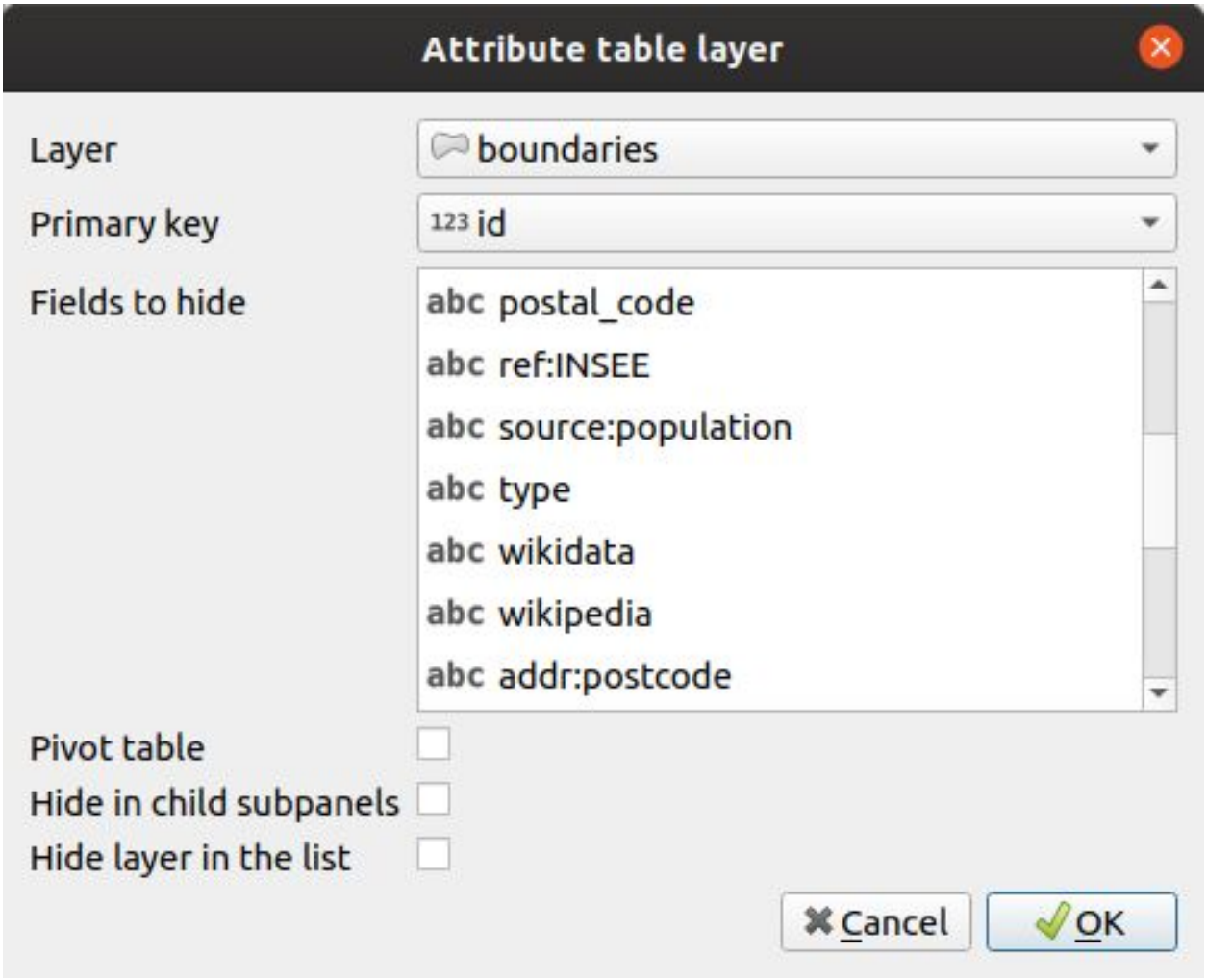
Sometimes this is not enough, and as a map editor, you would like the user to see all the data of a specific layer, as you can do in QGIS by opening the attribute table.

Prerequisites

The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

Configuring the tool

At the layer level



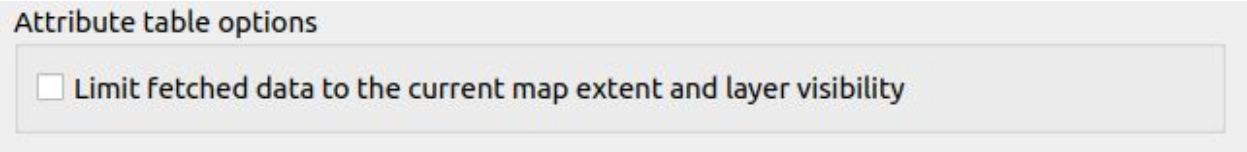
In the **Tools** tab of Lizmap plugin dialog, there is a group called "Attribute layers" which shows a table and some form fields dedicated to add or remove vector layers.

To add a layer to this tool:

1. Click the button.
2. **Layer**: Choose one of the vector layers (spatial or not). This can be any vector layer format : GeoJSON, Shapefile, PostGIS, CSV, etc.
3. **Unique ID**: The attribute table tool needs to be able to defined each feature as unique. We strongly advise you to add such a field if your layer has not one yet. Usually the unique ID field contains **integers**. If the layer do not have this kind of field, you can easily create it with the *Field calculator*. Choose the correct field with the combo box.
4. **Fields to hide**: You have 2 ways of hiding fields in the published attribute table.
   - In the *vector layer properties dialog* of the QGIS vector layer, in the *Fields* tab, you can uncheck the checkbox of the column **WFS** for the fields to unpublish. This means this fields will not be published via the WFS protocol. This is the **simplest and safiest way** to restrict the publication to some fields (for example to get rid of sensitive fields)
   - You can use this **Fields to hide** option to **hide** the given fields in the attribute table display. The hidden fields won't be visible for the end user, but will still be available for Lizmap Web Client. **You must use this option to hide the Unique ID field**. If you use the first way (uncheck WFS column), the unique ID won't be usable by Lizmap, and some of the attribute table features will not work properly.
5. Let the *Pivot table* unchecked. This is for relation.
6. Let the *Hide in child subpanels* unchecked. This is for relation.
7. Let the *Hide layer in the list* unchecked. This is for relation.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.
- Fields will be displayed with their alias if it's defined. See *Add a alias on a field*.

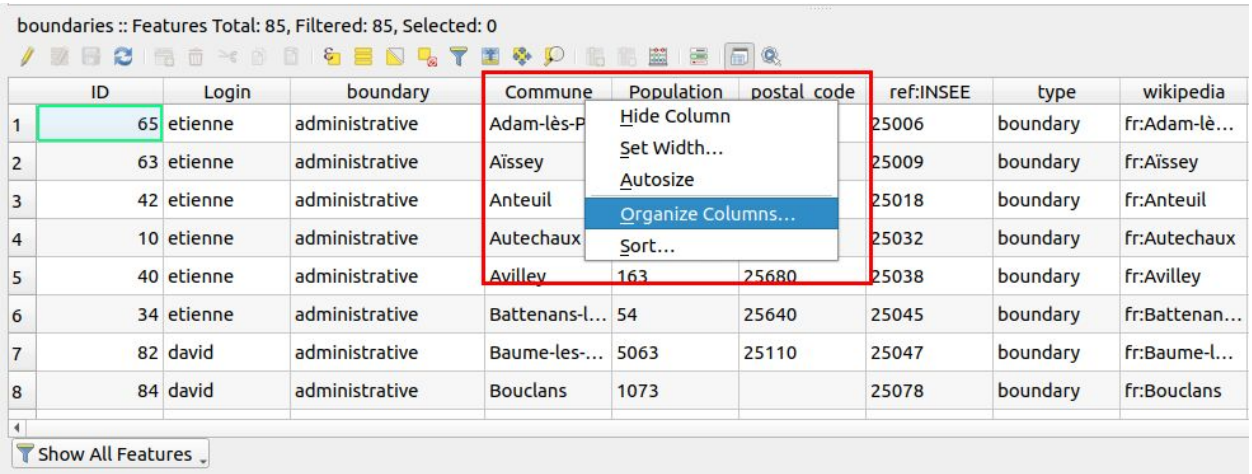
At the project level



Order of fields

Lizmap is reusing the order of the fields defined in the attribute table.

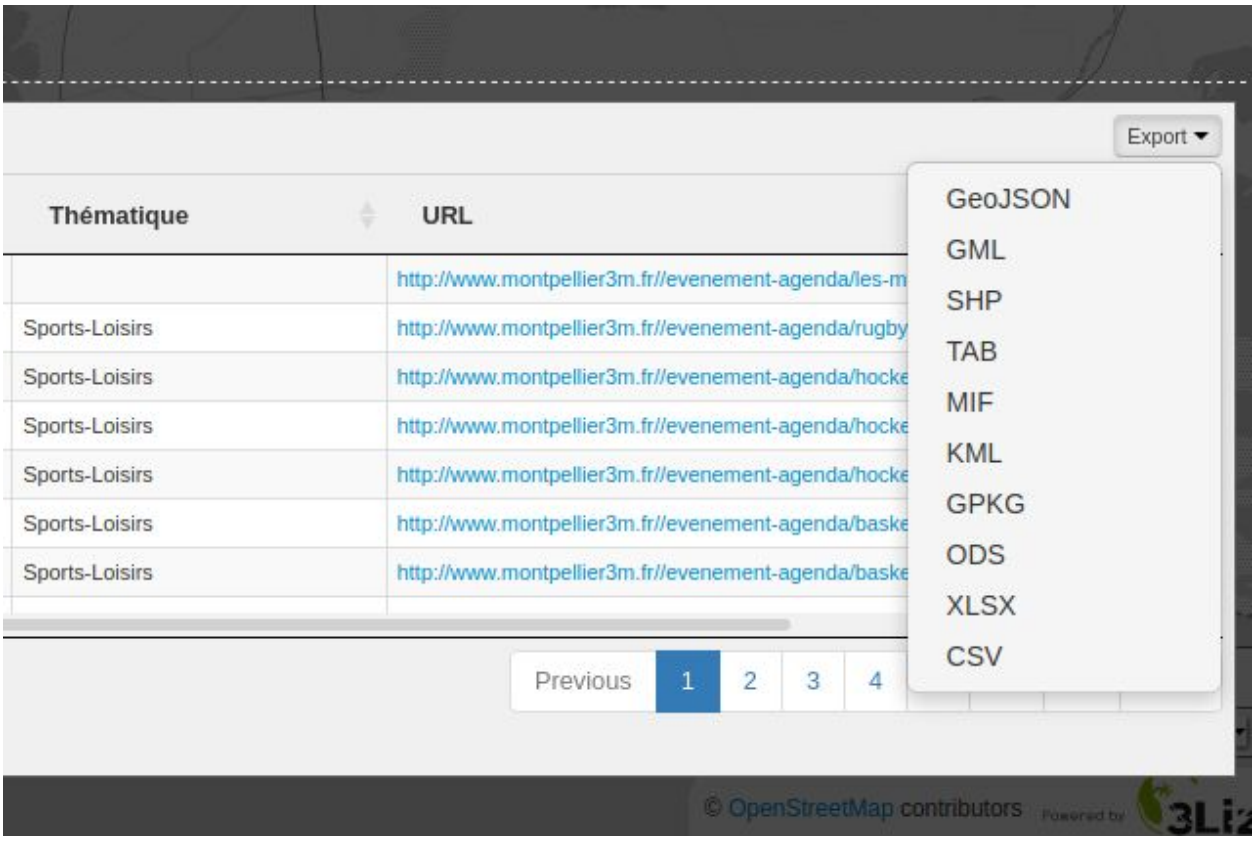
1. Open your attribute table.
2. Right click on a column header.
3. Click *organize Columns…*
4. Drag and drop columns.



	ID	Login	boundary	Commune	Population	postal_code	ref:INSEE	type	wikipedia
1	65	etienne	administrative	Adam-lès-P			25006	boundary	fr:Adam-lè...
2	63	etienne	administrative	Aïssey			25009	boundary	fr:Aïssey
3	42	etienne	administrative	Anteuil			25018	boundary	fr:Anteuil
4	10	etienne	administrative	Autechaux			25032	boundary	fr:Autechaux
5	40	etienne	administrative	Avilley	163	25680	25038	boundary	fr:Avilley
6	34	etienne	administrative	Battenans-l...	54	25640	25045	boundary	fr:Battenan...
7	82	david	administrative	Baume-les-...	5063	25110	25047	boundary	fr:Baume-l...
8	84	david	administrative	Bouclans	1073		25078	boundary	fr:Bouclans

Download data

By adding the layer in the attribute table tool, this will also enable the download capability on that layer. By default, only GeoJSON and GML are available. By installing the *WfsOutputExtension* plugin on QGIS Server, you can add more formats such as Geopackage, ODS, CSV, XLSX …. Read https://github.com/3liz/qgis-wfsOutputExtension



Using with relations

In QGIS, you can configure **relations** between layers, in the **project properties** dialog. If you publish in Lizmap more than one layers in the attribute layers tool, and if some layers are part of a relation, the end user will be able to see child tables under the parent layer table, and a click on one line in the parent table will trigger the filter of the child tables content.

For example, you could have a layer of cities, and a child layer of public building. Clicking on one city in the attribute table will make the public building child table refresh its content with only the public buildings of the clicked city.

You can use "many-to-many" (N:M) relations in QGIS since version 2.14 but Lizmap doesn't use it automatically yet. You have to define the relations in the Lizmap plugin too. In many cases, N:M relations are very handy. For example, you can have the three following vector layers in your project:

- **Tramway lines**: this layers name "Lines" contains one feature per tram line, and has a unique ID field *tram_id*
- **Tramway stops**: this layer named "Stops" contains one feature per tram stop, with a unique ID field called *stop_id*
- **Correspondance table between lines and stops**: this layer named "Pivot" is a pivot table between tram lines and stops, since a stop can be used for more than one line, and a line serves many stops. It has the following fields: *tram_id*, *stop_id* and *order* which defines the order of the stop in the line.

You can add 2 relations in QGIS project properties : one between Lines and Pivot using the tram_id field, and one between Stops and Pivot using the stop_id field.

In Lizmap, we added a simple way to configure the N:M relation. You can simply

- Create the two relations described above in QGIS project properties dialog, tab **Relations**
- Add the Lines and Stops layers in the attribute layers tool
- Add the Pivot layer in the attribute layers tool with the option **Pivot table** checked

Lizmap Web Client will then handle the relation as a N:M relation:

- The pivot table will be displayed under each parent attribute table and show only the corresponding children.
- The filter feature based on the attribute layers will trigger the cascading filter of the pivot and the other parent. For example, if the user uses the filter to show only one tramway line, Lizmap will also only show the corresponding stops in the map and in the Stops attribute tables

3.2.7 Edition

Contents

- *Edition*
  - *Principle*
  - *Examples*
  - *Prerequisites*
  - *Configuring the tool*
  - *Reusing data of edition layers*
  - *Adding files and images for features*
    - *Use the media/upload folder relative to the project*
    - *Use a specific destination folder*

Principle

It is possible to allow users to **edit spatial and attribute data** from the Lizmap Web Client interface for **PostgreSQL** layers. The Lizmap plugin allows you to add one or more layers and choose what actions for each will be possible in the web interface:

- creating elements
- modifying attributes
- modifying the geometry
- deleting elements

The **Web form** presented to the user to populate the **attribute table** supports **editing tools** available in the *fields* tab of the QGIS Vector *layer properties*. You can configure a dropdown, hide a column, make it non-editable, use a check box, a text area, etc. All configuration is done with the mouse, in QGIS and the Lizmap plugin.

In addition, Lizmap Web Client automatically detects the column type (integer, real, string, etc.) and adds the necessary checks and controls on the fields.

Examples

- **A town** wish that citizens identify visible problems on the road: uncollected trash, broken street lights, wrecks to remove. The QGIS project administrator creates a layer dedicated to collect data and displays them to all.
- **An engineering office** wants to allow project partners to trace remarks on the project areas. It allows the addition of polygons in a dedicated layer.

Prerequisites

To allow data editing in Lizmap Web Client, you must:

- Have vector layer stored in PostgreSQL.
- The vector layer mustn't have space in field names.
- Configure the editing tool for the layer in *Layer Properties* ‣ *Attributs Form*. This is not required but recommended to control the data entered by users. See *Customize the edition form* for more information about layout, widgets, expressions, constraints in a form.
- The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.
- Despite we want to edit the layer, there is no need to use *Update*, *Insert* and *Delete* checkboxes in the WFS table in the *QGIS Server* tab. Lizmap does not use WFS-T. Lizmap will make the edit directly on the datasource. The configuration is done only in the panel described below.
- As a consequence as above :
  - The credentials **must not** use the **Authentification system provided by QGIS** for a layer with edition capabilities. Credentials must be either in the **QGS** project file or in the PostgreSQL service file (recommended, more secure because credentials are not stored in the QGIS project) :
    - How to use service file on docs.qgis.org
    - How to use service file on postgresql.org

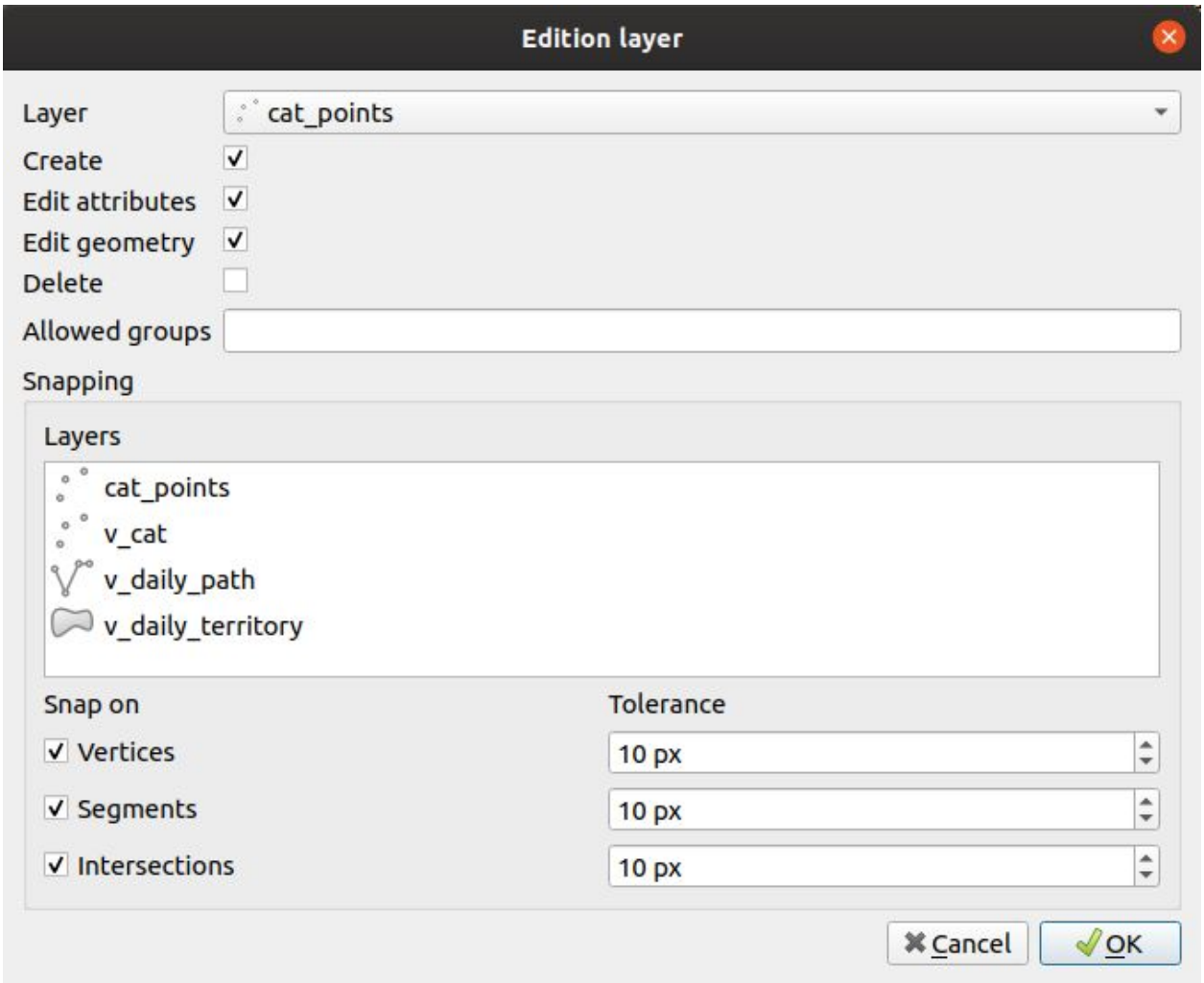
Note: Be careful if your layer contains some Z or M values, unfortunately Lizmap will set them to "0" which is the default value when saving to the database.

Configuring the tool

Here are the detailed steps:

- If necessary, **create a layer** in your database with the desired geometry type (point, line, polygon, etc.)
  - think about adding a **primary key**: this is essential!
  - the primary key column must be of type **auto-increment**. For example *serial* to PostgreSQL.
  - think about adding a **spatial index**: this is important for performance
  - *create as many fields as you need for attributes*: if possible, use simple field names!

Please refer to the QGIS documentation to see how to create a spatial layer in a PostgreSQL database: https://docs.qgis.org/latest/en/docs/user_manual/managing_data_source/index.html



- To enable a layer with edition capabilities:

1. Click the button.
2. *Select the layer* in the drop-down list
3. **Check the actions you want to activate from:**
   - *Create*
   - *Edit attributes*
   - *Edit geometry*
   - *Delete*
4. Optional, you can add a list of groups which are allowed to edit, separated by a comma.
5. **Snapping can be activated if you select at least one layer in the layer list.**
   - The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.
6. **If one layer is selected above, at least one checkbox must be used :**
   - *Vertices*
   - *Segments*
   - *Intersections*
7. It's possible to set the tolerance for the snapping.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.
- Fields will be displayed with their alias if it's defined. See *Add a alias on a field*.

Reusing data of edition layers

The layers that you have selected for the editing tool are "layers like the others", which means:

- **QGIS styles and labels are applied to these layers.** You can create styles and labels that depend on a value of a layer column.
- If you want to propose the editing tool, but does not allow users to view data from the online layer (and therefore the additions of other users): **you can simply hide edition layers** by putting them in a *hidden* directory. See *Masking individual layers*
- **The layers are printable** if they are not masked.
- **The data are stored in a layer of the project**. The administrator can retrieve this data and use them thereafter.

Note: Using the cache: whether to use the server or client cache for editing layers, do so by knowingly: the data will not be visible to users until the cache has not expired. We suggest not to enable the cache for editing layers.

Adding files and images for features

Use the media/upload folder relative to the project

It is possible to upload your files, including images, for each feature, during online editing; to achieve this, you need to:

- Configure edition for the layer, with one or more fields with the *edit type* `Photo` or `File`. For example, let say the field name is `photo`
- Create a folder at the root of the QGIS project file : `media/` and a subfolder `media/upload` (obviously you need to do that locally in your computer and server side).
- Give to the webserver user (usually www-data) **write permission** on the upload folder, so that it can create files and folders in `media/upload`:

  chmod 775 -R media/upload && chown :www-data -R media/upload

- Check you `php.ini` to see if the variables `post_max_size` and `upload_max_filesize` are correctly set (by default, PHP only allows uploading files up to 2 Mbyte)

Lizmap will then create folders to store the data, depending on the layer name, field name, etc. For example, a file would be stored in the folder `media/upload/PROJECT_NAME/LAYER_NAME/FIELD_NAME/FILE_NAME.EXT` and an image in `media/upload/environment/observations/species_picture/my_picture.png`.

Obviously you will be able to display this image (or any other file) in the popup, as it will be stored in the media folder. See *Use in popups*

Use a specific destination folder

You can override the default destination folder `media/upload/PROJECT_NAME/LAYER_NAME/FIELD_NAME/FILE_NAME.EXT` by manually setting the path where to store the media, relatively to the project. To do so, you must use the *External resource* field edit widget, and configure it with:

- a *Default path* written relative to the project. For example `../media/images/` if you want to store this field files in a folder media situated alongside the project folder. You can also choose set a path inside the project media folder. For example `media/my_target_folder/`.
- check the *Relative path* checkbox, with the default *Relative to project path* option
- if the field should store a image, you should also check the *Integrated document viewer* checkbox, with the type *Image*

This allow to store the sent media files and images in a centralized folder, for example a directory `media` at the same level than the Lizmap repositories folders:

- media
  - images <– to store images in this folder, use `../media/images/`
  - pdf
- environment
  - trees.qgs
  - trees.qgs.cfg
  - media
    - tree_pictures/ <– to store images in this folder, use: `media/tree_pictures/`

3.2.8 Tooltip

Contents

- *Tooltip*
  - *Principle*
  - *Prerequisites*
  - *Configuring the tool*

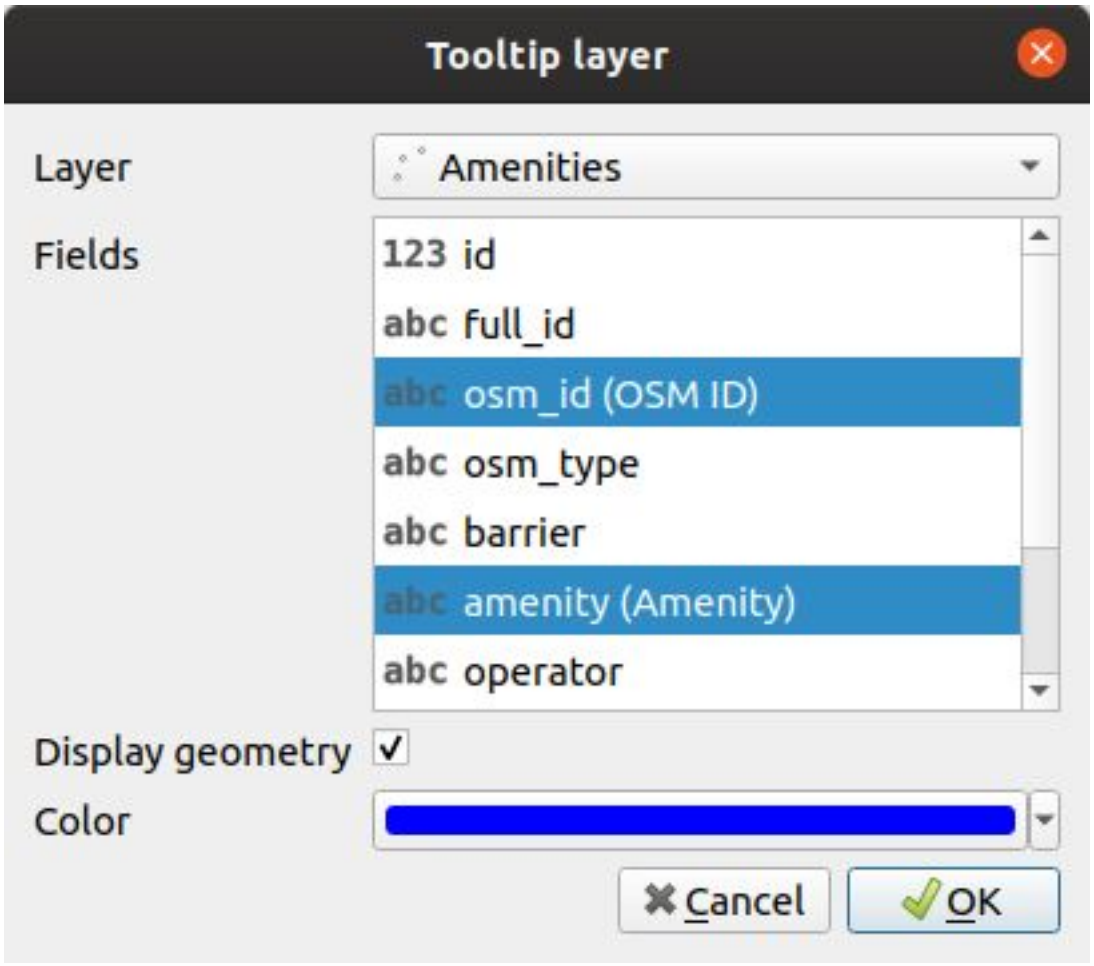
Principle

You can let the user activate map tooltips when hovering features by choosing a list of fields to display.

Prerequisites

The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

Configuring the tool



- For setting a tooltip on a layer:
  1. Click the button.
  2. Choose the layer.
  3. Select some fields to display in the tooltip.
  4. Optionally, choose to display the geometry.
  5. If you display the geometry, you can set the color.
- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- Fields will be displayed with their alias if it's defined. See *Add a alias on a field*.

3.2.9 Filtered layer by user

Contents

- *Filtered layer by user*
  - *Principle*
  - *Filtering by spatial relationship*
    - *Principle*
    - *Prerequisites*
    - *Configuring the tool*
  - *Filtering by attribute*
    - *Example*
    - *Configuring the tool*
    - *Edition*

Principle

The filtering feature presented allows you to filter the data displayed on the map based on the logged in user. It is possible to filter only vector layers because Lizmap uses a column in the attribute table.

Filtering currently uses the ID of the user group connected. It's active for all requests to QGIS server, and thus concerns:

- the vector layers images displayed on the map
- the popups. See *Popup*
- the *Locate by layer* feature lists. See *Locate by layer*
- drop-down lists of *Editing forms* from *Value relation*. See *Edition*
- the attribute table display etc

There are two kinds of filtering :

- based on attributes
- based on spatial relationship

Filtering by spatial relationship

Principle

In this panel, we can:

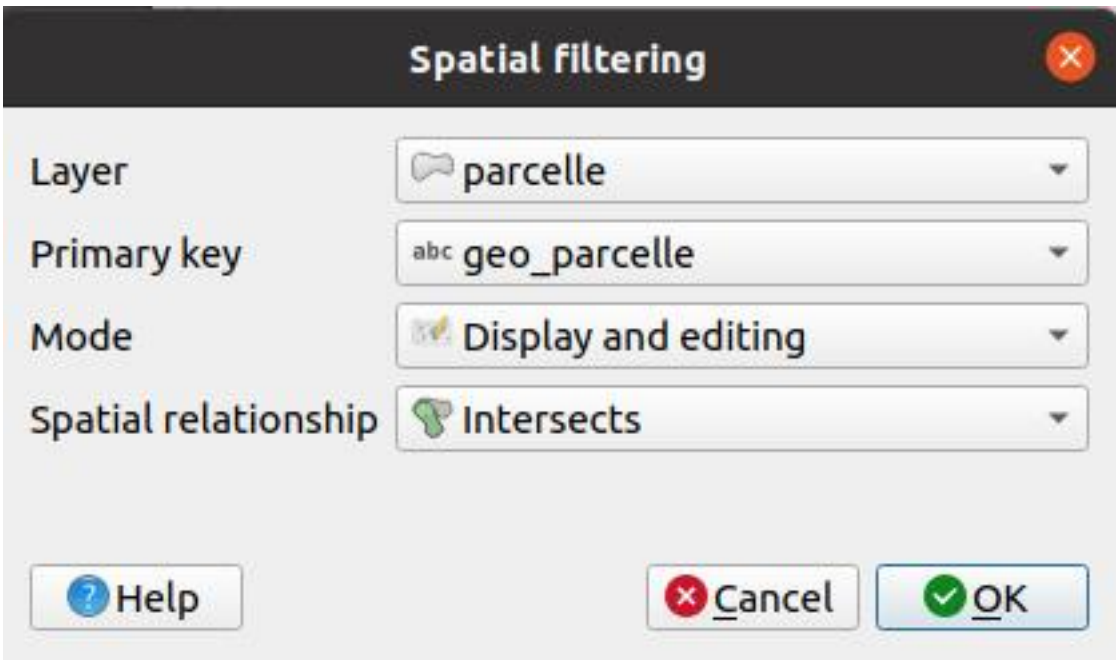
- configure a polygon layer to use for spatial filtering
- configure layers which must be filtered by the filtering layer

Prerequisites

This is a feature in **Lizmap 3.5**.

The Lizmap plugin in QGIS Server **must** be installed with at least **Lizmap QGIS plugin 3.6**. Otherwise, the filter won't work and all data will be visible. See *Lizmap QGIS Server plugin*.

Configuring the tool



1. Choose the polygon layer used for filtering
2. Choose the field in this layer which has Lizmap groups, separated by a comma.
3. Add a layer to filter to this tool:
    1. Click the button.
    2. Choose the layer to filter
    3. Choose the primary key of the layer
    4. Choose if the filtering if for both visualisation and editing or only editing
    5. Choose either **intersection** or **contain** spatial relationship.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.

For your information, performance will be better when using PostgreSQL layers : either filtering or filtered layers or both.

Filtering by attribute

Example

A video tutorial is available at: https://vimeo.com/83966790

Configuring the tool

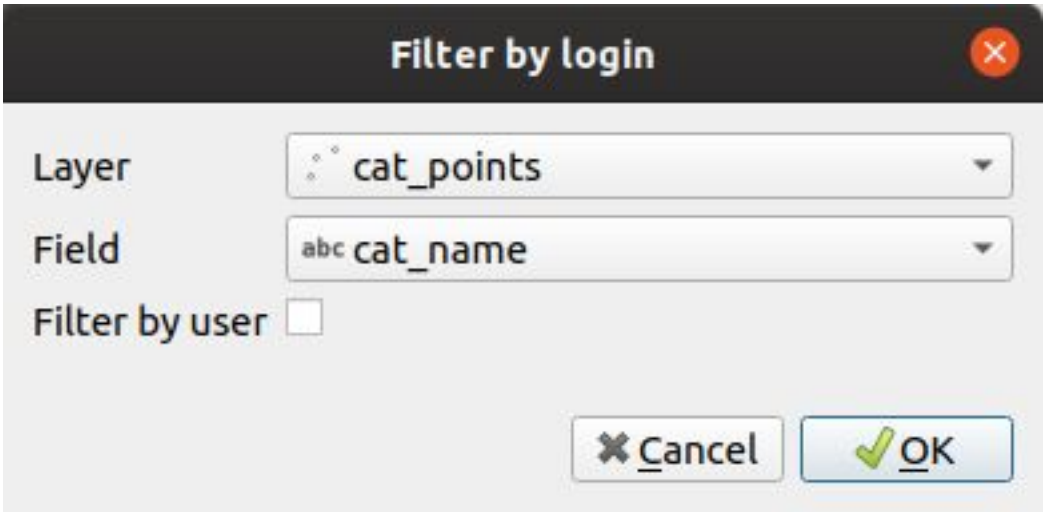
To use data filtering tool in Lizmap Web Client, you must:

- have **access to the administration interface** of Lizmap Web Client

Here are the detailed steps to configure this feature:

> **Warning:** In Lizmap Web Client administration, in the repository properties, be sure that `anonymous` and other relevant groups are not checked for *Always see complete layers data, even if filtered by login*. See *Define the rights for each group*.

- Check group ID or user ID in the administration interface:
  - To know group ID, you must go to the administration interface *SYSTEM ▸ Groups of users for rights*: ID appears in parentheses after the name of each group (under the title *Groups of new users*)
  - Or to know the user ID, just use its login.
- For all vector layers which is desired filter data, just add a text column that will hold the ID for each line who has the right to display this feature.
  - *Fill this column* for each line of the attribute table with the ID (using the calculator, for example).
  - It is possible to set `all` as the value in some lines to disable the filter: All users will see the data for these lines.
  - If the value in this column for a row does not correspond to a user group, then the data will be displayed for no user.



- Add the layer in the table *Filter Data by User* panel:
  1. Click the button.
  2. Select layer from the dropdown list

3. Select the field with the ID
4. Check the box if the field contains user ID, instead of group ID

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.

Warning: Disable the client cache and cache server for all filtered layers. Otherwise, the data displayed will not be updated between each connection or user logout!

Edition

If you have enabled the edition (see *Edition*) on the filtered layer, the field containing the ID will be editable with:

- either a combobox for a non admin user with either his own login or `all`
- or a line edit to write a value for a admin user

3.2.10 Dataviz - display some graphs

Contents

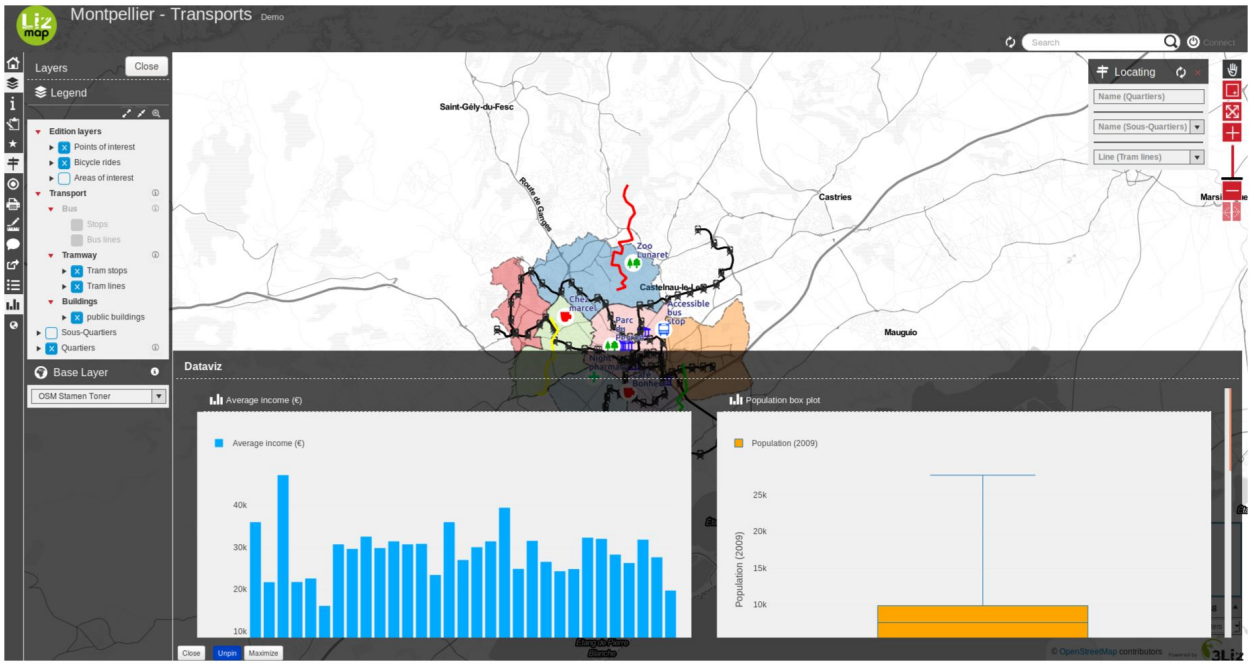
- *Dataviz - display some graphs*
  - *Principle*
  - *Prerequisites*
  - *Configuring the tool*
    - *At the layer level*
    - *At the project level*
    - *JSON layout*
    - *Types of charts*
    - *Miscellaneous*
  - *Examples*

Principle

With the dataviz panel, you can create a few kinds of graph with only a few clicks:

- scatter
- pie
- histogram
- box
- bar
- histogram2d

- polar
- sunburst **Lizmap 3.4**
- HTML **Lizmap 3.4**



Prerequisites

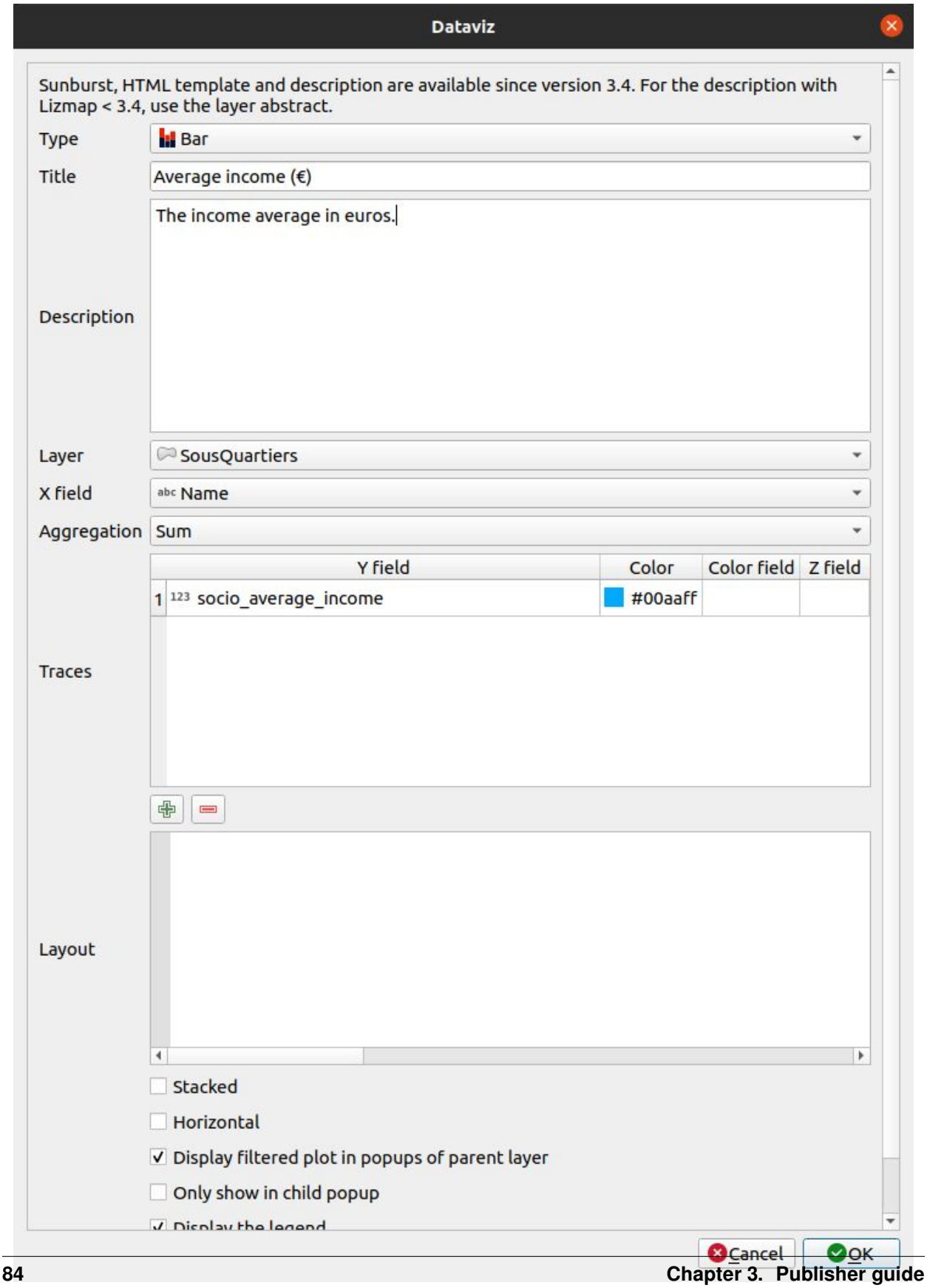
The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

Configuring the tool

Tip: You can start using the plugin `DataPlotly` to create your graph in QGIS itself. So you can have a preview about what is possible *more or less* about dataviz with your layers. But keep in mind that Lizmap and DataPlotLy, even if's using the same dataviz engine (plotly https://github.com/plotly/plotly.py), features are different between these two tools.

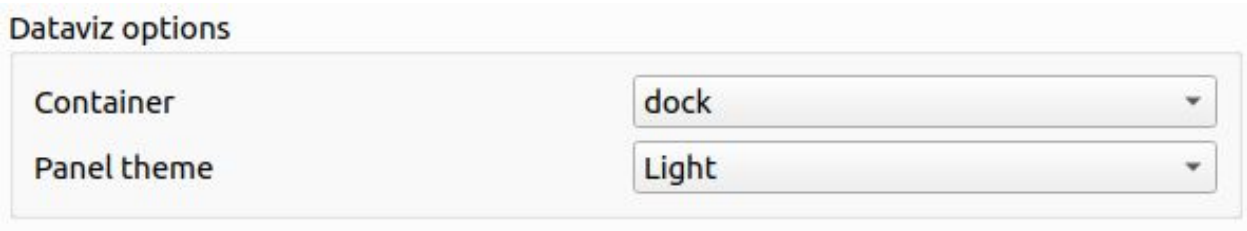
You can easily configure it with the plugin Lizmap in QGIS in the *Dataviz* panel.

At the layer level



- To enable a layer with dataviz capabilities:
  1. Click the button.
  2. Select the type of chart to add. According to your choices, the form will adapt it self. Read below the kind of charts.
  3. **Title** : Here you can write the title you want for your chart.
  4. **Description** : The description of the chart. You can include HTML.
  5. *Select the layer* in the drop-down list.
  6. **X field** : The X field of your graph. It might be empty for a few types.
  7. **Aggregation** [For a few types of charts like bar or pie, you can chose to aggregate the data in] the graph. There are a few aggregate functions available - average(avg), sum, count, median, stddev, min, max, first, last.
  8. **Traces** : Depending of the kind of chart, you can add one or many traces : the Y field of your graph.
  9. Depending of the kind of chart, there is now different options.
  10. **Layout** : The layout can be customized. It must be a JSON dictionary. You can read the documentation of Plotly documentation about the layout configuration https://plotly.com/javascript/reference/#layout
  11. **Display filtered plot in popups of parent layer** : if you check this checkbox, the children of your layer will get the same graph as the parent plot but filtered only for them. It's useful if you want to see the statistics of one entity instead of all.
  12. **Only show child popup** : The main graph will not be shown in the main container and only the filtered graph of the relation of the layer will be displayed in the popup when you select the element.
  13. **Display the legend**, sometimes, the legend is not necessary.
  14. **Display plot only when the layer is visible**.
  15. Some options might be visible or not according to the kind of chart, like choosing for horizontal/vertical layout for a bar chart.
- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.
- Fields will be displayed with their alias if it's defined. See *Add a alias on a field*.

At the project level

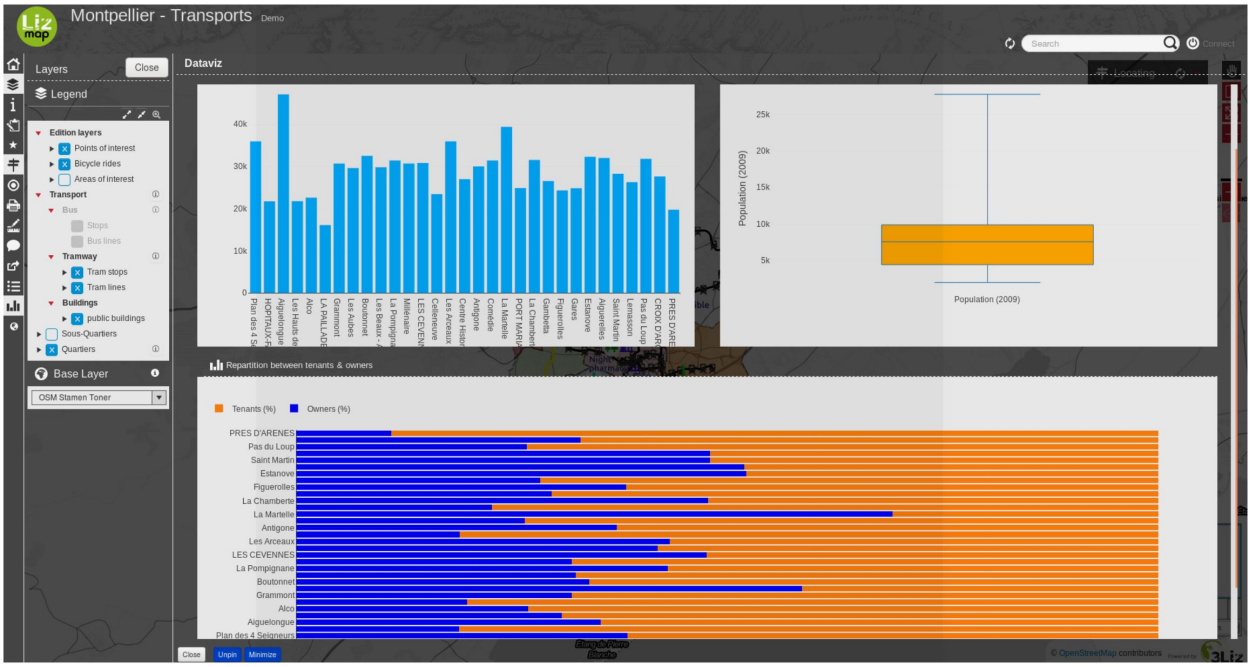


1. You have the possibility to change the value to **dock**, **bottomdock** or **right-dock** these options change where your dataviz panel will be located in your Lizmap's project. You have 3 positions available, at the right of the screen, bottom and right.

2. You have the possibility to write in HTML the layout of the container of your charts. If you are proficient in the HTML language, there are a lot of possibilities and you can customize your container the way you want.

For instance, this bootstrap HTML code will produce the layout below:

```
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span6">$0</div>
        <div class="span6">$1</div>
    </div>
    <div class="row-fluid">
        <div class="span12">$2</div>
    </div>
</div>
```



JSON layout

The layout option, allows you to configure many options by writing up a JSON object. See the full reference: https://plotly.com/javascript/reference/#layout

The following examples show how to use some properties depending of the layout property:

- Add units in the Y (vertical) axis numbers:

```
// Add " ha" suffix
{"yaxis": {"ticksuffix": " ha"}}
```

- Remove labels in the X (horizontal) axis:

```
// Hode X labels. Use yaxis for Y labels
{"xaxis": {"showticklabels": false}}
```

- Remove the automatic *k*, *M* or *G* suffix (ex: *10 M*) for big number in the axis and in the hover tooltip:

```
{"yaxis": {"tickformat": ",.2f"}}
```

- Order of the bars for a bar chart. Use ascending or descending values:

```
// By ascending X value
{"xaxis": {"categoryorder": "category ascending"}}

// By descending bar height (total Y value)
{"xaxis": {"categoryorder": "total ascending"}}
```

- Set margins. Use *l* for left, *t* for top, *r* for right and *b* for bottom:

```
// Shift the plot to the right (to display big Y axis labels)
{"margin": {"l": 70}}
```

- Define font properties:

```
// Change X axis font size
{"xaxis": {"tickfont": {"size": 10}}}
```

- You can combine any of the properties, for example:

```
{
    "xaxis": {
        "categoryorder": "total ascending"
        "tickfont": {
            "size": 10
        }
    },
    "yaxis": {
        "showticklabels": false
    }
}
```

Types of charts

The dataviz module will aggregate every **traces** for all unique **X** values.

HTML templates

A trace must be referenced by `{y1}` where `1` is the trace number.

```
<style type="text/css">
table.example-table {
    width: 100%;
    font-weight: bold;
    color:#4a4a4a;
    padding: 10px;
    border-spacing: 10px;
    border-collapse: separate;
}
table.example-table tr td {
    background-color: #fca487;
    border-radius: 5px;
```

(continues on next page)

(continued from previous page)

```
    border: 10px solid #fff0;
    padding: 10px;
    margin: 10px;
    text-align: center;
}
</style>
<table class="example-table">
    <tbody>
        <tr>
            <td>{$y1}</td>
        </tr>
    </tbody>
</table>
```

Miscellaneous

- For **pie chart** : the items in the legend are ordered by X as a default behaviour.
- QGIS Field **aliases** are used in the tooltip when hovering the bar or pie slices.

Examples

You can visit the Cats project on https://demo.lizmap.com

3.2.11 Time Manager - animation of temporal vector layers

Contents

- *Time Manager - animation of temporal vector layers*
  - *Principle*
  - *Example*
  - *Prerequisites*
  - *Configuring the tool*
    - *At the layer level*
    - *At the project level*

Principle

You can create animations of your vectors, provided you have at least a layer with a column with a valid date/time.

Example

A video tutorial is available here: https://vimeo.com/83845949. It shows all the steps to use the functionality.

Prerequisites

The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

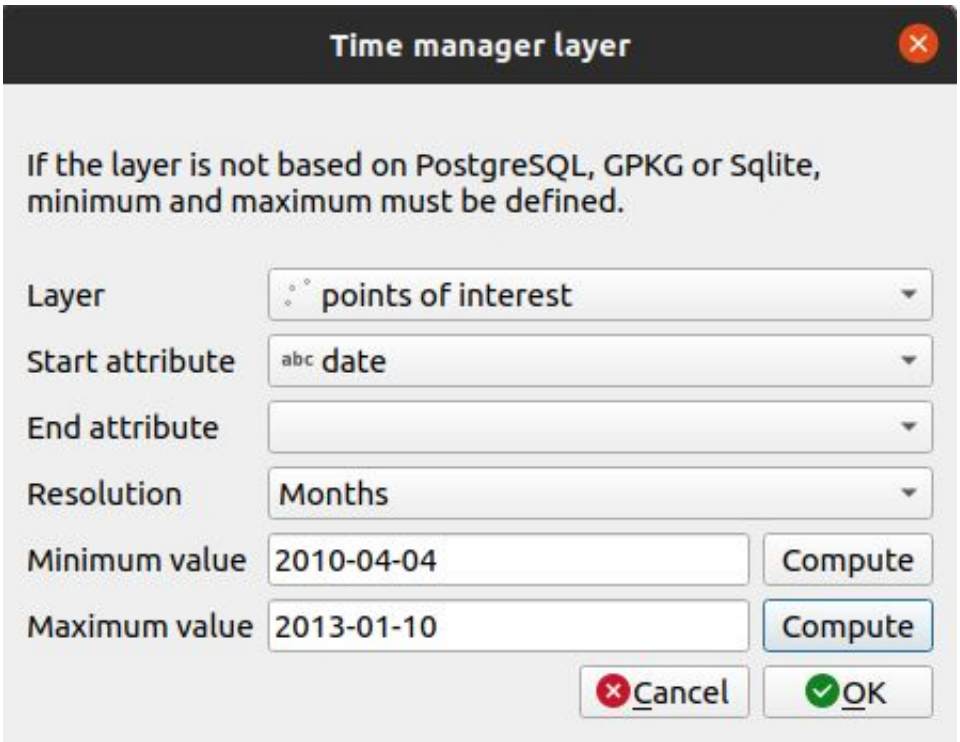
Configuring the tool

After the configuration, your web application will display the symbol of a watch; clicking on it will open a small panel that will allow you to move between steps, or paly the entire animation. At startup, the application will load the entire table, so if you have thousands of objects you may need to wait for several seconds before the application is available.

Note: Several different formats for date/time are acceptable (those supported by the JavaScript library *DateJS*). You can check whether your format is supported by entering it in this page: https://github.com/datejs/Datejs

---

At the layer level



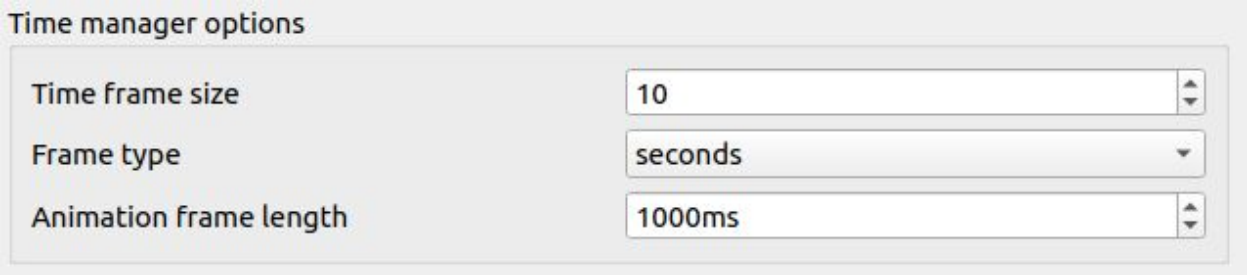
- For setting the time manager with one layer:

1. Click the button.
2. One layer with date/time capabilities.
3. The start column with date/time. It can any kind of fields.
4. The end column with date/time. This is optional.
5. Date/time resolution of the chosen attribute(s).
6. For not database based layer, you must compute the minimum and maximum values.

Warning: Date/time resolution can have a different value than *Frame type*. You might select `years` for *Frame type* but your date field could have a `Days` resolution.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.

At the project level



Options:

- Time frame size
- Frame type
- Animation frame length

3.2.12 Atlas - a sequence of entities

Contents

- *Atlas - a sequence of entities*
  - *Principle*
  - *Configuring the tool*
    - *At the layer level*
    - *At the project level*

Principle

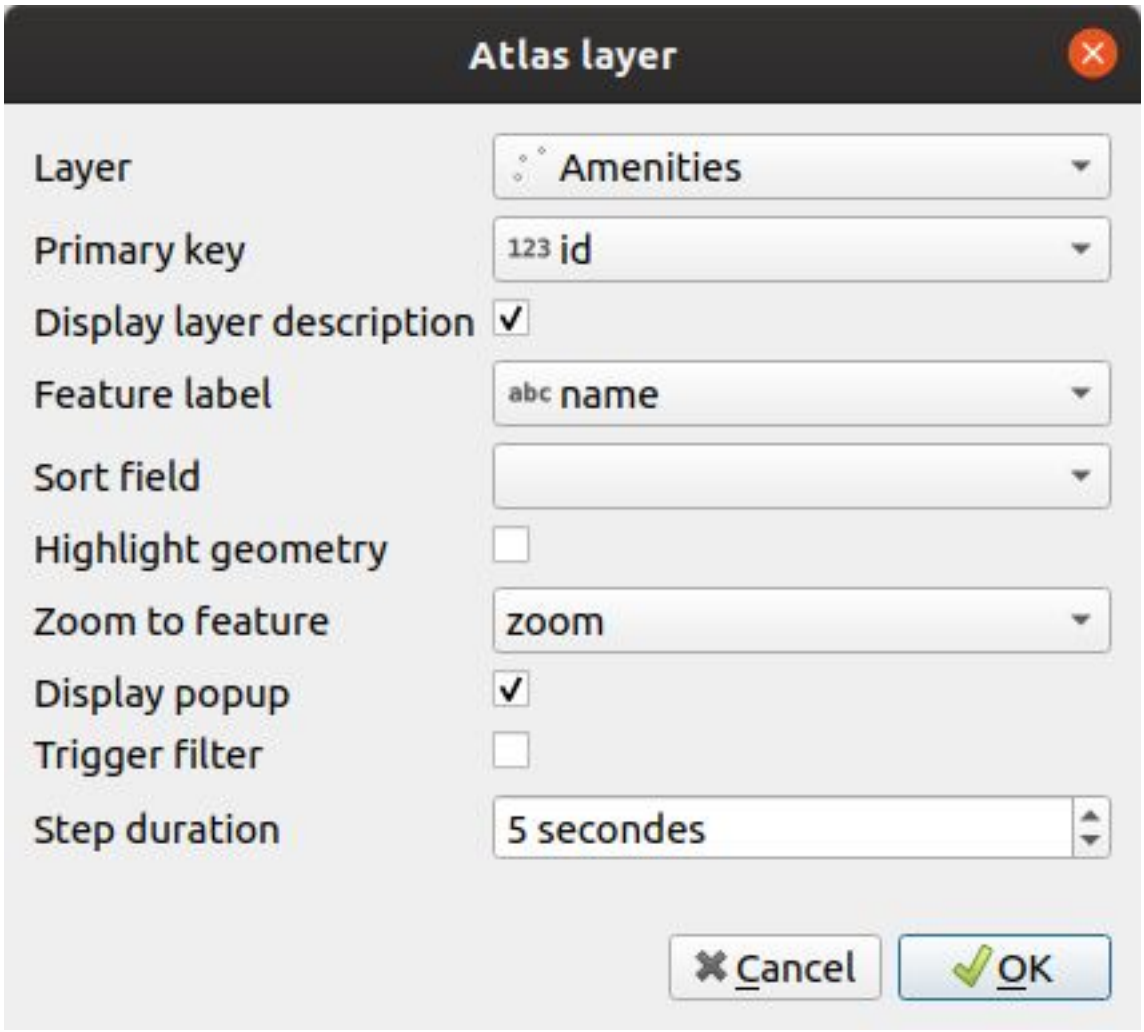
This feature let you chose and configure a layer to make a sequence of entities in your Lizmap project.

Since **Lizmap 3.4** :

Many layers can be configured in this tool. If the checkbox *Auto-play* is checked, the first layer in the list is used.

Configuring the tool

At the layer level

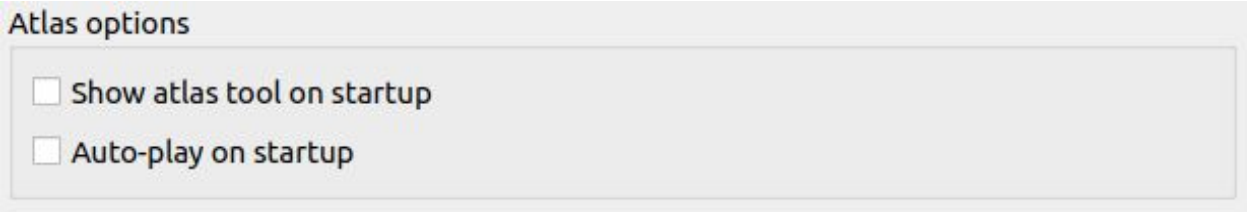


- For setting a atlas layer:
  1. Click the button.
  2. you need to chose the layer you want your atlas on
  3. select the primary key field, it must be an integer
  4. check if you want to display the layer description in the dock of your atlas

5. chose the field who contains the name of your features, it will be shown instead of the primary key in the list of features
6. your atlas will be sorted according to this field
7. you can chose to highlight the feature selected by the atlas, it will change every time it's switching to a new feature
8. chose between a zoom on the feature or to make it the center of your map
9. you can chose to display the popup in the feature in the atlas container or not
10. check if you want to activate filter on the feature selected by the atlas, it will hide all other features of the layer and only show the one selected
11. you can select the duration between each step

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.

At the project level



Options:

- check if you want to open the atlas tool when you open your project
- check if you want to launch the auto-play mode when you open your project

3.2.13 Form filtering - filter layer data based on field values

Contents

- *Form filtering - filter layer data based on field values*
  - *Principle*
  - *Example*
  - *Prerequisites*
  - *Configuring the tool*
  - *Description of the different form input types*
    - *Text*
    - *Date*
    - *Numeric*

    * *Unique values*

Principle

This tool displays a form in the left panel, based on some fields, and allow the users to search among the layer data with a variety of form inputs: combo boxes, checkboxes, text inputs with autocompletion, date selector with sliders between the min and max date, etc.

Warning: It works only with **database** layers: PostgreSQL (recommended), Spatialite and GeoPackage.

Using SQL statements, Lizmap will query the data to retrieve:

- the total count of features for the current filter
- the unique values of some fields (for the Unique Values type for example)
- the minimum and maximum of the numeric fields or date fields
- the extent of the data for the current filter

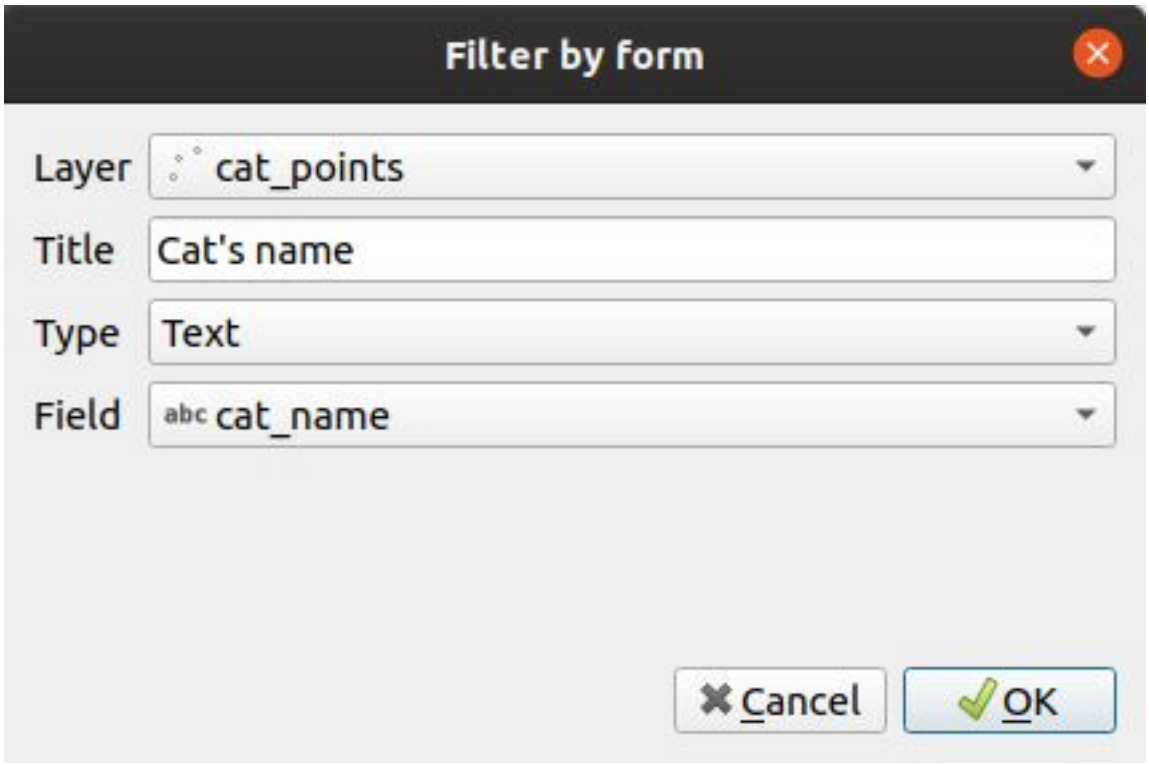
Example

You can see a video with an example: https://vimeo.com/331395259

Prerequisites

The layer must be published as WFS (see *OGC Services Capabilities*) and the primary key must be published as well in the *Layer properties*.

Configuring the tool



There is a new tab in the Lizmap plugin which lets you configure the filter inputs based on the layer fields. You can add one or more fields for one or more layer. If you add fields from 2 or more different layers, Lizmap Web Client will show a combo box to allow the user to choose the layer to filter. Selecting a layer will refresh the form and deactivate the current filter.

- You need to add a line in the plugin table for each field you need to add in the filter form. For each field, you need to configure some options:
  1. Click the button.
  2. **Layer**: the source layer.
  3. **Title**: the title to give to the input, which will be displayed above the form input. For example "Choose a category" for a layer field called "category".
  4. **Type**: the type of the form input, among one of the following: **Text, Unique Values, Date, Numeric**.
  5. **Field**: the field name (in the database table). Only for the Text, Unique Values and Numeric types.
  6. **Min date**: the field containing the start date of your object (ex: "start_date" of an event). This is only needed for the **Date** type. If you have only one date field in your data, you should select it in the Min Date field.
  7. – **Max date**: the field containing the end date of your data. If you have 2 fields containing dates, one for the start date and another for the end date, you can differentiate them. If not, you need to use the same field name for **Min date** and **Max date**.
  8. – **Format**: the format of the **Unique values** type only. It can be **select**, which will show a combo box, or **checkboxes** which will show one checkbox for each distinct value. The distinct values are dynamically queried by Lizmap Web Client.
  9. **Splitter**: for the **Unique values** type only. Use if you want to split the field values by a separator. Ex: `culture, environment` can be split into `culture` and `environment` with the splitter `` `,` ``.

- To edit a layer in the table, select it and click the button or double click on the row.
- To remove a layer from the table, select it and click the button.
- To move a layer up or down, select it and click or buttons. The order will change in Lizmap too.

Description of the different form input types

Text

This is the simplest type. With this type of input, the user will see a classical text input. You can type any text then validate. Autocompletion is enabled by default, which means Lizmap will retrieve the unique values for this field. This could be an option in the future if some datasets are too big and this autocompletion feature is too heavy.

The filter built will be like:

```
"field" LIKE '%foo%'
```

Date

This input type will show a slider with 2 handles to allow to search between the two selected values. The two text inputs are also shown and can be used to manually set the min and max dates.

The date is "truncated" to a date (no time data such as hour, minutes or seconds yet).

The slider step is hard coded and equals to 86400 seconds, which means 1 day.

The filter built will be like:

```
( ( "field_date" >= '2017-04-23' OR  "field_date" >= '2017-04-23' ) AND ( "field_date
↪" <= '2018-06-24' OR  "field_date" <= '2018-06-24' ) )
```

Numeric

This input type will show a slider with 2 handles to allow to search between the two selected values. Two text inputs are also shown and can be used to manually set the min and max values.

The filter built will be like:

```
( ( "field" >= 100 ) AND ( "field_date" <= 200 ) )
```

Unique values

Lizmap will query the data to get the distinct values of the field. You can choose two different input types: **select** or **checkboxes**.

If you have specified a splitter text, for example `,`, Lizmap will find the unique values of the separated text values. For example the value of one feature `culture, environment` will be split into `culture` and `environment` with the splitter `,`. Selecting `culture` or `environment` in the form input will show this feature.

You can choose to show two different input types:

- **Combo box**: this type will show a combo box with the list of distinct values for the field. The user will be able to choose only one item among the values.

- **Checkboxes**: this type will show as many comboboxes as distinct values for the field. The data will be filtered with a `UNION` between checked items.

The filter built will be like:

```
( "field_thematique" LIKE '%Cuisine%'  OR "field_thematique" LIKE '%Ecocitoyen%'  )
```

3.3 Configuration

Settings in Lizmap doesn't occur only in Lizmap QGIS plugin. Some settings are set in the QGIS project itself such as project properties, the legend, forms etc.

3.3.1 Project configuration

Contents

- *Project configuration*
  - *OGC Services Capabilities*
  - *Publish a layer as WFS*

These settings are in *Project* ‣ *Project properties* or `CTRL+SHIFT+P`.

OGC Services Capabilities

In *Project properties* ‣ *QGIS Server* ‣ *Service capabilities*, you can setup some metadata about your project:

- The title which will be used by Lizmap.
- Other information such as organization, the owner of the publication, the abstract, etc

Publish a layer as WFS

For many feature of Lizmap, it's necessary to publish your layer as WFS. It will be require when you are using the Lizmap QGIS plugin to activate some features.

Lizmap Web Client uses the **Web Feature Service** (WFS) to get data from a QGIS vector layer and display it in the web interface. This is why the first thing to do whenever you want to show a layer data in the web client is to **publish the vector layer through the WFS**.

- To do so, open the *Project properties* ‣ *QGIS Server* ‣ *WFS capabilities* and add the layer as "published" by checking the corresponding checkbox and save the project.
- You can also tune the number of decimals to decrease the size of data to be fetched from WFS (keep 8 only for a map published in degrees, and keep 1 for map projections in meters)

3.3.2 Layer configuration

Contents

- *Layer configuration*
  - *QGIS Server tab*
  - *Add a alias on a field*
  - *Customize the edition form*
    - *Advanced form*
  - *Server side simplification*

These settings are in *Layer properties*.

QGIS Server tab

In *Layer Properties* ‣ *QGIS Server*, you can set different settings for QGIS Server :

- *Short name* is a machine readable name for the layer.
- *dataUrl* is the URL to a HTML or PDF presenting the data. It can be a link to the open data portal webpage.

If the link is empty in *Lizmap* ‣ *Layers* dialog, the link in Lizmap will be automatically populated by the Lizmap plugin from set in this tab.

You can use the button in Lizmap to pick this value.

Add a alias on a field

Usually, the field names are defined without accent, spaces and can't be very long. In *Layer Properties* ‣ *Attributes Form* and clicking on a field, you can add an alias.

This will be used to substitute the field name when possible for a more friendly name.

Layer Properties ‣ *Source Fields*, you can the list of alias defined on the layer.

Customize the edition form

In *Layer Properties* ‣ *Attributes Form* and clicking on a field, you can setup the form.

To set the editing tools for your layer fields:

- In *Layer properties* ‣ *Attributes Form*.
- By selecting first a field on left panel, select the *Widget type*:
  - To hide a field, choose *Hidden*. The user will not see the field in the form. There will be no content inserting. *Use it for the primary key*.
  - To add a read-only field, unchecked *Editable* checkbox.
  - **Special case of the option *Value Relation*. You can use this option for a Lizmap map.** For users to have access to information of the outer layer that contains the data, you must enable the publication of the layer as a WFS layer in *Project properties* ‣ *QGIS Server* ‣ *WFS*.

Warning: Lizmap Web Client does not know the "QT Designer UI file" for form generation. Therefore only use the *Autogenerate* mode or *Drag and drop* mode for editing layers.

Note: To make the field compulsory you have to define it as `NOT NULL` in the properties of the table, **in the database**, not in *QGIS –> Layer Properties*.

Note: All the editing tools are not yet managed by Lizmap Web Client. Only the following tools are supported:

- Text edit
- Classification
- Range
- Value Map
- Hidden
- Check Box
- Date/Time
- Value Relation
- Relation Reference

If the tool is not supported, the web form displays a text input field.

Advanced form

Note: To group fields in different tabs, follow QGIS documentation https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/vector_properties.html#the-drag-and-drop-designer.

Lizmap can reproduce several behavior configured in QGIS :

- **Control visibility by expression**. For example, you can toggle tab's visibility based on a checkbox state.

  For that, we can create a field named `has_photo` defined as a `Checkbox` and a `photo` tab having `Control Visibility by Expression` checked and `"has_photo" = true OR "has_photo" = 't'` as Expression.

- **Constraints defined by expression.** For example, you want to simply assert users correctly type a website URL beginning by *http* (of course, regex would be better but we keep it simple).

  For that, we can create a field named `website` defined as a `Text Edit`, define `Constraints` with `left( "website", 4) = 'http'` as expression and `Web site URL must start with 'http'` as Expression description.

- **Filter expression for a Value Relation field.** For example, you want a field to automatically get the related value from another layer's field when drawing a point on the map. For that, we can create a field name `quartier` defined as a Value Relation to a `quartiers` layer and set `intersects($geometry, @current_geometry)` as Filter expression. We can also check `Not null` and `Enforce not null contraint` to assert no NULL value can be set.

Server side simplification

For PostGIS layers, you can enable server side simplification. This in *Layer properties ▸ Rendering* for each layers. You can change the default behavior for next new layer in *QGIS General properties ▸ Rendering*.

3.3.3 Legend

Contents

- *Legend*
  - *Groups*
  - *Theme switcher*
  - *Masking individual layers*
  - *Create an overview map*

Groups

You can create groups in your legend. Lizmap will use them too in the web interface.

If you want to collapse some groups by default, you need to use a JavaScript snippet, see *Adding your own JavaScript*.

Theme switcher

This is a feature in **Lizmap 3.4**.

Lizmap allows you to display and switch between themes configured in QGIS. To create your themes, follow QGIS documentation https://docs.qgis.org/latest/en/docs/user_manual/introduction/general_tools.html#configuring-map-themes.

Masking individual layers

You can exclude layers of your publication with the *QGIS Server* tab of the *QGIS project properties*. In this case the layers will not be available in Lizmap **at all**. With this method, you cannot use a layer in the locate by layer function and not display in the map.

To overcome this lack, Lizmap offers two simple ways to not display some layers only in the legend :

- Either create a group in your legend called `hidden` and put your layer into this group. This group (and its layers) won't be displayed in the Lizmap legend.
- Or use the check box *Hide in legend* in *Lizmap ▸ Layers* for the specific layer.

This feature can be used for:

- hide a layer used in the locate by layer (*Locate by layer*)
- hide a simple layer for adding data rendered with a view
- hide a layer for printing (*Allow printing of external baselayers*)

Create an overview map

To add an **overview map**, or location map, in the Lizmap's map, you must:

- Create an independent group in the QGIS project called `Overview` (with the 1st letter capitalized)
- **Add layers**, for example a layer of municipalities, a lighter terrain base layer, etc.

All layers and groups in the *Overview* group will **not be shown in the Lizmap's map legend**. They are drawn only in the Overview map.

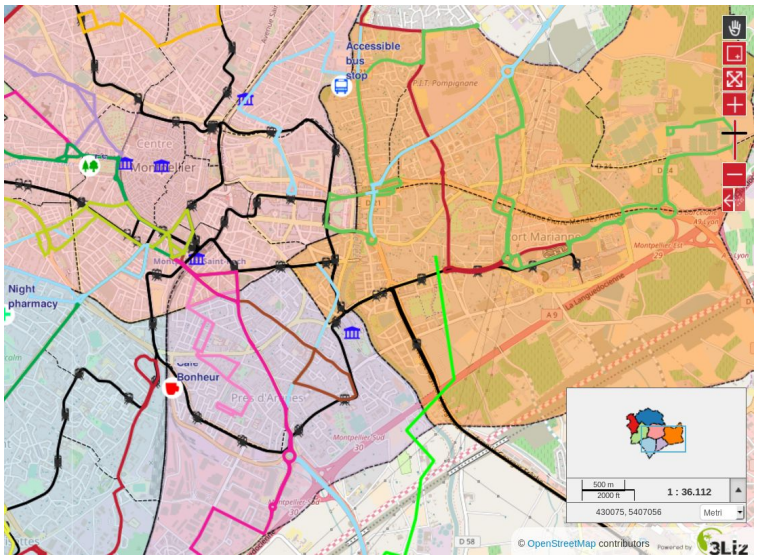
It is advisable to use:

- **light and simplified** (if necessary) vector layers
- use a **suitable symbology**: small strokes and simple or hidden labels

---

Note: The location map will use the extent of the WMS Capabilities, *Project Properties -> QGIS Server -> WMS*.

Here is an example of use:



3.3.4 Popup

Contents

- *Popup*
  - *Activate popup*
  - *Auto popup*
    - *Simple popup configuration*
    - *Mask or rename a column*
    - *Usage of media: images, documents, etc.*
    - *Usage of external links*

- *Lizmap popup*
  - *Introduction*
  - *Deploying*
  - *Usage of media and external links*
- *Form popup*
- *QGIS popup*
- *One to many relations*
  - *Link to a PDF QGIS layout*
  - *Display children in a compact way*

Activate popup

With the plugin, you can activate popups for a **layer** or for a **group** configured with the *Group as layer* option.

In the *Layers* tab, click on the *Popup* checkbox.

For the *Group as layer* option you must select the option for the group and for all the layers included you want to show in the popup: in this case, only the layers with the option *Popup* checked will be shown.

You have three types of popup sources:

- `auto`, read *Auto popup*
- `lizmap`, read *Lizmap popup*
- `qgis`, read *QGIS popup*
- `form`, read *Form popup*

In the web application Lizmap Web Client, a click on a feature will trigger the popup if (and only if):

- the popup **has been activated**
  - through the plugin for the layer or the group
  - or the layer has edition capabilities for existing features
- the layer is **active in the legend**, so that it is shown on the canvas
- the user has clicked on an **area of the canvas** where data for the layer with active popups are displayed.

Note: For point layers you need to click in the middle of the point to display the popup. The tolerance can be setup in *Lizmap plugin ▸ Map options ▸ Map tools*.

You can update where the popup is displayed in the web interface in *Lizmap plugin ▸ Map options ▸ Map interface*. You can choose between:

- `dock`
- `minidock`
- `map`
- `bottomdock`
- `right-dock`

Auto popup

The Lizmap Web Client `auto` popup displays a table showing the columns of the attribute table in two columns *Field* and *Value*, as shown below:

Field	Value
id	1
name	A name
description	This object …
photo	:-)

You can modify the info displayed through QGIS, and also display pictures or links.

Simple popup configuration

With the plugin, if you click on the checkbox **Activate popups** without modifying its content through the button *Configure* the default table is shown.

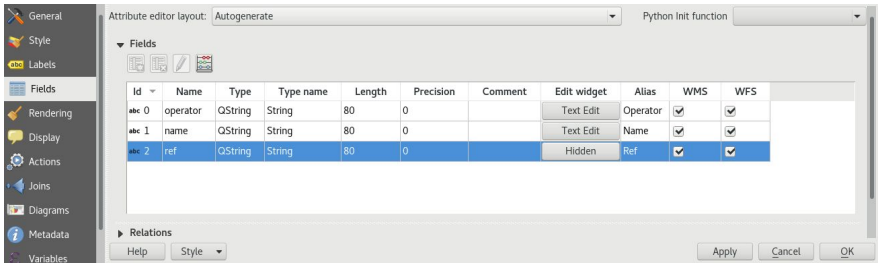
Nevertheless, you can tune several things in QGIS and with the help of Lizmap plugin to **parametrize the fields displayed**, **rename fields**, and even **display images, photos, or links to internal or external documents**.

Mask or rename a column

> **Warning:** This is working only using *Lizmap* or *Auto* popup. The *QGIS* popup follows the QGIS maptip.

You need to use the *Layer Properties* ‣ *Fields* tab in QGIS:

- to **hide** a column in the popup, uncheck the corresponding WMS checkbox. The WMS column is on the right of the table. This will hide the field in QGIS Server from any WMS requests.
- to **rename** a column, you should use the alias column. But to edit this column, add it from the tab *Layer Properties* ‣ *Attributes Form*.



Usage of media: images, documents, etc.

If you want to use some media in your popup (pictures, PDF documents…) in your popup, you must use the *Media* directory.

For a given feature, using a path in your field starting by `media/`, Lizmap will display:

- **the image itself** for jpeg or png files
- **the content** for txt or HTML files
- **a link** to any other file extension

See also:

Chapter *Media* for more details on the usage of documents of the directory media in the popups.

Usage of external links

You can also use, in a field, **full web links to a specific page or image**:

- the image referred to will be displayed, instead of the links
- the web link will be displayed and clickable

Lizmap popup

Introduction

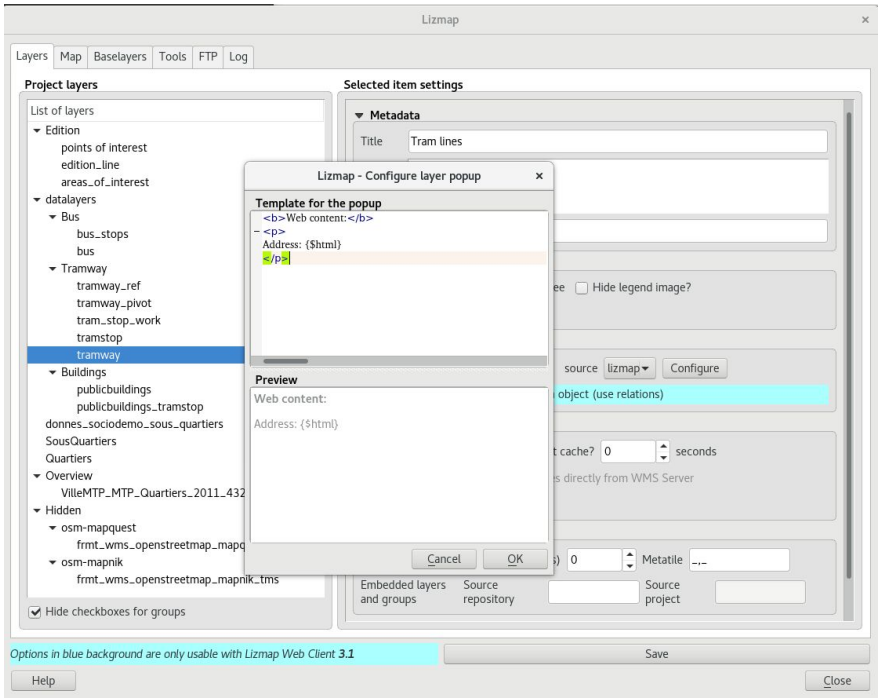
If the simple table display does not suit your needs, you can write a **popup template**. To do so, you should know some **HTML format**. See e.g.: https://www.w3schools.com/html/

> **Warning:** When you use the *lizmap* mode, the previous configuration to rename a field does not work anymore: you have to configure what is displayed and how through the template. Managing media is also possible, but you have to configure it as well.

Deploying

You can edit the popup template with the button *Configure* in the Lizmap plugin. Clicking on it you'll get a window with two text areas:

- an **area where you can type your text**
- a **read-only area**, showing a preview of your template



You can type simple text, but we suggest to write in HTML format to give proper formatting. For instance, you can add paragraphs, headings, etc.:

```
<h3>A Title</h3>
<p>An example of paragraph</p>
```

The behaviour is as follows:

- if the content of the two areas is empty, a simple table will be shown in the popup (default template)
- if the content is not empty, its content will be used as a template for the popup

Lizmap Web Client will replace automatically a variable, identified by the name of a field, with its content. To add the content of a column to a popup, you should use the name of the column precede by a dollar sign (*$*), all surrounded by curly brackets (*{}*). For instance:

```
<h3>A Title</h3>
<p>An example of paragraph</p>
<p>A name: <b>{$name}</b></p>
<p>Description: {$description}</p>
```

Note: If you have configured an alias for a field, you have to use the alias instead of the name, between the brackets.

You can also use the values of the columns as parameters to give styling to the text. An example here, to use the colour of a bus line as a background colour:

```
<p style="background-color:{$color}">
<b>LINE</b> : {$ref} - {$name}
<p/>
```

Usage of media and external links

You can **use the media** referred to in the table content, even if you use a *template model*. To do this, you should use the media column, taking into account the fact that Lizmap Web Client automatically replaces the relative path of the type `/media/myfile.jpg` with the full URL to the file, accessible through the web interface.

You can also use full URLs pointing to the pages or images on another server.

Here an example of a template handling media and an external link:

```
 <p style="font-size:0.8em;">A Title</p>
 <p>The name is {$name}</p>
<p>
   A sample image<br/>
   <img src="{$image_column}" style="">
 </p>

 <p><a href="{$website}" target="_blank">Web link</a></p>

 <p><img src="https://www.3liz.com/images/logo-lizmap.png"/></p>
```

See also:

Chapter *Media* for more details on the use of documents in the directory media.

Form popup

If you have defined a form layout with the *Drag and drop form designer* for a layer in *Layer Properties ▸ Attributes Form*, you can also display it in its popup. See the QGIS documentation about Drag and Drop form.

The QGIS Form, designed for HTML, is displayed in the popup directly. You can't alter the popup on runtime.

Tabs defined in the *Form Layout* in QGIS will also be displayed as HTML tabs in the popup.

If you want to alter the popup HTML, then you must use *QGIS Popup* instead of *Form Popup*.

QGIS popup

QGIS popups can be configured via *QGIS ▸ Layer properties ▸ Display ▸ HTML Map Tip*. The main advantages of this approach are:

- HTML is used
- you can use QGIS variables and expressions, thus adding information created dynamically
- the popup can be previewed in QGIS, using map tips. You can enable map tips in the menu *View ▸ Show Map Tips*
- the popup configurations are stored in QGIS project and layer style, so they can be reused in other Lizmap projects without replicating the configuration.

Similar to *Form popup*, you have a *Drag and drop form designer* for a layer, you can click on the *Copy the drag and drop designer* button. This will **erase** any existing maptip set on the layer and will generate the QGIS Expression matching the form layout.

To have a similar popup as the **auto** one, you need to use :

```
<table class="table table-condensed table-striped table-bordered lizmapPopupTable">
  <thead>
    <tr>
      <th>Field</th>
      <th>Valeur</th>
    </tr>
  </thead>
  <tbody>
      <tr>
        <th>NAME OF THE FIELD</th>
        <td>VALUE OF FIELD USING EXPRESSION</td>
      </tr>
      <tr>
        <th>NAME OF THE FIELD</th>
        <td>VALUE OF FIELD USING EXPRESSION</td>
      </tr>
  </tbody>
</table>
```

One to many relations

It is possible to display multiple objects (photos, documents) for each geographical feature. To do so, you have to configure both the QGIS project and the Lizmap config.

In QGIS project:

- Use 2 separate layers to store the main features and the pictures. For example *trees* and *tree_pictures*. The child layer must contain a field referencing the parent layer id
- Configure aliases and field types in tab Fields of the layers properties dialog. Use *Photo* for the field which will contains the relative path to pictures
- Add a relation in QGIS project properties between the main layer *trees* and the child layer *tree_pictures*
- Add data to the layers. You should use relative path to store the pictures path. Theses paths must refer to a project media subdirectory, for example: *media/photos/feature_1_a.jpg*

In Lizmap plugin:

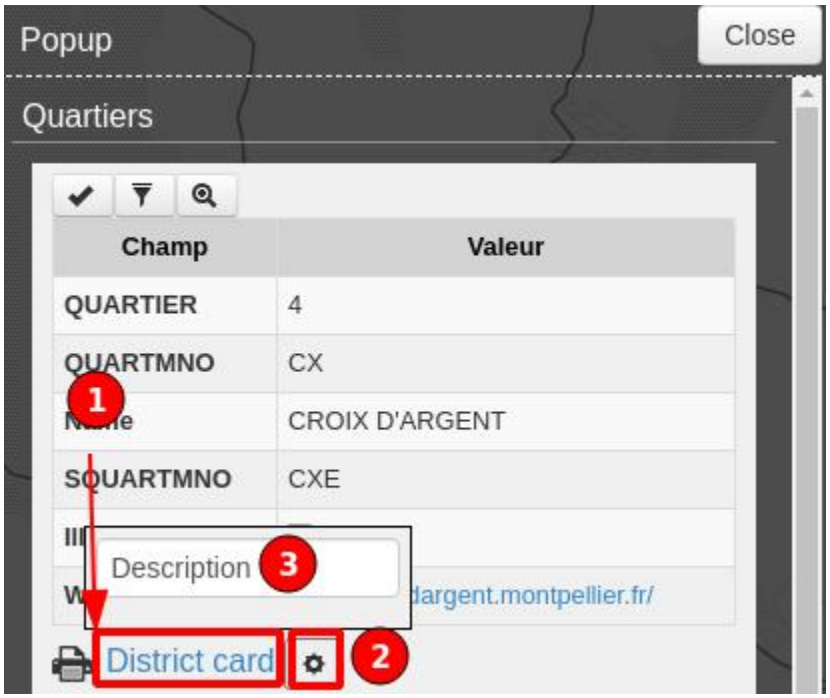
- In the *Layers* tab, activate popup for both layers. You can configure popup if you need specific layouts (See documentation on popups)
- For the parent layer, activate the option *Display relative children under each object (use relations)*
- Add the two layers in the *Attribute table* tab
- You can optionally activate editing for the two layers, to allow the web users to create new features and upload pictures
- Save and publish your project and Lizmap configuration

In Lizmap Web Client:

If relative children popup are defined as `auto`, this button will be visible in the feature's popup at the top of related objects. Click it to compact all related objects in one table with search, sort and paging capabilities.

Link to a PDF QGIS layout

Every feature of a layer with an atlas configured will have a link (1) at the end of its popup which open a PDF for this specific feature, using the QGIS Atlas layout. If the layout contains custom text fields, a button (2) will be displayed. Clicking this button, allows you to type values for those custom text fields before printing. To enable this feature, you need a QGIS Layout with atlas enabled on that layer **and** to download the *AtlasPrint* QGIS Server plugin on GitHub : https://github.com/3liz/qgis-atlasprint



Display children in a compact way

You can change the way children are displayed and make them look like a table. For that, you will need to adapt the HTML of your children layer and use a few classes to manipulate it.

- "lizmap_merged" : You need to attribute this class to your table
- lizmapPopupHeader : If you want to have a better display of your headers, you will need to put this class in the '<tr>' who contains them
- lizmapPopupHidden : This class permit you to hide some elements of your children that you want to hide when there are used as a child but you still want to see them if you display their popup as a main Popup

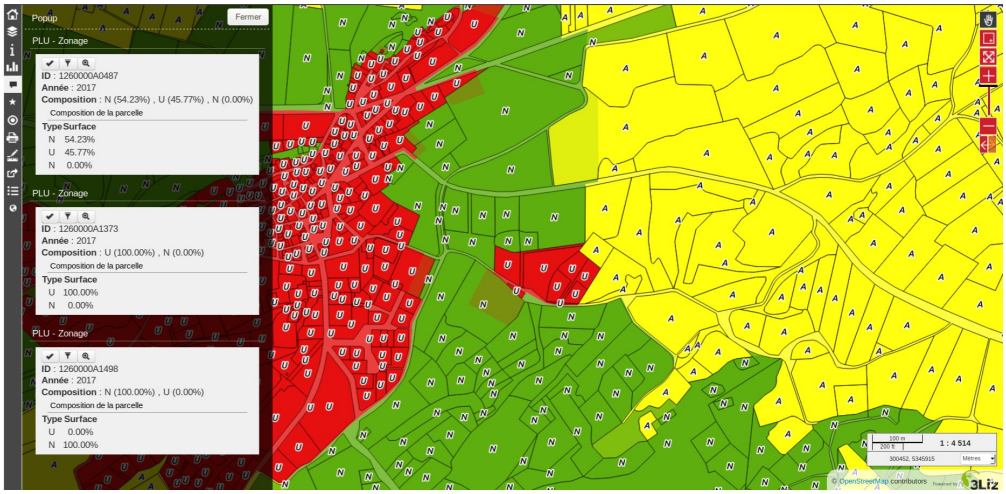
Here an example:

```
<table class="lizmap_merged">
 <tr class="lizmapPopupHeader">
     <th class="lizmapPopupHidden"><center> Idu </center></th>
     <th> <center> Type </center> </th>
     <th> <center> Surface</center> </th>
  </tr>
  <tr>
     <td class="lizmapPopupHidden"><center>[% "idu" %]</center></td>
     <td><center>[% "typezone" %]</center></td>
```

(continues on next page)

(continued from previous page)

```
      <td><center>[% "surface" %]</center></td>
  </tr>
</table>
```



3.3.5 Action in a popup

Contents

- *Action in a popup*
  - *Principle*
  - *Configuring the tool*

This is a feature in **Lizmap 3.4**.

Principle

This module allows to add one or several **action buttons** in the **Lizmap popup** displayed for a **PostgreSQL** object, which will **trigger a query** in the database and return a **geometry** to display on the map.

It reads a **JSON configuration file** which must be placed **aside the QGIS project** in the same directory. This file lists the **PostgreSQL actions** to be added in the **popup** for one or many QGIS PostgreSQL vector layers.

Configuring the tool

- Each action is **characterized** by a `layer id`, a `name`, a `title`, an `icon`, some optional `options`, `style` and `callbacks`. A new `confirm` property can be used since **Lizmap 3.5**
- A layer can have **one or several actions**
- You can have **one or several layers** with their own actions

Example of this JSON configuration file, name `myproject.qgs.action` if the QGIS project file is named `myproject.qgs`. In this project, there is a vector layer called `Points` with the internal layer ID `points_a7e8943b_7138_4788_a775_f94cbd0ad8b6` (you can get the QGIS layer internal ID with the expression `@layer_id`)

```
{
    "points_a7e8943b_7138_4788_a775_f94cbd0ad8b6": [
        {
            "name": "buffer_500",
            "title": "Buffer 500m around this object",
            "confirm": "Do you really want to show the buffer ?",
            "icon": "icon-leaf",
            "options": {
                "buffer_size": 500,
                "other_param": "yes"
            },
            "style": {
                "graphicName": "circle",
                "pointRadius": 6,
                "fill": true,
                "fillColor": "lightblue",
                "fillOpacity": 0.3,
                "stroke": true,
                "strokeWidth": 4,
                "strokeColor": "blue",
                "strokeOpacity": 0.8
            },
            "callbacks": [
                {"method": "zoom"},
                {"method": "select", "layerId": "bati_1a016229_287a_4b5e_a4f7_
↪a2080333f440"},
                {"method": "redraw", "layerId": "bati_1a016229_287a_4b5e_a4f7_
↪a2080333f440"}
            ]
        }
    ]
}
```

The **JSON configuration** file lists the **QGIS layers** for which you want to declare actions. Each layer is defined by its **QGIS layer ID**, for example here `points_a7e8943b_7138_4788_a775_f94cbd0ad8b6`, and for each ID, a list of objects describing the actions to allow.

Each **action** is an object defined by:

- a `name` which is the action identifier.
- a `title` which is used as a label in Lizmap interface
- an `icon` which is displayed on the action button (See https://getbootstrap.com/2.3.2/base-css.html#icons)
- an optional `confirm` property, since **Lizmap 3.5**, containing some text. If set, a confirmation dialog will be shown to the user to ask if the action should really be launched or not. Use it if the action can modify some data in

your database.

- an `options` object, giving some additional parameters for this action. You can add any needed parameter.
- a `style` object allowing to configure the returned geometry style. It follows OpenLayers styling attributes.
- a `callbacks` object allows to trigger some actions after the generated geometry is returned. They are defined by a `method` name, which can at present be:
  - `zoom`: zoom to the returned geometry
  - `select`: select the features from a given layer intersecting the returned geometry. The target layer **QGIS internal ID** must be added in the `layerId` property. In the example, the features of the layer containing buildings, ID `bati_1a016229_287a_4b5e_a4f7_a2080333f440` will be selected
  - `redraw`: redraw (refresh) a given layer in the map. The target layer QGIS ID must be added in the `layerId` property.

Lizmap detects the presence of this configuration file, and adds the needed logic when the map loads. When the users clicks on an object of one of this layer in the map, the **popup panel** shows the feature data. At the top of each popup item, **a toolbar will show one button per each layer action**. The action `title` will be displayed on hovering the action button.

Each button **triggers the corresponding action**, if it is not yet **active** (else it deactivates and erases the geometry):

- Lizmap backend checks if the action is well configured,
- creates the **PostgreSQL query** and execute it in the layer PostgreSQL database. (See example below)
- This query returns a **GeoJSON** which is then displayed on the map.
- If some **callbacks** have been configured, they are launched
- Since **Lizmap 3.5**, A Lizmap **event** `actionResultReceived` is emitted with the returned data and action properties.

The **created PostgreSQL query** is built up by Lizmap web client and uses the PostgreSQL function `lizmap_get_data(json)` which **must be created beforehand** in the PostgreSQL table database. This function also uses a more generic function `query_to_geojson(text)` which transforms any PostgreSQL **query string** into a **GeoJSON output**.

Here is **an example** below of the query executed in the PostgreSQL database by Lizmap Web Client internally, for the example configuration given above, when the users clicks on the button action *buffer_500*, for the **feature** with id `1` of the **layer** `Points` corresponding to the **PostgreSQL table** `test.points`:

```
SELECT public.lizmap_get_data('{
    "layer_name":"points",
    "layer_schema":"test",
    "layer_table":"points",
    "feature_id":1,
    "action_name":"buffer_500",
    "buffer_size":500,
    "other_param": "yes"
}') AS data;
```

You can see that Lizmap creates a JSON parameters with all needed information and run the PostgreSQL function `lizmap_get_data(text)`.

You need to create this PostgreSQL function `lizmap_get_data(text)` which returns a valid GeoJSON text with one single object in it. The following SQL code is **an example** to help you create the needed functions. Obviously, **you must adapt it to fit your needs**.

```
-- Returns a valid GeoJSON from any query
CREATE OR REPLACE FUNCTION query_to_geojson(datasource text)
RETURNS json AS
$$
DECLARE
    sqltext text;
    ajson json;
BEGIN
    sqltext:= format('
        SELECT jsonb_build_object(
            ''type'',  ''FeatureCollection'',
            ''features'', jsonb_agg(features.feature)
        )::json
        FROM (
          SELECT jsonb_build_object(
            ''type'',       ''Feature'',
            ''id'',         id,
            ''geometry'',   ST_AsGeoJSON(ST_Transform(geom, 4326))::jsonb,
            ''properties'', to_jsonb(inputs) - ''geom''
          ) AS feature
          FROM (
              SELECT * FROM (%s) foo
          ) AS inputs
        ) AS features
    ', datasource);
    RAISE NOTICE 'SQL = %s', sqltext;
    EXECUTE sqltext INTO ajson;
    RETURN ajson;
END;
$$
LANGUAGE 'plpgsql'
IMMUTABLE STRICT;

COMMENT ON FUNCTION query_to_geojson(text) IS 'Generate a valid GEOJSON from a given␣
↪SQL text query.';

-- Create a query depending on the action, layer and feature and returns a GeoJSON.
CREATE OR REPLACE FUNCTION lizmap_get_data(parameters json)
RETURNS json AS
$$
DECLARE
    feature_id integer;
    layer_name text;
    layer_table text;
    layer_schema text;
    action_name text;
    sqltext text;
    datasource text;
    ajson json;
BEGIN

    action_name:= parameters->>'action_name';
    feature_id:= (parameters->>'feature_id')::integer;
    layer_name:= parameters->>'layer_name';
    layer_schema:= parameters->>'layer_schema';
    layer_table:= parameters->>'layer_table';
```

(continues on next page)

(continued from previous page)

```
    -- Action buffer_500
    -- Written here as an example
    -- Performs a buffer on the geometry
    IF action_name = 'buffer_500' THEN
        datasource:= format('
            SELECT
            %1$s AS id,
            ''The buffer '' || %4$s || ''m has been displayed in the map'' AS message,
            ST_Buffer(geom, %4$s) AS geom
            FROM "%2$s"."%3$s"
            WHERE id = %1$s
        ',
        feature_id,
        layer_schema,
        layer_table,
        parameters->>'buffer_size'
        );
    ELSE
    -- Default : return geometry
        datasource:= format('
            SELECT
            %1$s AS id,
            ''The geometry of the object have been displayed in the map'' AS message
            geom
            FROM "%2$s"."%3$s"
            WHERE id = %1$s
        ',
        feature_id,
        layer_schema,
        layer_table
        );

    END IF;

    SELECT query_to_geojson(datasource)
    INTO ajson
    ;
    RETURN ajson;
END;
$$
LANGUAGE 'plpgsql'
IMMUTABLE STRICT;

COMMENT ON FUNCTION lizmap_get_data(json) IS 'Generate a valid GeoJSON from an action␣
↪described by a name, PostgreSQL schema and table name of the source data, a QGIS␣
↪layer name, a feature id and additional options.';
```

- The function `lizmap_get_data(json)` is provided here as an example. Since it is the **key entry point**, you need to adapt it to fit your needs. It aims to **create a query for each action name**, dynamically created for the given parameters, and return a GeoJSON representation of the query result data. You should have **only one feature** returned: use aggregation if needed. In the example above, we use the `format` method to set the query text, and the function `query_to_geojson` to return the GeoJSON for this query.
- You can use all the given parameters (action name, source data schema and table name, feature id, QGIS layer name) to create the appropriate query for your action(s), by using the PostgreSQL `IF THEN ELSIF ELSE` clauses. See the content of the `parameters` variable in the example above, containing some of the JSON configuration file properties, and some properties of the QGIS layer:

  - the **action name** `action_name`, for example `buffer_500`. You should use a simple word with only letters, digits and _,
  - QGIS **layer name** (as in QGIS legend): `layer_name`, for example `Points`,
  - the PostgreSQL table **schema** `layer_schema` and **table name** `layer_table` for this layer,
  - the object **feature id** `feature_id`, which corresponds to the value of the **primary key** field for the popup object,
  - the other **properties** given in the JSON configuration file, in the `options` property, such as `buffer_size` which is `500` in the example

- The `IF ELSE` is used to do a different query, built in the `datasource` variable, by checking the **action name**
- If the return data contains a `message` field, such as shown in the example above, the text contained in this field will be **displayed in the map** in a message bubble.
- The **geometry** returned by the function **will be displayed on the map**.
- You could use your function to **edit some data** in your database, before returning a GeoJSON. To do so, you need to replace the `IMMUTABLE` property par `VOLATILE`. Please **use it with care** !

Since Lizmap Web Client **triggers an event** `actionResultReceived` any time the user clicks on an action button, and data is returned (in the same time as the result geometry is drawn on the map), you could use your own Javascript code to add some logic after the result is shown.

See also:

Chapter *Adding your own JavaScript*

For example, here we just write in the browser console the content received:

```
lizMap.events.on({

    actionResultReceived: function(e) {
        // QGIS Layer id
        var layerId = e.layerId;
        console.log('Layer ID = ' + layerId);
        // Feature ID, which means the value of the primary key field
        var featureId = e.featureId;
        console.log('Feature ID = ' + featureId);
        // Action item with its name and other properties: name, title, options,␣
↪styles, etc.
        var action = e.action;
        console.log('Action properties = ');
        console.log(action);
        // Features returned by the action
        var features = e.features;
        console.log('Returned object = ');
        console.log(features);
    }
});
```

You could use these data as you like in your JS code.

3.3.6 Media

Contents

- *Media*
  - *Principle*
  - *Use for links*
  - *Use a single media folder for many Lizmap folders*
  - *Use in popups*
    - *Principle*
    - *Example*
    - *Result*
    - *Illustration*

Principle

It is possible to provide documents through Lizmap. To do this, you simply:

- Create a directory called `media` (in lower case and without accents) *at the same level as the QGIS project*
- Put documents in it : pictures, reports, PDFs, videos, HTML or text files
- You can use subdirectories per layer or theme: the organization of `media` directory content is free.

Lizmap is using the directory for other purposes:

- Adding some javascript, `media/js`, see *Adding your own JavaScript*.
- Replacing the default theme, `media/theme`, see *Creating simple themes*.
- In the *Edition* in `media/upload/layer_name` for pictures uploaded from users.
- Providing documents

Then in Lizmap Web Client you can provide access to these documents for 2 things:

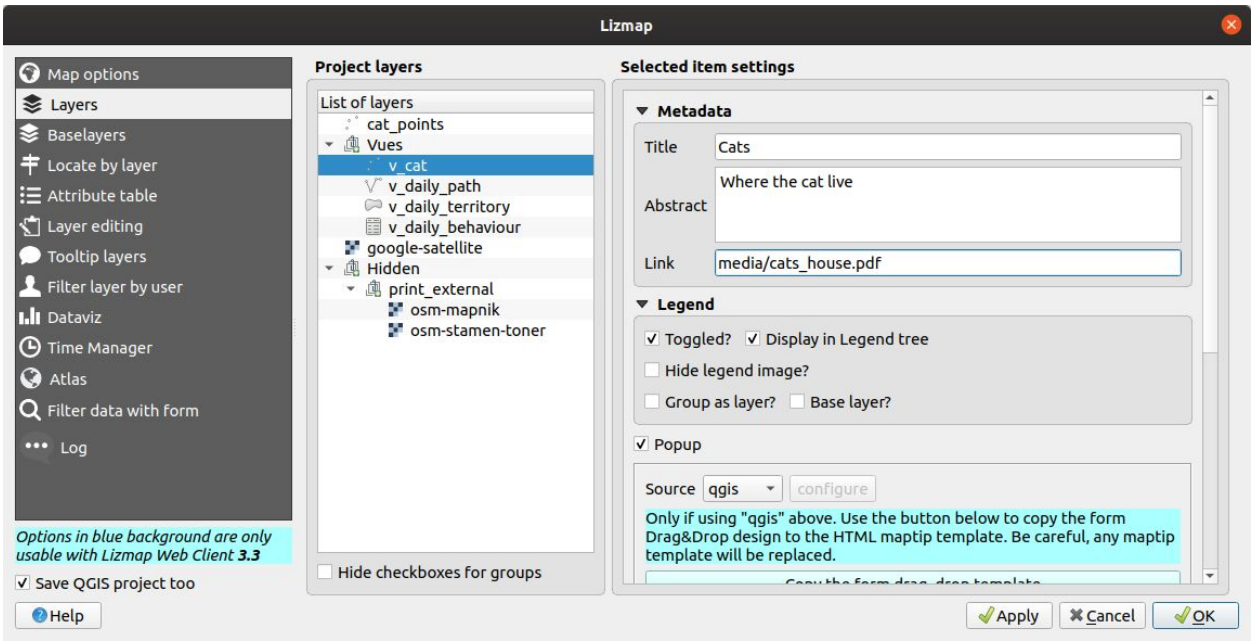
- the **popups**: the content of one or more field for each geometry can specify the path to the media. For example a *photo* or *pdf* field. See *Popup*.
- the **link** provided for each group or layer in the Lizmap plugin *Layers* tab.

Details of these uses is specified below.

Warning: Check your file permissions on the `media` folder. If the folder is not readable, an error will occur.

Use for links

It is possible to use a relative path to a document for layers or groups link. Links can be filled with the Lizmap plugin *Layers* tab after selecting the layer or group. See *Configure your layers for the Web*



The path should be written:

- Starting with `media/`
- Or with `../media/` if you want to use a single **media** folder, read *Use a single media folder for many Lizmap folders*
- With slashes `/` and not backslashes `\`

Some examples:

- `media/my_layer/metadata_layer.pdf`
- `media/reports/my_report_on_the_layer.odt`
- `media/a_picture.png`

On the Lizmap Web Client map, if a link has been set up this way for one of the layers, then an icon (i) will be placed to the right of the layer. Clicking this icon opens the linked document in a new browser tab.

Warning: For editing capabilities, in *Layer properties* ‣ *Attributes Form*, for the given field, you should choose `Attachment` widget and set the path relative to the project.

If the media is a picture, you should set the *Integrated Document Viewer* to `Image`.

Use a single media folder for many Lizmap folders

It's possible to use a single `media` folder located in the root repository of the Lizmap installation. As the folder is located in the parent folder of the QGIS project, it's allowed to use `../media` in the QGIS project, for instance in the attribute table of a layer.

Use in popups

Principle

As described in the introduction above, you can use **a media path** in the spatial data layer.

For example, if you want that the popups associated with a layer displayed a picture that depends on each object, just create a new field that will contain the media path to the picture in each row of the layer attribute table, then activate popups for this layer.

Example

Here for example the attribute table of a layer *landscape* configured to display pictures in the popup. The user has created a `picture` field in which he places the path to the pictures and a `pdf` field in which he puts the paths to a pdf file describing the object corresponding to each line.

id	name	description	picture	pdf
1	Marsh	blabla	media/photos/photo_1.png	media/docs/paysage-1.pdf
2	Beach	blibli	media/photos/photo_2.png	media/docs/paysage-2.pdf
3	Moor	bloblo	media/photos/photo_3.png	media/docs/paysage-3.pdf

Note: In this example, we see that the pictures and PDF file names are normalized. Please follow this example because it allows using the QGIS Field Calculator to create or update automatically the media column data for the entire layer.

Result

Here are the display rules in the popup:

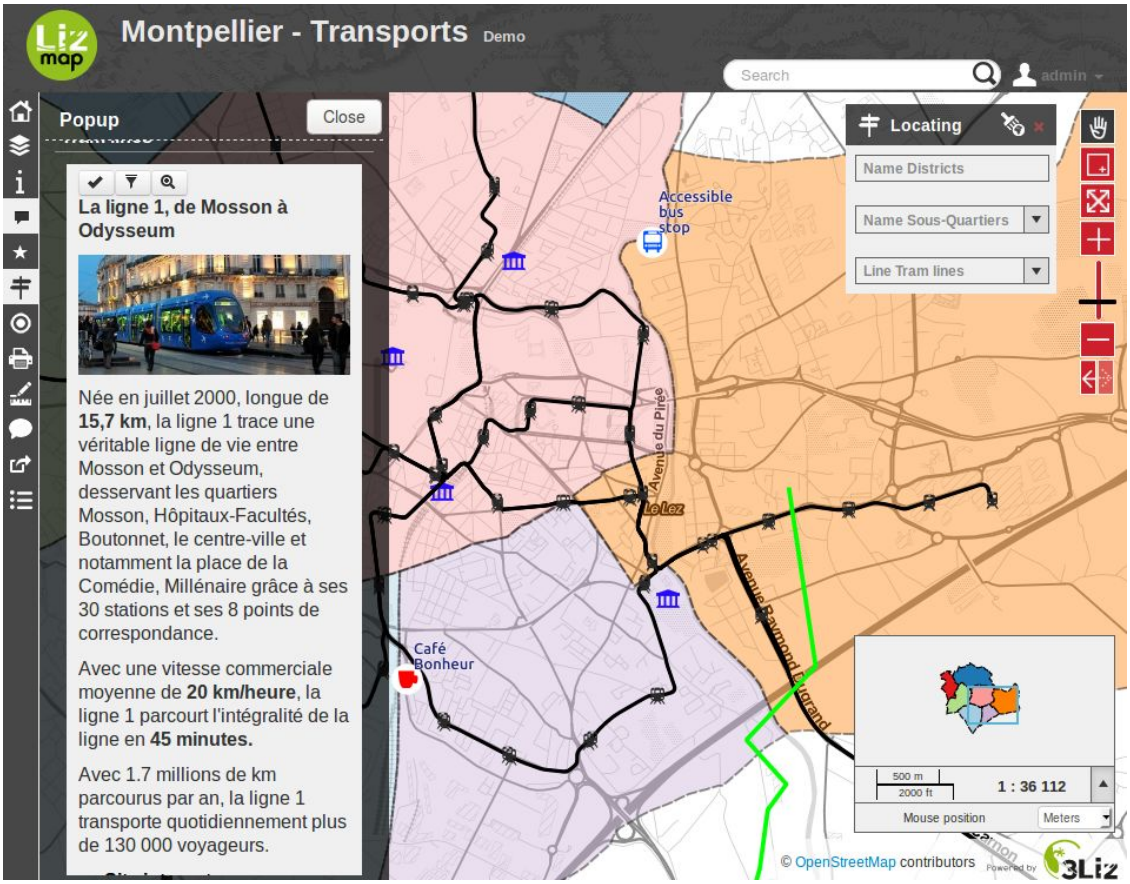
- If you are using a *auto* popup:
  - If the path points to a picture, the image will be displayed in the popup. Clicking on the picture will display the original image in a new tab.
  - If the path points to a text file or HTML file, the file contents will be displayed in the popup.
  - For any other file types, the popup will display a link to the document that users can download by clicking on the link.
- If you are using a *lizmap* popup, `${name_of_the_field}` will have the full URL to the media, starting by `http`. This needs to be encapsulated to some HTML, such as `<img />` or `<a href></a>`.
- Then if your are using a *qgis* popup:
  - `[% "name_of_the_field" %]` will return only the value of the field, like `media/test.pdf`.
  - So for links, you need to use HTML, such as `<a href="[% "name_of_field" %]">Link</a>`.

- And for images, you need `<img>` (with an optional link to open it fullscreen) such as

```
<a href="[% "name_of_field" %]" target="_blank">
    <img src="[% "name_of_field" %]" border="0">
</a>
```

Illustration

Below is an illustration of a Lizmap popup displaying a picture, a text and a link in the popup:



3.3.7 QGIS Expression

As in QGIS Desktop, Lizmap can use QGIS expression .

Project level

At the project level, Lizmap will set two variables in QGIS Server:

- `@lizmap_user` : *string*, the current user connected in Lizmap. It might be empty.
- `@lizmap_user_groups` : *array*, the current user groups as a list. It might be empty.

Note: It's possible use these variables in a symbology for instance, to set a different color for a given user, or to print the current Lizmap user in a layout.

Advanced forms

For editing, it's possible to use expressions for constraints, default value, group visibility. Read *Advanced form*.

Layouts

It's possible to use expressions in layouts, for dates, path of a Lizmap media.

In a layout, the expression `map_credits` or more complex expressions are useful.

3.3.8 Printing

Contents

- *Printing*
  - *Extent defined by the user on the fly in Lizmap*
    - *Creating the layout*
    - *Dynamic content*
    - *Scales*
    - *Excluding a layout*
  - *Layout with an atlas when using a popup*
  - *Allow printing of external baselayers*
  - *Adding your own images in a layout*

Extent defined by the user on the fly in Lizmap

To add print capabilities in the online map, you have to enable the printing tool in the plugin *Map* tab (*Map options*) and the QGIS project needs at least one print layout without atlas enabled.

Creating the layout

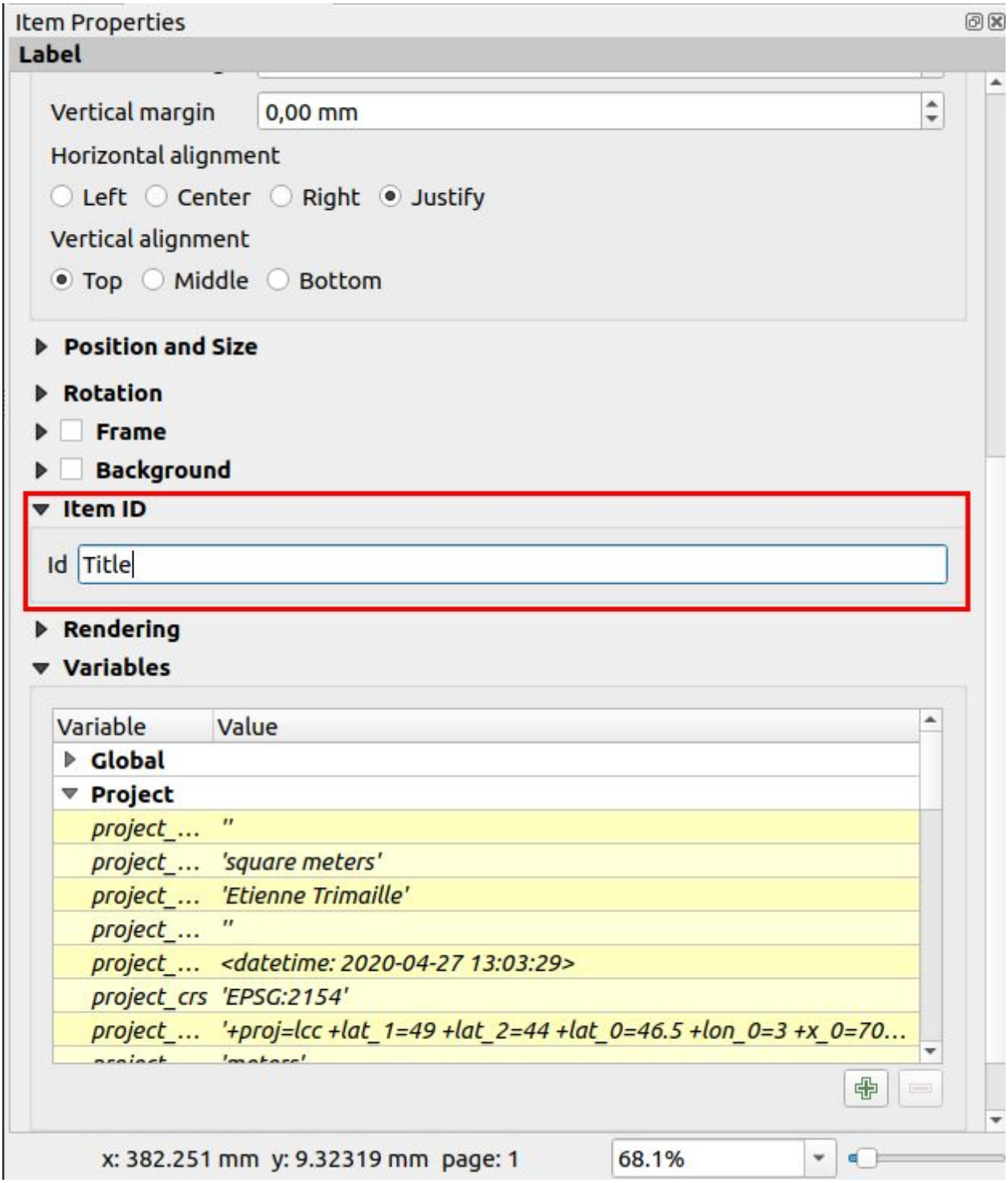
In your layout, you can add :

- A map, **without** an atlas enabled
- An image to North arrow
- An image for the logo of your organization
- A legend that will be fixed for all printing
- Labels
- **A scale**
  - Either *Numeric*
  - Or set *Fit segment width* with a correct reference anchor point to adjust the position of the scale bar
- A location map, a map for which you have enabled and configured the function of *Overview*, read *Create an overview map*
- Since **QGIS 3**, you can use QGIS expressions, in your labels for instance. You can create automatic source label according to visible layers following this example https://docs.qgis.org/latest/en/docs/user_manual/print_composer/composer_items/composer_label.html#id4 on the QGIS Documentation.

Dynamic content

You can allow the user to modify the contents of certain labels (title, description, comment, etc). To do this, you need to add an identifier to your label in the composer.

> **Warning:** Each identifier must be unique for the whole composer. Identifiers must be strings **not** integers (e.g. 'my_title' **not** '1').

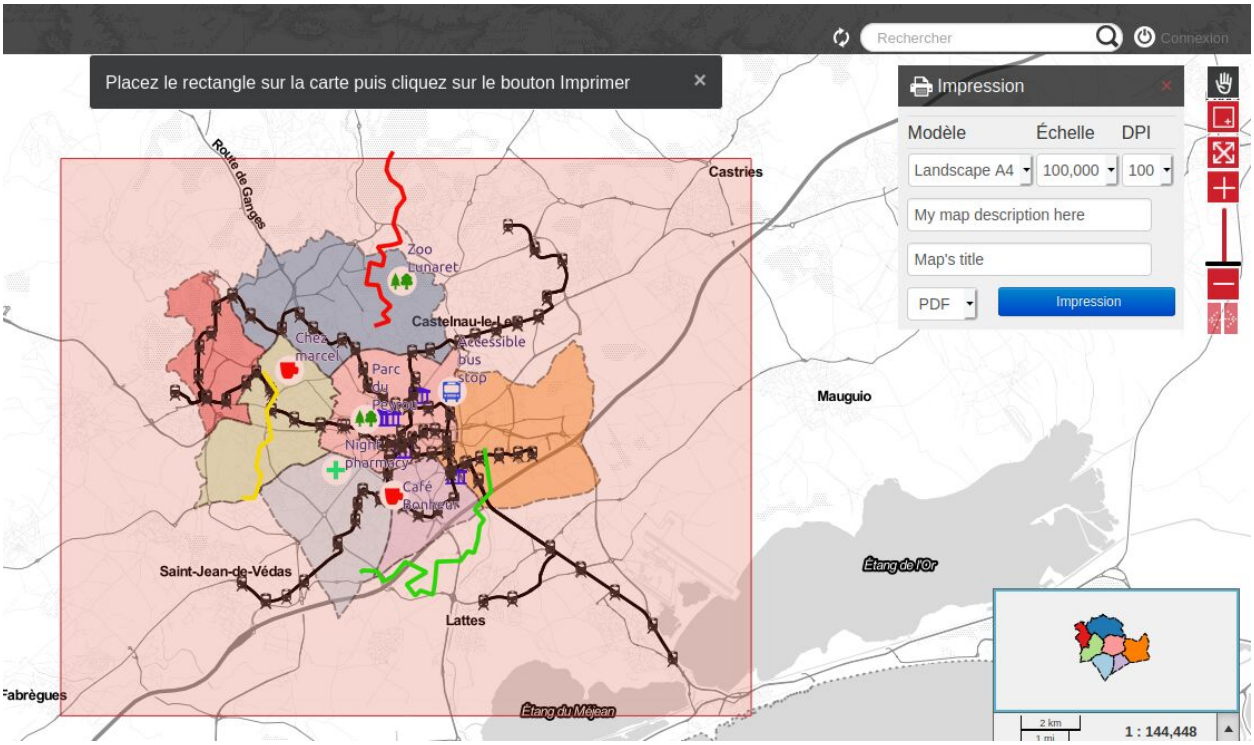


Lizmap will automatically ask the user in the webbrowser to fill each fields.

More tips:

- If your label is pre-populated in QGIS, the field will be pre-populated too in the webbrowser.
- If you check 'Render as HTML' for your label in QGIS, you will have a multiline label in Lizmap accepting HTML code. Use `<br>` for line breaks for example.

The preview in Lizmap will be similar to this screenshot. The red rectangle is the area that the user can define in the webbrowser and the user can also set the map description and the map title.

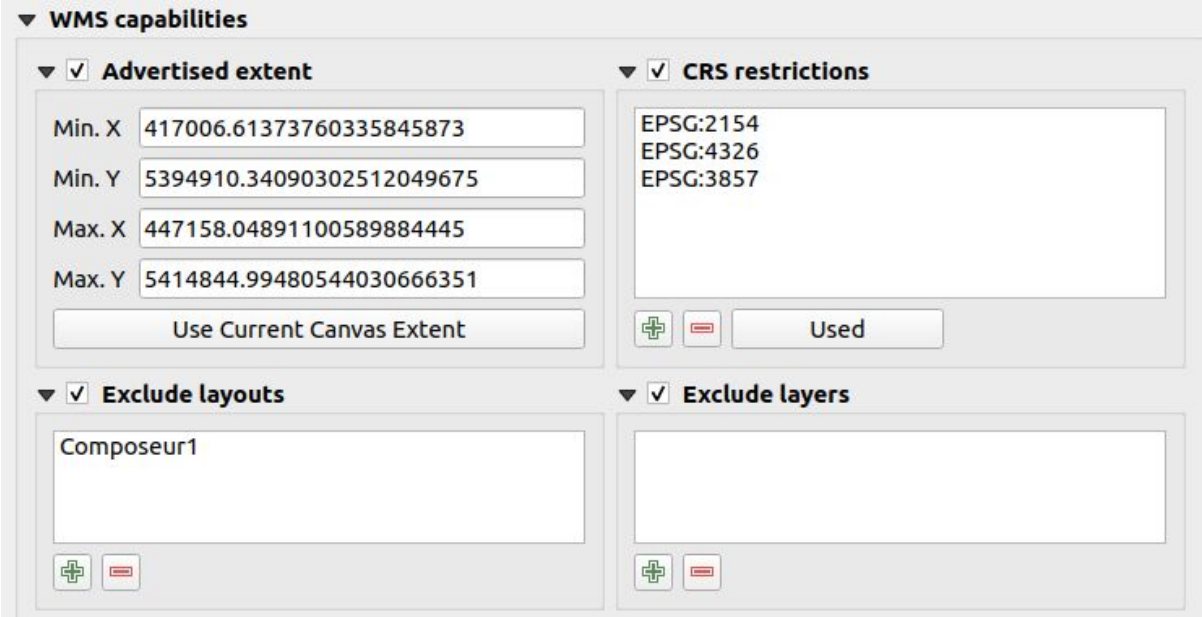


Scales

The print function will be based on the map scales that you set in the plugin *Map* (*Map options*).

Excluding a layout

It is possible to exclude printing compositions for the web. For example, if the QGIS project contains 4 compositions, the project administrator can exclude 2 compositions in the *Project properties* ▸ *QGIS server*. Only the published compositions will be presented in Lizmap.

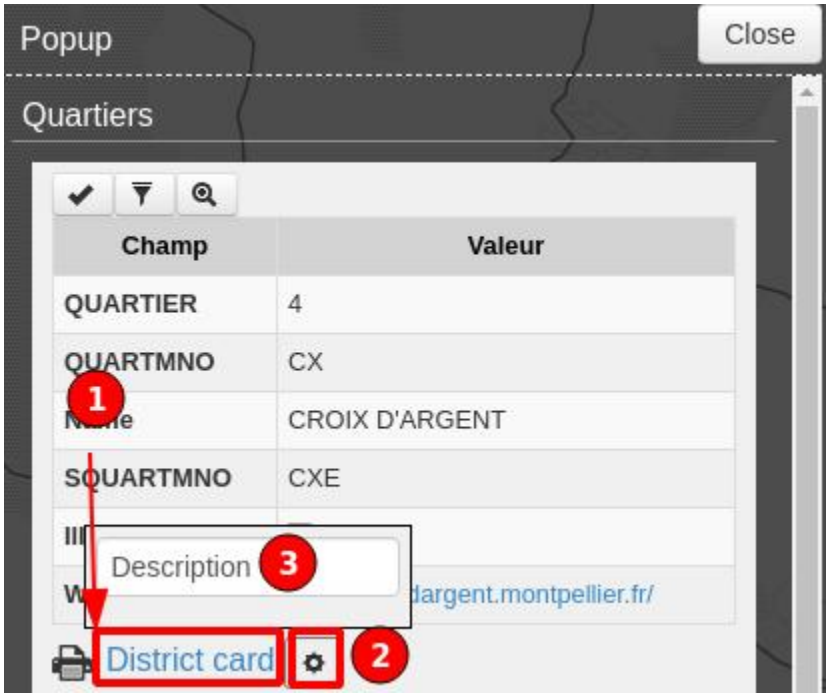


Layout with an atlas when using a popup

Using the `AtlasPrint` https://github.com/3liz/qgis-atlasprint plugin on QGIS Server, it's possible to automatically add a link to the PDF.

- Install the AtlasPrint plugin on the server
- Enable an atlas layout on a layer
- Enable *Popup* on the same layer

A link will be displayed automatically at the bottom of the popup. It's not possible to let the user fill some inputs. These PDF are linked to a feature due to the atlas.



Allow printing of external baselayers

The Lizmap plugin *Baselayers* tab allows you to select and add external baselayers (*Base layers*). These external baselayers are not part of the QGIS project, default print function does not integrate them.

To overcome this lack Lizmap offers an easy way to print a group or layer instead of the external baselayer. To be able to print a layer which is visible in Lizmap Web Client only:

- You need to add the equivalent layer in the QGIS project.
- You need to hide it from the Lizmap legend, see *Masking individual layers*.
- Rename the layer to one of these names:
  - `osm-mapnik` for OpenStreetMap
  - `osm-stamen-toner` for OSM Stamen Toner
  - `osm-cyclemap` for OSM CycleMap
  - `google-satellite` for Google Satellite
  - `google-hybrid` for Google Hybrid
  - `google-terrain` for Google Terrain
  - `google-street` for Google Streets
  - `bing-road` for Bing Road
  - `bing-aerial` for Bing Aerial
  - `bing-hybrid` for Bing Hybrid
  - `ign-scan` for IGN Scan
  - `ign-plan` for IGN Plan
  - `ign-photo` for IGN Photos

- `ign-cadastral` for IGN Cadastre

Note: The use of this method must be in compliance with the licensing of external baselayers used (*Base layers*).

Warning: If it's not working, check that your server is able to access to the internet. These base layers are provided online only. Some proxy or firewalls might block some requests to the internet. If your server is behind a proxy, check that QGIS Server is configured with the proxy settings (using the file `QGIS3.ini` and the section `[proxy]`). Refer to the QGIS Server documentation for these settings.

To add these layers, you can use existing WMS/WMTS services, XYZ providers (with QuickMapServices), local files…

For OpenStreetMap baselayers, it is possible to use an XML file for GDAL to exploit the OpenStreetMap tile services. Its use is described in the GDAL documentation https://gdal.org/frmt_wms.html or in this blog post https://www.3liz.com/blog/rldhont/index.php?post/2012/07/17/OpenStreetMap-Tiles-in-QGIS (beware, EPSG code should be 3857).

For IGN baselayers, you can use IGN's WMS or WMTS url. The key used for this url need to be protected by referer and IP. In referer, you have to indicate your projects page's URL like this : `.*your-url.fr.*`. In IP, you have to indicate your Lizmap server's IP and your computer's IP (to open IGN's WMS url in QGIS on your computer). Both IP addresses must be separated by a comma. Beware, if you use IGN WMS or WMTS layers, QGIS project's EPSG code should be 3857.

Adding your own images in a layout

If you add some custom images in a layout, such as custom North arrow or your organization logo, the server needs to access these images too.

- Either use an image with an URL `http://` so that your image is accessible on both your local computer and on the server.
- Or use QGIS expression to build a compatible path on both desktop and server:
  1. Put your images in the `media` directory (see *Media*), this is not mandatory, you can put it next to your project file.
  2. Use an QGIS expression `@project_home || '/media/organization_logo.png'`.
  3. Use slash even if you are on Windows.

3.3.9 Spatial searching

Contents

- *Spatial searching*
  - *PostgreSQL search*
    - *Prerequisites*
    - *Create the lizmap_search table or view*
    - *Optimisation*
    - *Configure access*

- *QuickFinder Plugin*
  - *Prerequisites*
  - *Configuration*

In the map options, you can activate and configure the address search bar, based on external web services (nominatim, google or french IGN). See *Map options* . Additionally, you can add spatial searching capability to Lizmap. This means you will allow the users to search within spatial data, such as countries, points of interests, etc. You have two ways to add searching capability in Lizmap:

- For **QGIS 2** and **QGIS 3**, you can create a table or view `lizmap_search` in your PostgreSQL database to store the search data for all your Lizmap projects.
- For **QGIS 2** only, you can use the plugin `QuickFinder` to configure a data search per QGIS project.

PostgreSQL search

When you have many projects and data, the best solution to provide searching capabilities is to set up a dedicated relation (table or view) inside your database. It's possible to use a PostgreSQL database to store the search data.

Prerequisites

- A PostgreSQL database, accessible from Lizmap Web Client.
- PostgreSQL extensions activated in this database : `unaccent` and `pg_trgm` (for effective LIKE queries)
- A custom function `f_unaccent` which can be used in an index. See this Stack Overflow post for explanation

```
-- Add the extension pg_trgm
CREATE EXTENSION IF NOT EXISTS pg_trgm;

-- Add the extension unaccent, available with PostgreSQL contrib tools. This is␣
↪needed to provide searches which are not sensitive to accentuated characters.
CREATE EXTENSION IF NOT EXISTS unaccent;

-- Add the f_unaccent function to be used in the index
CREATE OR REPLACE FUNCTION public.f_unaccent(text)
RETURNS text AS
$func$
SELECT public.unaccent('public.unaccent', $1)  -- schema-qualify function and␣
↪dictionary
$func$ LANGUAGE sql IMMUTABLE;
```

Note: We choose to use the `pg_trgm` extension and this custom `f_unaccent` function instead of the Full Text Search (FTS) tool of PostgreSQL, to keep the tool as simple as possible and avoid the need to create FTS "vectors" in your search data.

Create the lizmap_search table or view

The database admin must create a table, view or materialized view called `lizmap_search`. This relation must be accessible in the `search_path` (you can put it in the public schema, or configure the `search_path` variable for the database or the user which connects to the database).

The relation `lizmap_search` must contain the following columns:

- `item_layer` (text). Name of the layer. For example "Countries"
- `item_label` (text). Label to display the results, which is the data to search among. Ex: "France" or "John Doe - Australia". You can build it from a concatenation of several fields values.
- `item_project` (text). List of Lizmap projects separated by commas. Optionnal. When set, the search will be done only for the specified Lizmap projects
- `item_filter` (text). Username or group name. When given, the results will be filtered by authenticated user login and groups. For example, 'admins'
- `geom` (geometry). We advise to store all the geometries with the same SRID.

Here is an example of SQL code you can use, to add data from two different spatial tables into lizmap_search (here as a materialized view to ease further maintenance)

```
DROP MATERIALIZED VIEW IF EXISTS lizmap_search;
CREATE MATERIALIZED VIEW lizmap_search AS
SELECT
    'Commune' AS item_layer, -- name of the layer presented to the user
    concat(idu, ' - ', tex2) AS item_label, -- the search label is a concatenation␣
↪between the 'Commune' code (idu) and its name (tex2)
    NULL AS item_filter, -- the data will be searchable for every Lizmap user
    NULL AS item_project, -- the data will be searchable for every Lizmap maps␣
↪(published QGIS projects)
    geom -- geometry of the 'Commune'. You could also use a simplified version, for␣
↪example: ST_Envelope(geom) AS geom
FROM cadastre.geo_commune
UNION ALL -- combine the data between the 'Commune' (above) and the 'Parcelles'␣
↪(below) tables
SELECT
    'Parcelles' AS item_layer,
    concat(code, ' - ', proprietaire) AS item_label,
    'admins' AS item_filter, -- only users in the admins Lizmap group will be able to␣
↪search among the 'Parcelles'
    'cadastre,urban' AS item_project, -- the Parcelles will be available in search␣
↪only for the cadastre.qgs and urban.qgs QGIS projects
    geom
FROM cadastre.parcelle_info
;
```

Optimisation

- You should use a table, or a materialized view, on which you can add indexes to speed up the search queries.
- We strongly advise you to add a trigram index on the unaccentuated `item_label` field, to speed up the search query:

```
-- Create the index on the unaccentuated item_label column:
DROP INDEX IF EXISTS lizmap_search_idx;
CREATE INDEX lizmap_search_idx ON lizmap_search USING GIN (f_unaccent(item_label) gin_
↪trgm_ops);

-- You can refresh the materialized view at any time (for example in a cron job) with:
REFRESH MATERIALIZED VIEW lizmap_search;
```

Warning: At present, Lizmap PostgreSQL search cannot use 3D geometries, or geometries with Z or M values. You have to use the `ST_Force2D(geom)` function to convert geometries into 2D geometries.

Configure access

Once this table (or view, or materialized view) is created in your database, you need to check that Lizmap can have a read access on it.

If your Lizmap instance uses PostgreSQL to store the users, groups and rights, a connection profile already exists for your database. Then you can just add the `lizmap_search` relation inside this database (in the `public` schema).

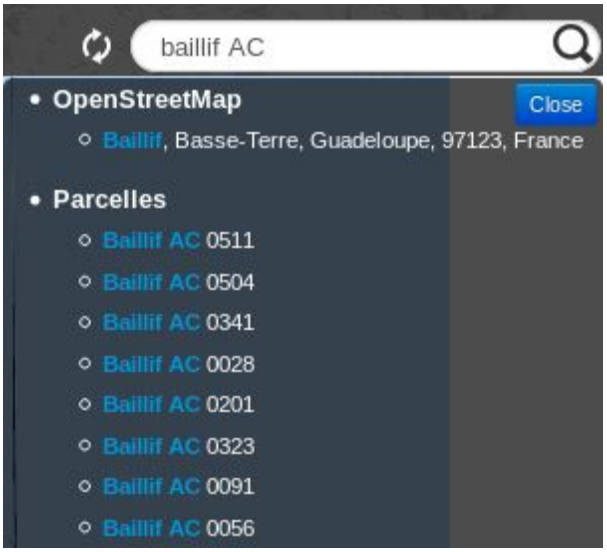
If not, or if you need to put the search data in another database (or connect with another PostgreSQL user), you need to add a new **database connection profile** in Lizmap configuration file `lizmap/var/config/profiles.ini.php`. The new profile is a new jdb prefixed section, called **jdb:search**. For example, add the following section (please replace the `DATABASE_` variables by the correct values):

```
[jdb:search]
driver="pgsql"
database=DATABASE_NAME
host=DATABASE_HOST
user=DATABASE_USER
password=DATABASE_PASSWORD
; search_path=DATABASE_SCHEMA_WITH_LIZMAP_SEARCH,public
```

You don't need to configure the *locate by layer* tool. The link with Lizmap Web Client is done automatically if you can run the query below successfully in PgAdmin using the same credentials as the Lizmap application. You **mustn't** specify the schema where the lizmap_search table is located. It **must** work without specifying the schema.

```
SELECT * FROM lizmap_search LIMIT 1;
```

You can now use the default search bar in Lizmap which is located on top right corner.



QuickFinder Plugin

Warning: This is for **QGIS 2** only. This plugin has not been ported to **QGIS 3**.

The purpose of this plugin is to provide fast searching among big datasets, searching in a qtfs file generated by QGIS Desktop.

Prerequisites

- You must have install at least the **7.x** version of **PHP** in your Lizmap server.

Configuration

Inside QGIS:

- install QuickFinder Plugin, for **QGIS 2** only
- choose a layer(s), define the fields to search among, pick the geometry storage format (WKT or Extent) and store Full Text Searchs (FTS) vector into a file database (.qfts). The filename must be identical to the QGIS project filename. Ex: `myproject.qfts` for a QGIS project stored as `myproject.qgs`.

Warning: Only **WKT** or **Extent** formats for geometry storage are working, since binary format (WKB) can not be decoded by LWC.

Inside LWC:

- put the database file beside the QGIS project, use the Search tool (input) and zoom to the chosen feature.

3.3.10 Other configuration

Contents

- *Other configuration*
  - *Changing the default image of a project*

Changing the default image of a project

By default the following image is displayed for a project:

You can change this default image by adding in the same project folder a `.png` or `.jpg` image with the exact project name and extension. Example, if the project is called `montpellier.qgs` you must add an image named `montpellier.qgs.png`. Note that the image has the project extension too.

Warning: It might be needed to clear the cache and force reload the page from the web browser.

3.3.11 Optimization

Contents

- *Optimization*
  - *General concepts*
  - *Tips*
  - *In detail: how to activate the caches*
    - *Server side*
    - *Client side*
  - *Centralizing the cache with the integration of groups and layers from a master project*

General concepts

Rendering speed is crucial for a WebGIS application, more than in a desktop application:

- web users expect to have everything available almost immediately
- each user can sends requests to the same application; if you have tens or hundreds of users, you can easily understand that optimising your web application is an important task.

You have to think about a web publication for many users rather than the display of a map to a single user.

By default, for each QGIS layer you add to your Lizmap project, you can choose from the Lizmap plugin whether to toggle the layer visibility on (checkbox *Toggled*) at the startup of the application. You have to be careful not to abuse this feature, because if the project contains e.g. 30 layers, Lizmap at startup will send a request to QGIS server for each of them.

If the checkbox *Single tile* is ticked, this will request 30 images of the size of your browser window. If not, Lizmap, through OpenLayers, will request 30 series of tiles (about 250 by 250 pixels). Each tile is an image, and is created as a function of the total window size and zooming level. Therefore, subsequent users will zoom in the same area, the tiles already generated will be reused.

The tiles can be cached with two non exclusive systems:

- *Server side*, on the machine where QGIS server and Lizmap are installed. If the tile has been requested and generated earlier, and not expired, Lizmap will reuse it and send it to the client, avoiding a new request to QGIS server.
- *Client side*: tiles will be saved in the browser cache and reused until they expire. This avoid both the request to QGIS server and the internet traffic.

The server cache has to be generated. Read *Cache management as administrator* to know how to generate the cache.

To optimize the performance, is therefore important to understand how Lizmap uses the tiles to be displayed.

Let's say you have a screen of 1280 by 768 pixels. If you have all your layers tiled, Lizmap has therefore to show about 5 by 3 = 15 tiles (256 by 256 pixel each) per layer, and more for a larger screen. If surrounding tiles are only partially shown, the total number will be even greater. An average of 20 tiles per layer is a reasonable estimate. With 30 layers, as in our example, this will mean a total of about 20 by 30 = 600 tiles (therefore, 600 requests to Lizmap server) per user, at each startup of Lizmap and for every zoom & pan. If you have 10 concurrent users, this gets quite heavy for the server, if the cache has not been generated previously, and QGIS server has therefore to create them. The time required for each tile will depend heavily on the performance of the server and the complexity of the project.

The size of each tile will depend on:

- the type of data (single raster or vector, or combination of several layers)
- the image format chosen (PNG, JPEG)

A typical tile could be around 30 Kb. In our example, the client will therefore download about 20 by 30 = 600 Kb per layer, which, for 30 layers, will give a grand total of about 18 Mb, which is heavy both for the server (lots of connection bandwidth consumed) and for the users (long delay, even with a reasonably fast connection).

These calculations show clearly that to achieve good performances in webmapping you have to make choices, and simplify as much as possible.

If one looks, for instance, at the approach taken by Google Maps or similar services, it is quite obvious that, besides having powerful servers, they have simplified as much as possible: only one tile series as a base layer, and very few additional layers (and not all at the same time). Even if you cannot create such a simple map, it's important nonetheless knowing which layers should absolutely be shown at the first display of the map, and which compromises are acceptable for your users.

If your project has 50 layers to be switched on and off, the vast majority of your users will never select most of them. Of course, there are real use cases where individual layers must be displayed selectively, and it is therefore not possible to group them to reduce the number of layers displayed.

Tips

To optimize your application as much as possible, we suggest you to:

- Create separate QGIS projects, and therefore different Lizmap maps, for different aims, thus grouping data in logical themes. For instance, a map about urban development with maybe 10 layers and one about environment, with about 5 layers, are usually more readable, and much faster, than a single overcomplex project with all the data. Adding a small image for each project will help users to select the relevant project at first sight. You can also share some of the layers among different projects, through the embedding mechanism in QGIS.
- Use the option *Maps only* in the administrator web interface. This option allows the user to switch automatically from one map to another, through the button *Home*, maintaining as much as possible the localization and the zooming level. In this case, the Lizmap welcome page with the list of projects and their thumbnails is not displayed, and the user is directed automatically to one of the projects, at the administrator choice.
- Do not show all the layers at startup (deactivate the checkbox *Toggled* as described above). Only very important layers should be visible by default, and users should activate only the layer they need. This allow a sensible reduction in the number of requests, and of the total network traffic.
- Create groups of layers, and use the option *Group as layer* in Lizmap plugin. Generally a series of layers of the same general theme can be displayed as a whole, with an appropriate choice of styles. In this case, Lizmap will only show one checkbox for the whole group, and more importantly it will request only one series of tiles for the whole group, thus reducing the number of tiles and server requests, and the total volume of data to be downloaded. The legend of the group will be displayed.
- Use the option *Single tile* for some layers. In this case, Lizmap will request only one image per layer, of about the size of the screen, instead of a series of tiles. This will therefore greatly reduce the number of requests to the server. For instance, in our example above, without the optimizations described, if all the layers are displayed, every user will request 30 images (one per layer) for every zoom or pan, instead of 480. The total size of data to be downloaded is however similar. On the other hand, different users will be very unlikely to request exactly the same image, therefore using a cache is pointless in this case, and is avoided by Lizmap (the two options are mutually exclusive). The optimal choice (single tile vs. tiled) is different for different layers. For instance, a complex base layer, created by combining 15 individual layers, will be best used as a group (*Group as layer*), tiled and cached. A simple linear layer, like a series of bus lines, can be displayed as a single tile.
- Use the option *Hide checkboxes for groups*: this avoids the users to click on a group with e.g. 20 layers without really needing it, thus firing a big series of requests to the server. In any case, avoiding groups of more than 5-10 layers is usually good practice.
- Optimize the data and the QGIS project. As mentioned above, publishing a map over the internet will change your point of view: as said, you have to remember that many users can hit the server in parallel, so avoiding to overload it is crucial to:
  - create a spatial index for all your vector layers
  - pyramidize all your raster layers (except the very small ones)
  - only display data at appropriate scale: for instance, displaying a detailed building layer at 1:500,000 is meaningless, as the image is almost unreadable, and puts a lot of stress on the server
  - use simplified version of a layer to display it at different scales. You can then group the original layer (to be displayed e.g. around 1:1,000) with the simplified versions (to be displayed e.g. around 1:10,000, 1:50,000, etc.), and *Goup as a layer* to let the user see this as a single layer, using the most appropriate data at each scale

  - be careful about On The Fly (OTF) reprojection. If, for instance, you display data in Lambert 93 (EPSG:2154) on a base map from OpenStreetmap or Google (in Pseudo Mercator, EPSG:3857), QGIS Server needs to reproject rasters and vectors before generating the map. This may have an impact in rendering times for large and complex layers. In France, you can avoid reprojection by using the base map from IGN Géoportail directly in EPSG:2154
  - be aware of the fact that certain rendering options (e.g. labels, expressions, etc.) can be very demanding from the server
  - if you use PostGIS, optimize it: always add spatial indexes, indexes for filtered fields, for foreign keys, appropriate parameters for the configuration of PostgreSQL, possibly a connection through Unix socket instead of TCP/IP (you can do this through the use of services), etc.
  - use an appropriate image format. For the base layers, where you do not need transparency JPEG is usually the best option: the tiles will be smaller, and faster to download. For other layers, try smaller depth PNGs (16bit or 8bit): for some symbols, the visual result may be the same, and the tiles smaller. Have a check to see if the image quality is acceptable in your case
  - Use server side simplification if possible. Read *Server side simplification*.
- Upgrade your server. This is always an option, but is often useless if you did not optimize your project as described above. In any case, a low end server (e.g. 2 Gb RAM, 2 cores at 2.2 GHz) is unsuitable. A fast quad-core with 8 Gb RAM is a reasonable minimum. Avoid installing QGIS server and Lizmap on Windows, it's more complex and slower.
- Lizmap avoids the automatic download of the legends at startup and at every zoom level. This will be done exclusively on demand, if the legend is displayed, thus saving one request per layer for each zoom.

In detail: how to activate the caches

In *Lizmap plugin ▸ Layers*, you can enable for each layer or group the cache (client and server side) for generated images.

Server side

This feature is not compatible with the option *Single tile*. Lizmap Web Client can dynamically create a cache tiles on the server. This cache is the storage of the images already generated by QGIS-Server on the server. The Lizmap Web Client application automatically generates the cache as the tiles are requested. Enable caching can greatly lighten the load on the server, since we do not want more QGIS-Server tiles that have already been made.

To activate it, you must:

- check the box *Server tile cache*
- specify the expiration time of the cache server in seconds: *Expiration (seconds)*. *0* means no expiration on the server, the tile will be kept on the server until the cache is cleared.

The *Metatile* option allows you to specify image size in addition for generating a tile. The principle of *Metatile* is to request the server for a bigger image than hoped, to cut it to the size of the request and return it to the Web client. This method avoids truncated labels at the edges and discontinuities between tiles, but is more resource intensive. The default value is *3,3*, an image whose width and height are equal to 5 times the width and height request. This option is useless for rasters.

Note: For vector layers with metatiles, you have to set label position to a fixed position. If the label is very long, do not forget to use a word wrap function or to increase the metatile size a little bit. So labels are displayed in the same area.

Client side

The *Browser client cache* option allows you to specify an expiration time for the tiles in the Web browser (Mozilla Firefox, Chrome, Internet Explorer, Opera, etc.) cache in seconds. When browsing the Lizmap map with the browser, it stores displayed tiles in its cache. Enable client cache can greatly optimize Lizmap because the browser does not re-request the server for tiles already in cache that are not expired.

Warning: Values *0* and *1* are equivalent and do not activate the option.

We suggest to set to the maximum value (1 month equals to 24 x 3600 x 30 = 2,592,000 seconds), except of course for layers whose data changes often.

Note:

- **The cache must be activated only once mastered rendering**, when you want to move the project into production.
- **These two cache systems, server and client, are completely independent** of one another. But of course, it is interesting to use the two together to optimize the application and free server resources.

Centralizing the cache with the integration of groups and layers from a master project

In QGIS, it is possible to integrate in a project, groups or layers from another project (which will be called "parent"). This technique is interesting because it allows you to set the properties of the layers once in a project and use them in several other, for example for baselayers. In the "child" projects that integrate these layers, it is not possible to change these properties.

Lizmap uses this feature to centralize the tiles cache. For all child projects using integrated layers of the parent project, Lizmap requests QGIS-Server tiles from the parent project, not from child projects. The cache will be centralized at the parent project and all child projects that use layers benefit the shared cache.

To use this feature, you must:

- publish the parent QGIS project with Lizmap
  - you must **choose the right announced extent** in *project properties ‣ QGIS Server*, because this **extent will be reused identically in child projects**.
  - you must **configure the cache** for the layers to integrate. Also, note the options chosen here (image format, metatile, expiration) for use as such in child projects.
  - It is possible to hide the project from the main page of Lizmap with the check box in *Map ‣ Hide the project Web Client Lizmap*.
- open the child project and integrate layers or groups in this project, for example orthophoto. Then you must:
  - verify that the **announced extent** in *QGIS project properties ‣ QGIS Server* is **exactly the same as the parent project**.
  - you must **configure the cache** for the integrated layer **with exactly the same options as those selected from the parent project**: image size, expiration, metatile.
  - you must set the Lizmap id of the **Source repository** of the parent project (The one configured in the Lizmap Web Client administration interface).
  - the code of the "Source project" (the name of the parent QGIS project without the *.qgs* extension) is automatically entered for layers and integrated groups.

- Publish the child project to the Lizmap Web Client as usual.

3.4 Customization

This section shows how to customize Lizmap with some lines of code.

3.4.1 Adding your own JavaScript

Contents

- *Adding your own JavaScript*
  - *Principle*
  - *Prerequisites*
  - *Configuring the tool*
  - *Video tutorial*
  - *Library of scripts*
  - *Available Javascript events*
  - *Examples*
    - *Collapse a group in the legend*
    - *Disable right click*
    - *Send current login user-ID*
    - *URL of a static file*

Principle

Adding some JavaScript (JS) is useful for a variety of advanced usage. For instance, you can:

- hide some UI elements that you don't want to display by default
- add a custom button in the UI
- add a popup when the project is opened (to display funders, credits…)
- avoid people being able to download elements of the page by right clicking on them, and of course much more.

Prerequisites

- This function needs to be activated by the administrator of the Lizmap instance.
- The `media` directory. Read how to use *Media* folder in Lizmap.

Configuring the tool

- In your repository (e.g. `/home/data/repo1/myproject.qgs`), you should have these directories:

```
media
|-- js
  |-- myproject
  |-- default
```

- All the JavaScript code you copy in the `/home/data/rep1/media/js/myproject/` directory will be executed by Lizmap for this **specific** project only.
- All the JavaScript code in `default` will be executed for **all** projects.
- To allow the execution of JavaScript code, in the Lizmap admin interface, you **must** add the privilege *Allow themes for this repository*.

Video tutorial

This video is an quick start how to add a JavaScript to change the default panel in Lizmap.

https://www.youtube.com/embed/xQQ34nvRZ-w

Library of scripts

You can find some examples in the repository https://github.com/3liz/lizmap-javascript-scripts.

Also, in the directory `lizmap-web-client/lizmap/install/qgis/media/js/` (or in GitHub https://github.com/3liz/lizmap-web-client/tree/master/extra-modules/lizmapdemo/qgis-projects/demoqgis/media/js/montpellier), you can find examples of suitable JavaScript code.

Remove the extension `.example` and copy them to your `media/js/default/` folder to activate them.

Available Javascript events

The Javascript code can use many events fired by Lizmap Web Client. Here is a list of all the events available, with the returned properties.

Table 1

Event name	Description
treecreated	Fired when layer tree has been created in legend panel
mapcreated	Fired when OpenLayers map has been created
layersadded	Fired when Openlayers layers have been added
uicreated	Fired when interface has been created
dockopened	Fired when a dock is opened (left panel)
dockclosed	Fired when a dock is closed (left panel)

Table

Event name	Description
minidockopened	Fired when a mini-dock (right container for tools) is opened
minidockclosed	Fired when a mini-dock is closed
bottomdockopened	Fired when the bottom dock is opened
bottomdockclosed	Fired when the bottom dock is closed
lizmapbaselayerchanged	Fired when the baselayer has been changed
lizmapswitcheritemselected	Fired when a layer has been highlighted in the layer legend panel
layerstylechanged	Fired when a layer style has been changed
lizmaplocatefeaturecanceled	Fired when the user has canceled the locate by layer tool
lizmaplocatefeaturechanged	Fired when the user has selected an item in the locate by layer tool
lizmappopupdisplayed	Fired when the popup content is displayed
lizmappopupallchildrendisplayed	Fired when the all children popups are displayed
lizmappopupdisplayed_inattributetable	Fired when the popup content is displayed in attribute table (right sub-panel)
lizmapeditionformdisplayed	Fired when a edition form is displayed
lizmapeditionfeaturecreated	Fired when a layer feature has been created with the edition tool
lizmapeditionfeaturemodified	Fired when a layer feature has been modified with the edition tool
lizmapeditionfeaturedeleted	Fired when a layer feature has been deleted with the edition tool
attributeLayersReady	Fired when all layers to be displayed in the attribute layers tool have been set
attributeLayerContentReady	Fired when a table for a layer has been displayed in the bottom dock
layerfeaturehighlighted	Fired when a feature has been highlighted in the attribute table (grey rectangle). Firing this ev
layerfeatureselected	Fire this event to trigger the selection of a feature for a layer, by passing feature id. Once the se
layerfeaturefilterselected	Fire this event to trigger the filtering of a layer for the selected features. You must select some
layerFilteredFeaturesChanged	Fired when a filter has been applied to the map for a layer. This event also trigger the redrawin
layerFilterParamChanged	Fired when the WMS requests parameters have changed for a layer. For example when a STYI
layerfeatureremovefilter	Fire this event to remove any filter applied to the map. Once done, the event layerFilteredFeatu
layerSelectionChanged	Fired when the selection have been changed for a layer. This also trigger the redrawing of attril
layerfeatureselectsearched	Fire this event to select all the features corresponding to the displayed lines of the attribute tabl
layerfeatureunselectall	Fire this event to remove all features from selection for a layer. Once done, Lizmap responds w
lizmapexternalsearchitemselected	Fired when the user has selected an item listed in the results of the header search input
actionResultReceived	Fired when a Lizmap popup action has been performed and the result has been received

There are also some variables which are available.

Table 2: Lizmap Web Client available variables

Variable name	Description
lizUrls.media	URL to get a media
lizUrls.params.repository	Name of the current repository
lizUrls.params.project	Name of the current project

Examples

Collapse a group in the legend

```
lizMap.events.on({
    uicreated: function(e) {
        $('#group-groupname td a.expander').click();
    }
});
```

Disable right click

Add a file named e.g. `disableRightClick.js` with the following code:

```
lizMap.events.on({
   uicreated: function(e) {
      $('body').attr('oncontextmenu', 'return false;');
   }
});
```

- If you want this code to be executed for all projects of your repository, you have to copy the file in the directory `/home/data/rep1/media/js/default/` rather than in `/home/data/rep1/media/js/myproject/`.

Send current login user-ID

An example allowing you to send current login User-ID (and/or other user data) to PostgreSQL table column, using edition tool:

```
var formPrefix = 'jforms_view_edition';

// Name of the QGIS vector layer fields which must contain the user info
// In the list below, replace the right side by your own fields in Lizmap
var userFields = {
   login: 'your_lizmap_user_login_field',
   firstname: 'your_lizmap_user_firstname_field',
   lastname: 'your_lizmap_user_lastname_field',
   organization: 'your_lizmap_user_organization_field'
};


lizMap.events.on({

   'lizmapeditionformdisplayed': function(e){

      // If user is logged in
      if( $('#info-user-login').length ){
            // Loop through the needed fields
            for( var f in userFields ){
               // If the user has some data for this property
               if( $('#info-user-' + f).text() ){
                  // If the field exists in the form
                  var fi = $('#' + formPrefix + '_' + userFields[f]);
                  if( fi.length ){
                        // Set val from lizmap user data
                        fi.val( $('#info-user-' + f).text() )
                        // Set disabled
                        fi.hide();
                  }
               }
            }
      }

   }

});
```

URL of a static file

If you want to get the URL of a static file, located in the *Media* folder:

```
var mediaUrl = OpenLayers.Util.urlAppend(
     lizUrls.media,
     OpenLayers.Util.getParameterString({
          "repository": lizUrls.params.repository,
          "project": lizUrls.params.project,
          "path": "picture.png"
     })
);
```

or

```
var domain = 'https://' + document.domain + lizUrls.basepath;
var image  = '<img src="' + domain + 'index.php/view/media/getMedia?
↪repository=REPOSITORY&project=PROJECT&path=/media/logo.png" title="Logo" />';
```

3.4.2 Creating simple themes

Contents

- *Creating simple themes*
  - *Principle*
  - *Prerequisites*
  - *Configuring the tool*
  - *Example*

Principle

It is possible to create themes for all maps of a repository or for a single map within a repository.

The principle is:

- the directory `media` contains a directory named `themes`
- the directory `themes` contains a `default` directory for the theme of all the maps of the repository
- the directory `themes` may contain too one directory per project, for the themes specific for each project

```
-- media
  |-- themes
    |-- default
    |-- map_project_file_name1
    |-- map_project_file_name2
    |-- etc
```

Prerequisites

- This function needs to be activated by the administrator of the Lizmap instance.
- The `media` directory. Read how to use *Media* folder in Lizmap.

Configuring the tool

Warning: The web browser has some caching mechanism. Do not forget to refresh and force the cache with Ctrl+F5.

In order to simplify the creation of a theme for a repository or a map, Lizmap allows you to obtain the default theme from the application, through the request: `index.php/view/media/getDefaultTheme`.

The request returns a zipfile containing the default theme, with the following structure:

```
-- lizmapWebClient_default_theme.zip
  |-- default
    |-- css
      |-- main.css
      |-- map.css
      |-- media.css
      |-- img
        |-- loading.gif
        |-- etc
      |-- images
        |-- sprite_20.png
        |-- etc
```

Once downloaded the zipfile, you can:

- replace the images
- edit the CSS files

Warning: The files and directories must be readable (755:644)

Tip: To preview your results without deploying it in production, you can add your theme in the `lizmap/www/themes`. Add `&theme=yourtheme` at the end of your URL (e.g. `https://your.lizmap.instance/index.php/view/map/?repository=montpellier&project=montpellier&theme=yourtheme`).

Once your theme is ready, you can just publish it copying it in the directory `media`.

Example

We want to change the logo and the navigation bar background color (e.g. blue) *only* in a specific project called `roads` and we want to keep the default theme from the Lizmap *instance*:

- We don't need the `media/themes/default` folder.
- Create `media/themes/roads`.
- Extract the `css/` directory from the zip file inside.
- Change the file `css/img/logo.png`

This would work. But you still have a lot of CSS which is the same from the Lizmap main instance. So we can make our style smaller:

- Remove all images which are the same as Lizmap instance
- Search in the `css` folder where `logo.png` is used.
- Remove every files `*.css` except `css/main.css` and `css/map.css` and keep only:

```
#logo {
  background : url(img/logo.png) no-repeat;
  background-size:contain;
}
```

for `css/main.css` and:

```
#navbar button.btn {
  background-color : blue;
}
```

for `css/map.css`

By following these steps, we keep our custom theme as small as possible.

3.4.3 Custom templates

In Lizmap each module (i.e., view, admin, action, etc.) defines some templates. Templates of each module are located in `lizmap/modules` folder. Each module has one or more templates (.tpl) in the templates folder `lizmap/modules/moduleX/templates/`. In order to customize the template of a module it is advisable to create a copy of the files without modify the original files. To do this, it is necessary to create a copy of the templates inside the default theme's folder of Lizmap, which is in `var/themes/default`.

Prerequisites

- A folder with the same name of the module to which the template to redefine belongs, in the `var/themes/default` folder.

Configuring the tool

Simply copy the template to redefine in the module folder and customize it with a text editor.

Example

We want to change the default title in the header of the main page of lizmap. The "view" module and the `main.tpl` template are involved in this procedure. We can see the name of the module in the URL bar (`myhost/lizmap/index.php/view/`).

- Create a directory named view in the theme's default folder

```
nano mkdir lizmap/var/themes/default/view
```

- Copy the `main.tpl` file from the default location in `lizmap/modules/view/templates/` to the `lizmap/var/themes/default/view` folder

```
cp lizmap/var/themes/default/view/main.tpl lizmap/var/themes/default/view
```

- Find the title div and replace the original code with your custom text

```
<div id="title">
  <h1>{$repositoryLabel}</h1>
</div>
```

```
<div id="title">
  <h1>Some Text</h1>
</div>
```

Done!

3.4.4 Iframe

Contents

- *Iframe*
  - *Principle*
  - *Example*

Principle

It's possible to embed Lizmap in another webpage by using a iframe. We suggest you to change the view `map` by `embed` in the URL to have lighter version of the interface.

For instance, use this URL : `/index.php/view/embed/?repository=my_repo&project=my_project`.

Example

You can visit both links to see the difference :

- https://demo.lizmap.com/lizmap/index.php/view/map/?repository=feat1&project=lampadaires
- https://demo.lizmap.com/lizmap/index.php/view/embed/?repository=feat1&project=lampadaires

CHAPTER

FOUR

ADMINISTRATOR GUIDE

This guide is for the system administrator on the server.

4.1 Overview of the administration interface

Contents

- *Overview of the administration interface*
  - *Access the administration interface*
  - *Change password*
  - *Reminder on Lizmap Web Client repositories*

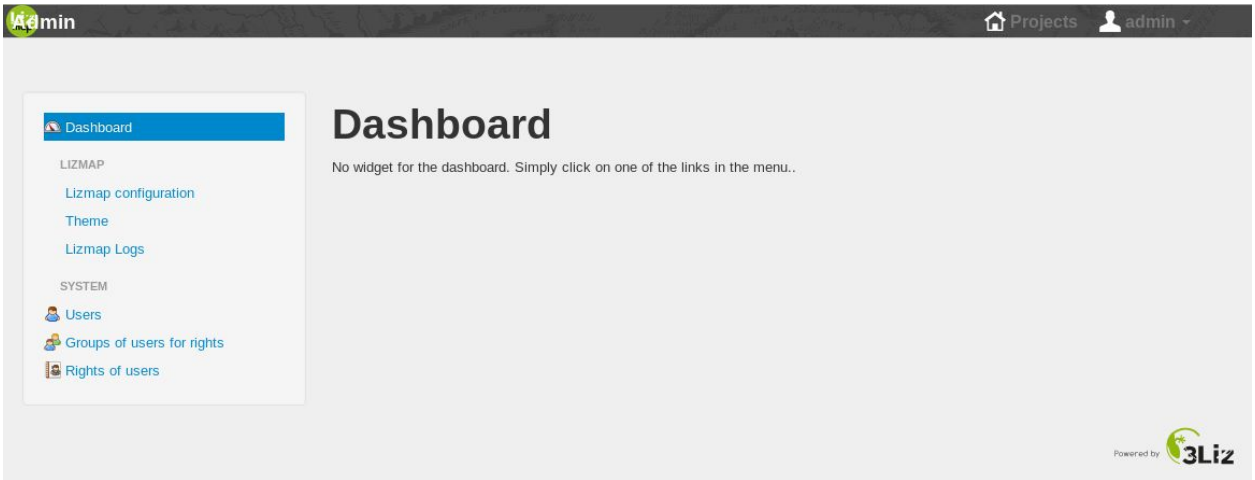
4.1.1 Access the administration interface

For safety reasons, no link leads to the administration interface. The address is:

http://mydomain.lizmap.3liz.com/admin.php

Temporary identifiers

- login = admin
- password = admin



4.1.2 Change password

Once connected, it is possible to change the password:

- Click on your login at the top right menu and then *Your account*
- Click on the button *Change your password*
- Type the new password, confirme it and save

Warning: It is imperative to change your password. Use a complex password that mixes numbers, letters and punctuation.

4.1.3 Reminder on Lizmap Web Client repositories

Lizmap Web Client can use QGIS maps stored in different folders on the server.

- A *Lizmap Web CLient repository* is a folder on the server that stores one or more QGIS projects and their related data.
- Repositories can be used to *group related projects by theme*.
- Repositories can be used to *manage the access rights on maps*.

4.2 Manage groups and users in Lizmap Web Client

Contents

- *Manage groups and users in Lizmap Web Client*
  - *The groups and users: principles*
  - *Manage Groups: create, rename, delete*
  - *Manage Users*

- *Setting password for users*
- *Putting users in groups*
  - *Assigning groups for each user created*

4.2.1 The groups and users: principles

As an administrator, you can:

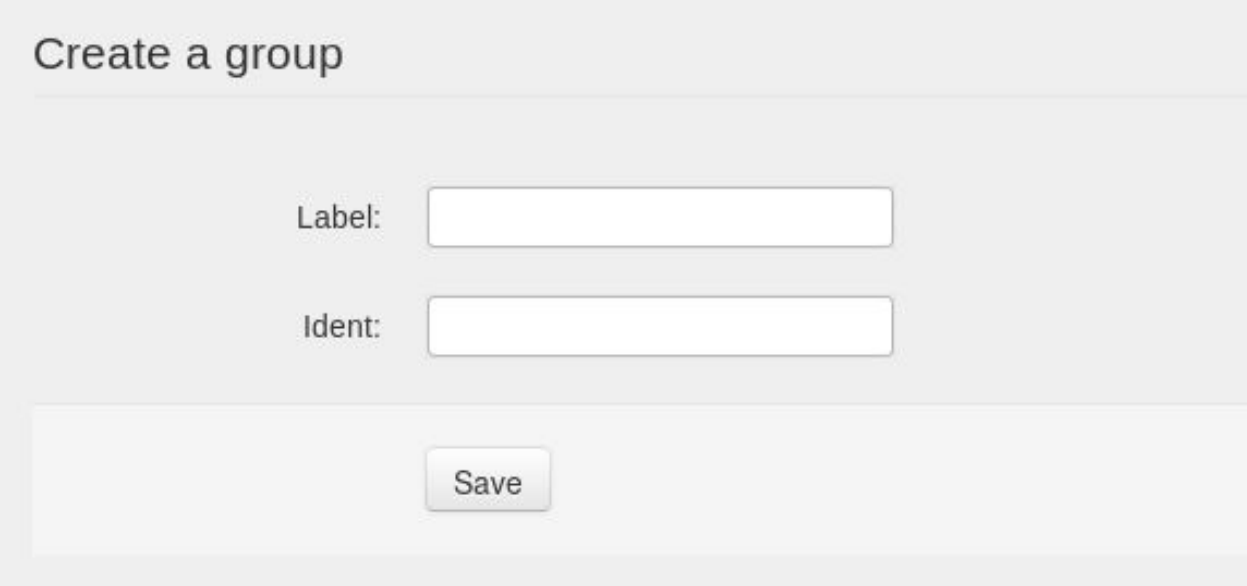
- create, rename, delete user groups
- create, modify, delete users
- linking a user to one or more groups

Note: Rights on Lizmap Web Client repositories are managed at group level, not at the user level.

4.2.2 Manage Groups: create, rename, delete

- *Create a group*: In the left menu click on *Groups of users for rights* and scroll down to *Create a group*.
- Define the *label*: it is possible to use spaces and accents
- Define the *ident*: one word without special characters

In the page *Groups of users for rights*, it is also possible to *Change the name* and *Delete a group*.

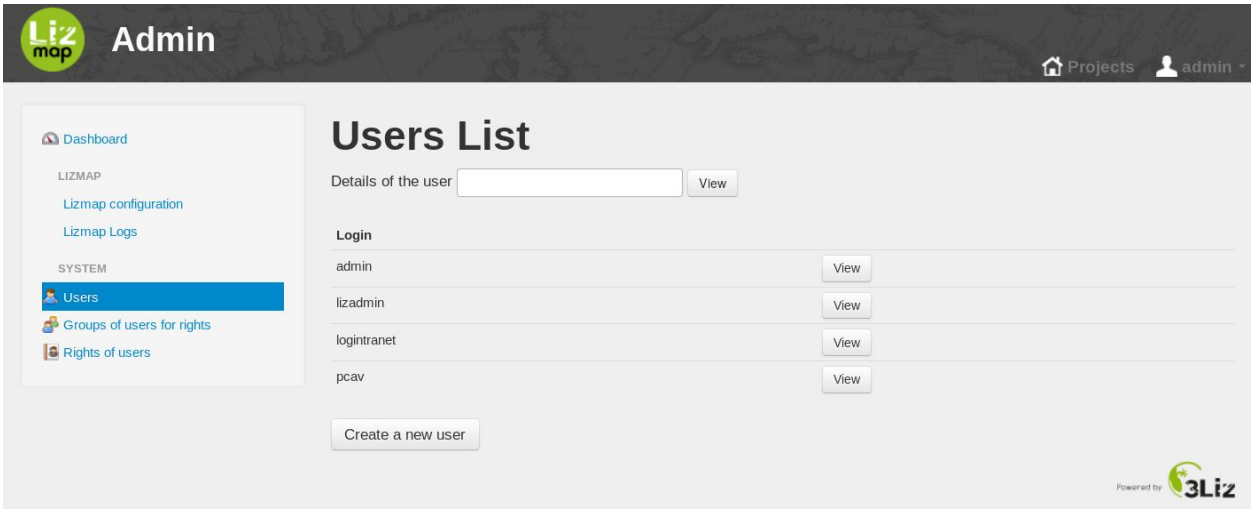


Note: The **users** group is a group system to give the rights to authenticated users to edit their own user information including passwords. We excluded this group of Lizmap Web Client configuration because all identified users must be part of this group.

4.2.3 Manage Users

- *Create a user* : In the left menu click on *Users* and click on the button *Create a new user*:
  - give a *Nickname* which will be used for the login
  - give an email
  - set name and firstname
- The user will have an email for setting up his password. If you want to set the password yourself, read the section below.
- It is also possible to *view* and *edit* informations about users.

Warning: Once users created, you must put them into groups to assign the associated rights.



4.2.4 Setting password for users

By default, in Lizmap, when a new user is created, the user will get an email for setting up his own password. If for some reasons, the administrator needs to setup the password, the `localconfig.ini.php` needs to be edited:

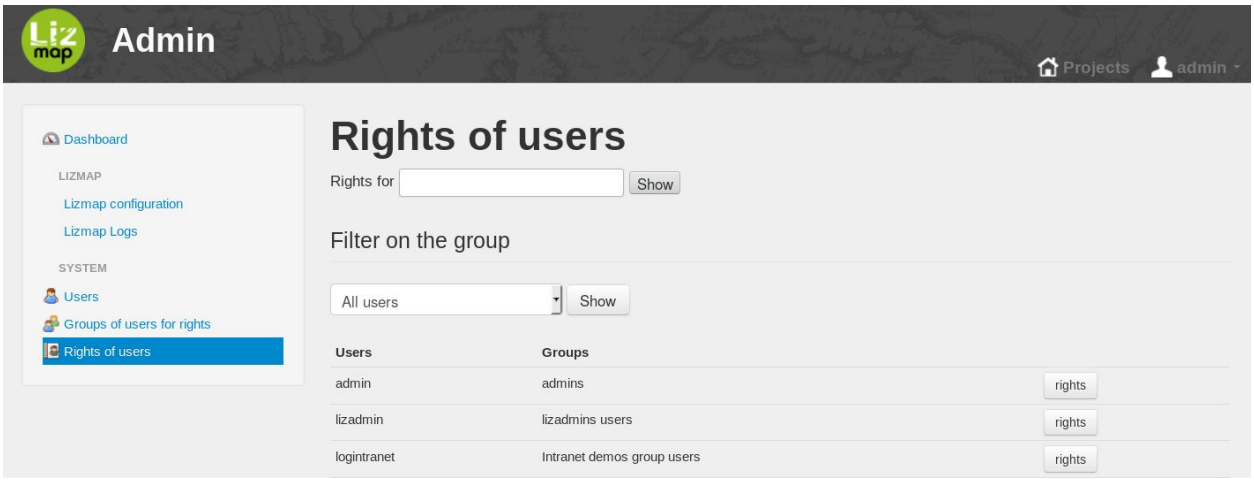
```
[jcommunity]
resetAdminPasswordEnabled = off
```

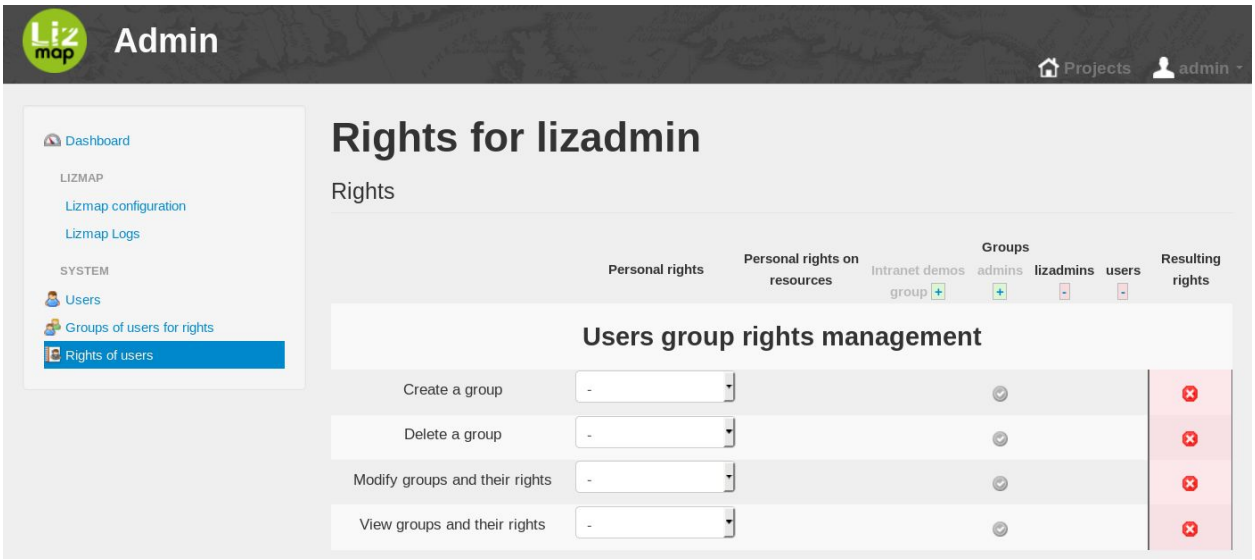
4.2.5 Putting users in groups

- In the left menu click on *Rights of users*
- it is possible to filter the visible users with the dropdown menu
- to put a user in groups, click on the its button *rights*
  - The page displays a table with *rights online and groups on column*
  - The last column shows *the resulting rights*
  - In the *head row*, the + and - buttons allow to set / remove the user from a group

Warning: Do not use the *Personal rights* column.

Assigning groups for each user created





4.3 Lizmap Web Client configuration

Contents

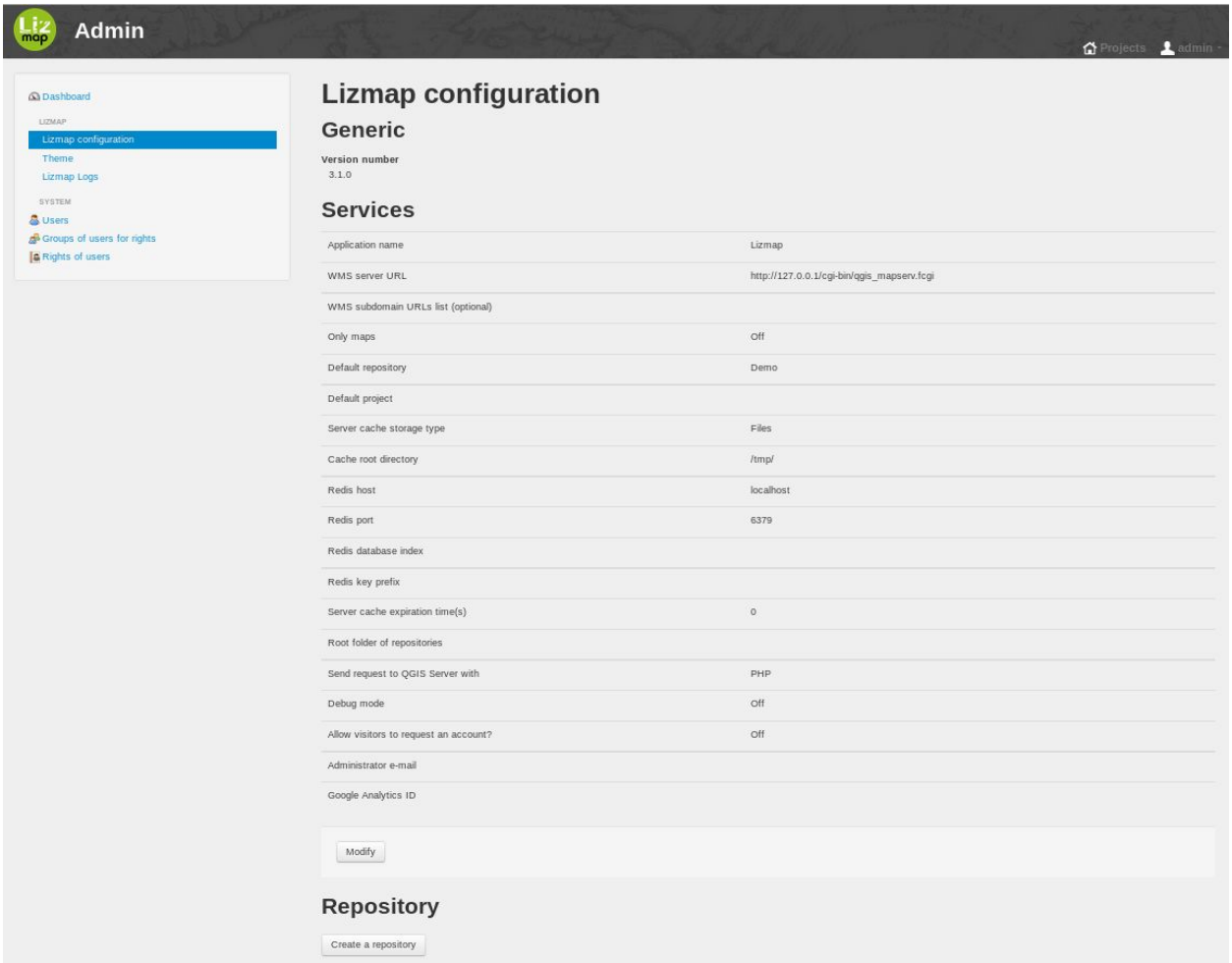
- *Lizmap Web Client configuration*
  - *Introduction*
  - *Services*
    - *proxy for External requests*

- *Repositories*
  - *Add a repository*
  - *Define the rights for each group*

4.3.1 Introduction

The *Lizmap configuration* menu is divided into 2 parts:

- **Services**: the general configuration of Lizmap Web client - server, cache, etc.
- **Repositories**: create and configure Lizmap repositories



4.3.2 Services

To configure **Services**, click the *Modify* button under the summary.

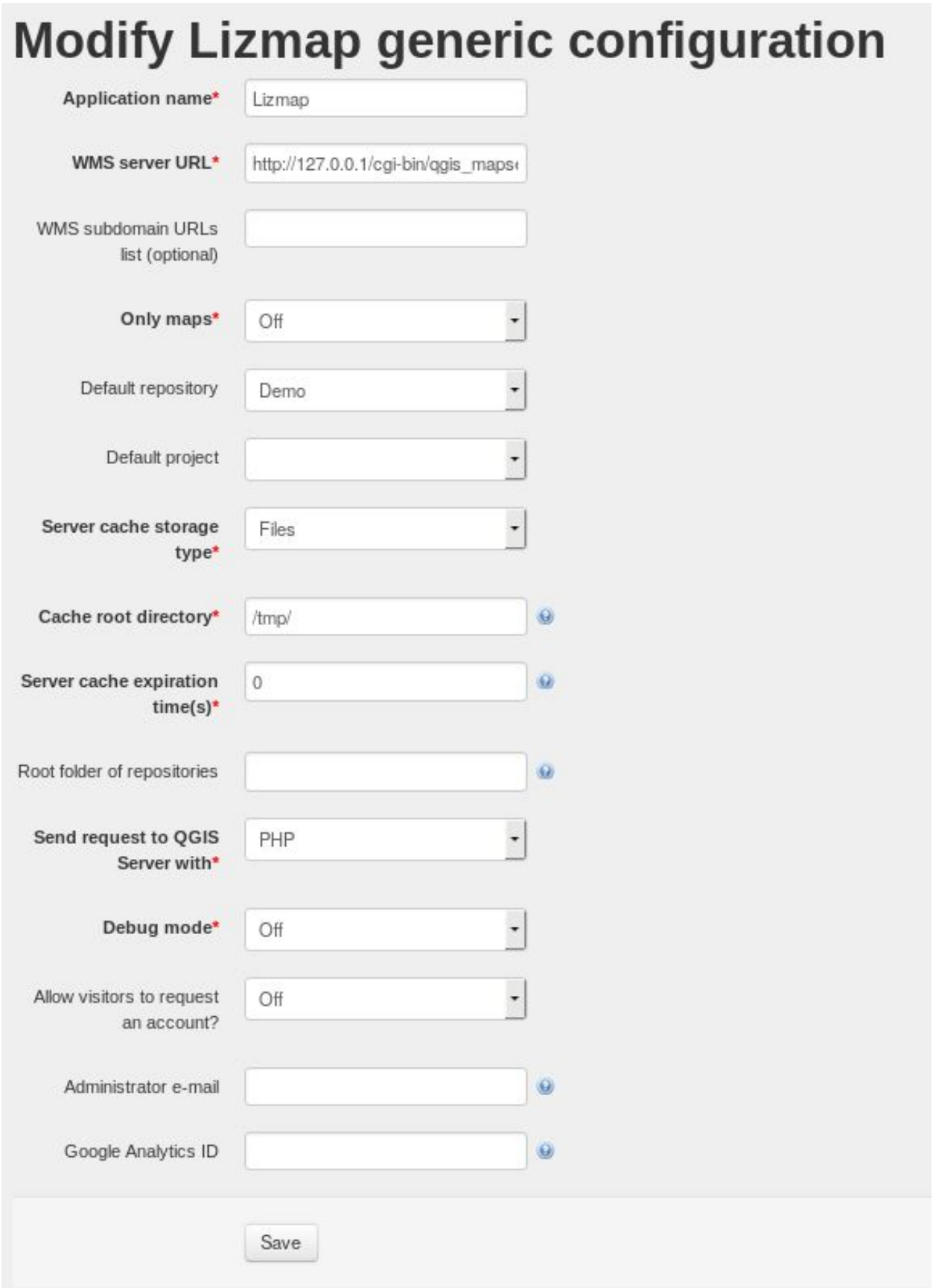
- **WMS server URL**: The full QGIS Server URL, such as http://localhost/cgi-bin/qgis_mapserv.fcgi or http://localhost/cgi-bin/qgis_mapserv.fcgi.exe.

> **Warning:** QGIS Server must be installed on the same computer as Lizmap Web Client

- **WMS subdomain URLs list (optional)**: The use of multiple domain names is one of the classic optimizations when a web application uses OpenLayers (as Lizmap Web Client). You can enter a list of subdomains separated by comma.
  - You must use a **list of subdomains** relative to the domain with which is used Lizmap Web Client. For example, if your main domain name is **maps.example.com**, then you can use **a.maps.example.com, b.maps.example.com, c.maps.example.com, d.maps.example.com**.
  - Of course you must have configured the Apache/Nginx server *virtual host* to take into account these subdomains, for example with the variable:

```
ServerAlias \*.maps.example.com
```

- **Server cache storage type**
  - *file*: Tiles cached are stored in a server directory by layer
  - *sqlite*: The tiles are stored in a sqlite database by layer
  - *redis*: the tiles are stored into a Redis database
- **Cache root directory**: the folder where the cache is stored. It must be writable by the Apache/Nginx server.
- **Server cache expiration time (s)**: the time in seconds during which each tile is retained. This is a default value for layers whose time has not been configured with the plugin.
  - The cached tiles older than that time are automatically refreshed.
  - A value of 0 means that the tiles do not expire.
  - The expiration time must be adapted to changing data.
- **Send request to QGIS Server or external service with**: 2 methods *php* or *curl*. Use first if curl is not installed on the server.
- **Debug mode**: saves some queries to a log file: *lizmap/var/log/messages.log*
- **Allow visitors to request an account**: If this option is enabled, a new link 'Registration' will be added in the public menu. By clicking on this link, the visitor displays a form that allows it to request an account to administrator. He must complete some fields (name, email, due to demand) then validates the form to send the request.
- **Administrator e-mail**: If a valid e-mail address is given, then the Lizmap Web Client notifications will be sent. For example, each account creation request via the registration form generates an email sent to this address.
- **sender e-mail** and **sender name**: when Lizmap sends an e-mail like notifications or for password reset, if needs the email address and the name of a sender.


Modify Lizmap generic configuration
Application name*
Lizmap
WMS server URL*
http://127.0.0.1/cgi-bin/qgis_maps
WMS subdomain URLs list (optional)
Only maps*
Off
Default repository
Demo
Default project
Server cache storage type*
Files
Cache root directory*
/tmp/
Server cache expiration time(s)*
0
Root folder of repositories
Send request to QGIS Server with*
PHP
Debug mode*
Off
Allow visitors to request an account?
Off
Administrator e-mail
Google Analytics ID
Save

proxy for External requests

Lizmap may need to access to some internet services. So, if it is installed behind a web proxy (typically in an intranet), you need to indicate what is the proxy.

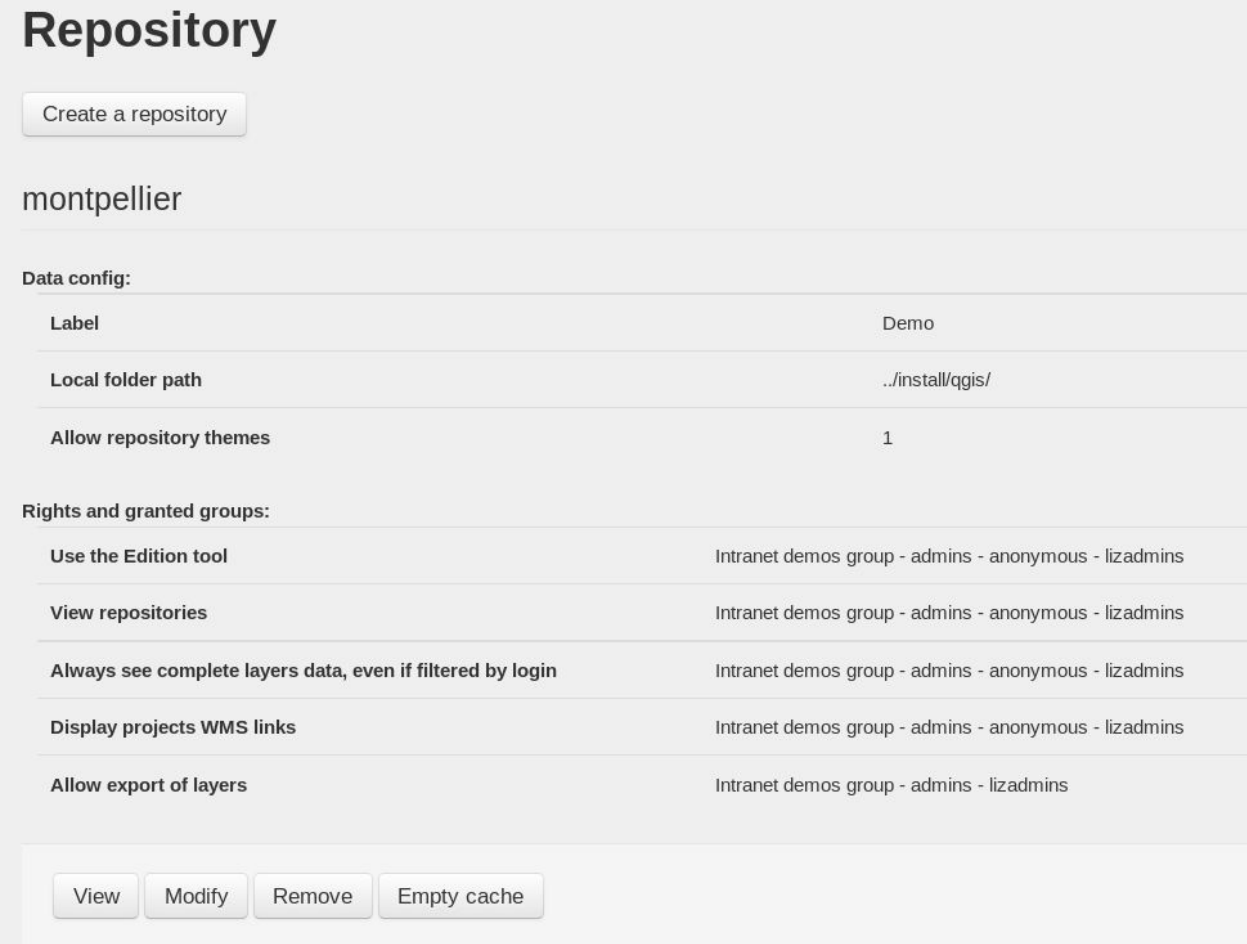
You should check the checkbox "Use a proxy server" and fill these fields:

- **Host of the proxy server**: the address of the proxy
- **Port of the proxy server**: the tcp port of the proxy
- **Type of the proxy**: the protocol used to access to the proxy: http or socks5. It is only supported when curl is used.
- **Login** and **password**: sometimes an authentication is needed to access to the proxy
- **Domains for which the proxy will not be used**: there are some requests that should not go to the proxy, typically requests to your QGIS server, or requests to a local server that serves map backgrounds etc… You have at least to set it with "localhost,127.0.0.1".

4.3.3 Repositories

For each Lizmap repository, are listed:

- **The main informations**: label and path
- **The list of rights** with the groups concerned
- **Action buttons**:
  - *View*: displays a page that lists the maps in this repository
  - *Modify*: displays the repository modification form
  - *Remove*: delete the repository from the Lizmap Web Client configuration
  - *Empty cache*: deletes the entire cache of all layers of the repository projects



You can create a new repository using the button **Create a repository** located at the top and bottom of the section *Repositories*.

Add a repository

To create a repository, you must give:

- **an id**: a word without spaces, accents or special characters
- **a label**: the name that will be displayed for this repository, accents and spaces allowed
- **a local folder path**: the full path on the server to the folder that contains the QGIS projects and data
- **Allow repository themes**: This option enables the possibility for the publisher to set a theme for the repository and themes for each map. See *Creating simple themes*.

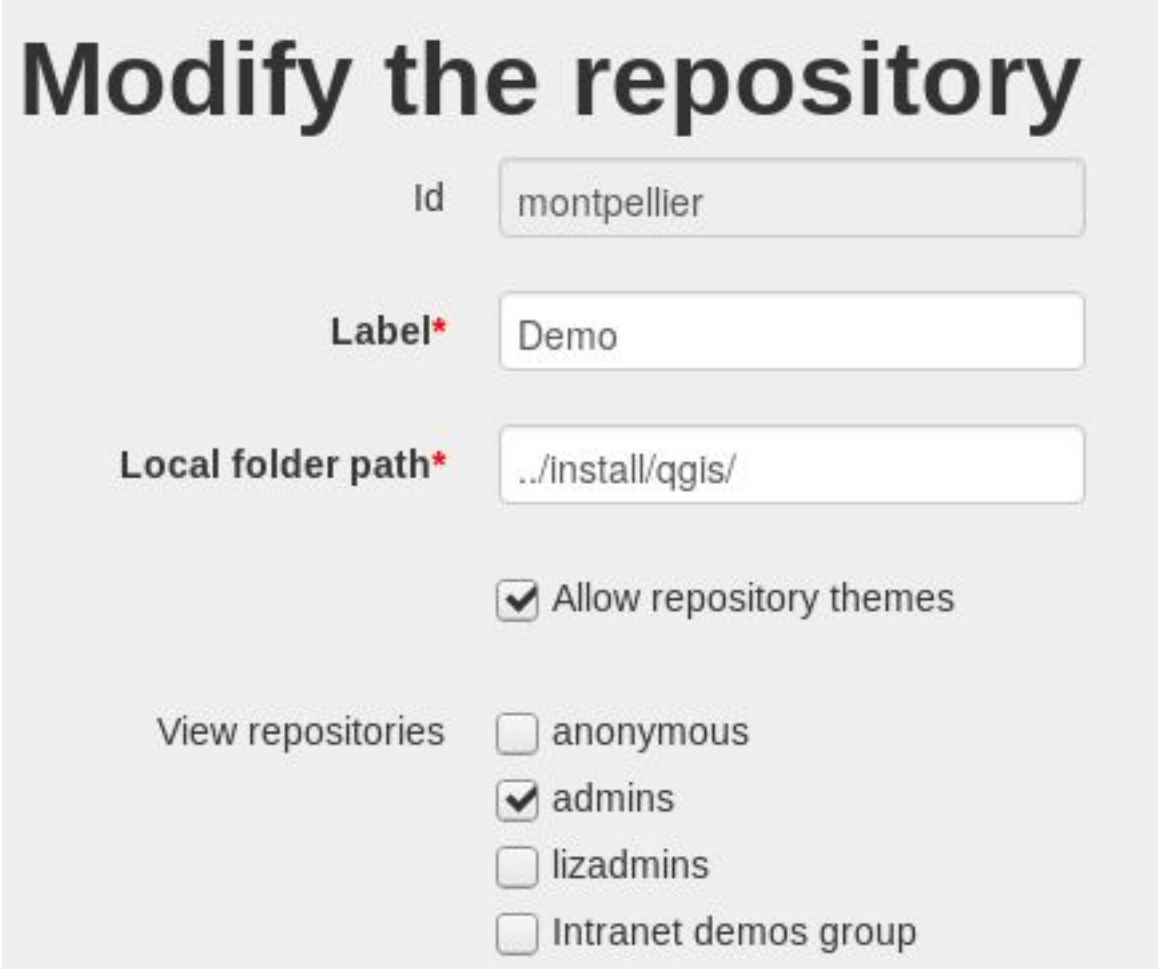
Define the rights for each group

After creating the directory, the repository modification form is displayed automatically and defines the following rights for each group:

- **View repository** :
  - all users of checked groups can access maps of this repository
  - the group *anonymous* is unregistered users and allows to make public maps
- **Use the Edition tool**

  When this option is checked, the group users have access to the editing tool for all of Lizmap repository maps for which the edition was configured.
- **Always see complete layers data, even if filtered by login**

  This option is in connection with the filtering feature data layers per group. See *Filtered layer by user*. Check the box for deciding which groups can see all the time all the data, even when a filter is active on some layers.



4.4 Cache management as administrator

Contents

- *Cache management as administrator*
  - *Remove all the cache by Lizmap repository*
  - *Delete the cache, layer by layer, for each Lizmap project*
  - *Configuring the caching system*
  - *Seeding*

For a layer with server cache activated, Lizmap will keep the cache when the user pan and zoom on the map. To have better performance, Lizmap Web Client can also automatically generate the tile cache on the server.

In some cases, it is desirable to remove the server cache, for example, when the style has changed and the tiles need to be updated. For this, two solutions are possible:

4.4.1 Remove all the cache by Lizmap repository

In the administration interface, in the *Maps management* page, for each repository, the administrator can delete the cache for all layers of all the projects repository by clicking the button *Empty cache*.

4.4.2 Delete the cache, layer by layer, for each Lizmap project

When the administrator is connected and consults a Lizmap map, a **little red cross** is displayed on the right of the name of each layer that is configured with server cache. Clicking on the cross allows, after confirmation, to delete the server cache only for this layer in the QGIS project.

Only logged administrator sees these red crosses and has the right to delete the cache.

4.4.3 Configuring the caching system

In the *Administration -> Lizmap Configuration -> Cache*, you can configure the caching system which is used. You can choose between different kind of caching :

- Files
- Sqlite
- Redis (You would need a Redis server)

4.4.4 Seeding

You can pre-generate the tiles for any layers of a QGIS project configured with server-side caching on. You need to have full access on the server where Lizmap is installed and use a terminal to connect to it. You also need to know the ids of Lizmap repositories and the code name of the project (the QGIS project filename without the extension).

In this example, we will show commands to manage the tiles cache for the demo project `Montpellier`, shipped with Lizmap Web Client. We also assume that your Lizmap application is installed in the folder `/var/www/lizmap-web-client/`.

```
# Go to the application folder
cd /var/www/lizmap-web-client/
```

It is important to know that Lizmap publish the cached layers in WMTS (Web Map Tiled Service). The following concepts are used as options of Lizmap tile cache seeder:

- **TileMatrixSet** - In Lizmap, this represents the projection code, for example *EPSG:3857* (Pseudo mercator).
- **TileMatrixMin** - This is the minimum zoom level.
- **TileMatrixMax** - This is the maximum zoom level.

Warning: The zoom level is **not** a map scale, but the the zoom level. In Lizmap plugin, the project publisher can configure scales for the published project, for example the list : *100000, 50000, 25000, 10000*. The zoom level ID depends on some CRS and how you configured your Lizmap project. You can have a idea of the scale ID by typing *lizMap.map.getZoom()* in your webbrowser Javascript console when zooming on your map.

First you **must** get the cache capabilities of one project and some details on a specific layer.

```
# Command help
# php lizmap/scripts/script.php lizmap~wmts:capabilities [-v] repository project␣
↪[layer] [TileMatrixSet]

# Get the capabilities for a given project published with Lizmap
# and generate the cache about these capabilities.
php lizmap/scripts/script.php lizmap~wmts:capabilities montpellier montpellier

# Get precisions about a specific layer
php lizmap/scripts/script.php lizmap~wmts:capabilities -v montpellier montpellier bus␣
↪EPSG:3857
# which will return
For "bus" and "EPSG:4326" from TileMatrix 13 to 15
For "bus" and "EPSG:900913" from TileMatrix 14 to 16
For "bus" and "EPSG:3857" from TileMatrix 14 to 16
```

In this example, you see that the bus layer has 3 different TileMatrixSet, corresponding to the 3 different spatial coordinate systems available for this project in Lizmap (configured in the *Project properties ▸ QGIS Server*).

Note: If your layer name has some spaces, you need to use *“”*.

It's important to generate the cache capabilities **before** generating the cache for a specific layer. The cache capabilities is used in the next command. The next command might fail if the cache capabilities is not present.

Once you have a good knowledge of a layer, you can generate the cache for it:

```
# Command help
# php lizmap/scripts/script.php lizmap~wmts:seeding [-v] [-f] repository project␣
↪layer TileMatrixSet TileMatrixMin TileMatrixMax

# Example
php lizmap/scripts/script.php lizmap~wmts:seeding -v -f montpellier montpellier bus␣
↪EPSG:3857 12 14
# Which will return:
81 tiles to generate for "bus" "EPSG:3857" "14"
81 tiles to generate for "bus" "EPSG:3857" between "12" and "14"
Start generation
================
Progression: 6%, 5 tiles generated on 81 tiles
Progression: 12%, 10 tiles generated on 81 tiles
Progression: 18%, 15 tiles generated on 81 tiles
Progression: 24%, 20 tiles generated on 81 tiles
Progression: 30%, 25 tiles generated on 81 tiles
Progression: 37%, 30 tiles generated on 81 tiles
Progression: 43%, 35 tiles generated on 81 tiles
Progression: 49%, 40 tiles generated on 81 tiles
Progression: 55%, 45 tiles generated on 81 tiles
Progression: 61%, 50 tiles generated on 81 tiles
Progression: 67%, 55 tiles generated on 81 tiles
Progression: 74%, 60 tiles generated on 81 tiles
Progression: 80%, 65 tiles generated on 81 tiles
Progression: 86%, 70 tiles generated on 81 tiles
Progression: 92%, 75 tiles generated on 81 tiles
Progression: 98%, 80 tiles generated on 81 tiles
================
End generation
```

After seeding, update rights on cache :

```
lizmap/install/set_rights.sh www-data www-data
```

4.5 Log management in Lizmap

Contents

- *Log management in Lizmap*
  - *Principle and description*
  - *Configure logs*
  - *View logs*
  - *Log storage file*

4.5.1 Principle and description

Since version 2.8, you can configure *Lizmap Web Client* so that certain user actions are stored in a SQLite database:

- User Login
- Displaying a Lizmap map
- Printing a map
- Displaying a popup
- Using the editing tool

For each of these actions can be chosen:

- to record a new line in the logs containing various information: user, date and time, action, Lizmap repository, QGIS project, IP address
- to increment the counter for this action, the Lizmap repository and QGIS project.

4.5.2 Configure logs

For now, it is not possible to change the configuration of logs in the administration interface. It is necessary to manually edit the configuration file **lizmap/var/config/lizmapLogConfig.ini.php**. This file is in *ini* format and contains many sections as action to save. For each action, you can choose to activate with *on* or off with *off* the recording of the log.

For example, the following section shows that the administrator has chosen to record a count in the logs every time a user connects. But he does not want to save the details for each connection.

```
[item:login]
label="User logs in"
logCounter=on
logDetail=off
logIp=off
```

4.5.3 View logs

To view logs, simply connect to the Lizmap administration interface as an administrator. Then the logs can be accessed through the menu **Lizmap Logs**. This page shows the general statistics on the 2 log tables: *Log count* and *Log detail*. For each, it is possible to:

- **View Tables** containing the raw data
- **Completely empty logs**: completly reset!

4.5.4 Log storage file

The log database is located here in relation to the installation directory: **lizmap/var/logs.db**. For example:

```
# ifLizmap  Web Client is installed here : /var/www/lizmap-web-client-2.8.1/, the␣
↪file is:
/var/www/lizmap-web-client-2.8.1/lizmap/var/logs.db
```

This database can be accessed with a Sqlite database reading tool, like *SQLite Browser* or the Firefox add-on *SQlite Manager*. If you know SQL, so you can make queries to extract information from the detailed logs.

CHAPTER

FIVE

INSTALLING AND UPGRADING LIZMAP

For more information on versions (Web-browsers, QGIS Server, PostgreSQL etc) which are supported, it's written on the Lizmap Web Client wiki page : https://github.com/3liz/lizmap-web-client/wiki/Versions

5.1 Requirements before installing Lizmap Web Client

Note: If you want to quickly install and test Lizmap Web Client in a few steps, you can follow those instructions.

5.1.1 QGIS Server

Warning: Before installing the QGIS Server part, it is **very highly** recommended to use the **same** version between QGIS Desktop and QGIS Server.

Even if some functions might work, there is a probability that some configuration won't work if these two versions are **different**.

The reason is QGIS Server version X might not be able to a QGIS project made with a QGIS Desktop version Y.

Follow the QGIS Documentation how to install QGIS Server : https://docs.qgis.org/latest/en/docs/server_manual/

Using a webserver (Apache or Nginx), you must install QGIS Server. With Nginx, the preferred way is to use `spawn-fcgi`. Do **not** use the `fcgiwrap`, this solution is not efficient.

In the Nginx configuration, it's good to use the `QGIS_OPTIONS_PATH` variable for a folder with write permissions for `www-data`. These is explained in the QGIS Server documentation.

You should also install and configure `XVFB` mentioned in the QGIS Documentation. This is useful for printing PDF. You can only skip this section if you don't plan to print PDF on the server side.

After you have setup your web server with QGIS-Server, check that the URL of QGIS Server is working. You probably get a XML like:

```
<ServerException>Project file error</ServerException>
```

Keep this URL, we will use it later in the Lizmap admin panel.

Warning: We **strongly** encourage you to set up QGIS Server on a different virtual host than Lizmap Web Client. QGIS Server URL should stay private, accessible by the Lizmap PHP application **only**.

Otherwise, especially after the **Lizmap** plugin on QGIS Server is installed, your user might be able to access private data if they by-pass Lizmap, by using straight QGIS Server URL.

QGIS Server plugins

Some plugins can be added to QGIS Server. This will enable some features in Lizmap. It's not compulsory but in some situations, it's better.

Either you should setup the `QGIS_PLUGINPATH` environment variable during the installation of QGIS Server or use the default one provided by QGIS. https://docs.qgis.org/latest/en/docs/server_manual/config.html#environment-variables

Tip: To manage QGIS Server plugins, we encourage you to use **qgis-plugin-manager**, a CLI tool to install and upgrade plugins. https://pypi.org/project/qgis-plugin-manager/

- AtlasPrint
  - To enable the PDF based on a QGIS Layout Atlas
  - https://github.com/3liz/qgis-atlasprint
- Cadastre
  - French use-case only
  - Needed for the Lizmap Cadastre module
  - https://docs.3liz.org/QgisCadastrePlugin/module-lizmap/
- Lizmap
  - Lizmap **is not only** a PHP application, there is also Python plugin for **QGIS Server**.
  - https://github.com/3liz/lizmap-plugin/
  - The plugin is highly recommended, a few features won't work as expected if the plugin is not installed or not upgraded.
  - **Important**, read below for more information the Lizmap QGIS Server plugin. (*Lizmap QGIS Server plugin*)
- WfsOutputExtension
  - To add new format when exporting vector data
  - https://github.com/3liz/qgis-wfsOutputExtension
- Logging
  - To log QGIS Servers log and to flush the cache on QGIS Server
  - https://github.com/3liz/qgis-logging-plugin.
  - This plugin is deprecated.

Lizmap QGIS Server plugin

Adding the Lizmap QGIS Server plugin will add some features on Lizmap Web Client :

- retrieve information from QGIS Server.
- evaluate QGIS Expressions in forms about :
  - constraints
  - default value
  - group visibility
  - Read *Advanced form*.
- check User Access Rights (ACL) for features and layers :
  - filter by polygon
  - by attribute
- use the **Form** popup, read *Form popup*.
- enable `@lizmap_user` and `lizmap_user_groups` variables in QGIS projects

Warning: Starting from Lizmap plugin 3.7.0 on the server, the plugin will require you to **add** a environment variable on QGIS server.

For **security** reason, to enable all features on the QGIS server side, you must enable the environment variable `QGIS_SERVER_LIZMAP_REVEAL_SETTINGS=True` on QGIS server.

You **must** be ensured that this API `http://your.qgis.server.url/lizmap/server.json` is protected on your webserver. The **best** is to restrict the access to QGIS server `http://your.qgis.server.url` on a virtual host, not accessible on the internet. All requests to QGIS server will be sent by Lizmap Web Client. QGIS server mustn't be accessible from outside. It was already **highly** recommended before to protect the QGIS Server from the internet. Users **must use** WFS/WMS links provided by Lizmap Web Client, so Lizmap can check user permissions.

This variable will **expose** server settings such as QGIS server version, which is used by Lizmap Web Client.

If your are using QGIS Server with FCGI, the Lizmap API URL must be empty. Otherwise, if you are using Py-QGIS-Server, the Lizmap API URL must be configured in the administration interface.

In Py-QGIS-Server, you must explicitly publish the API as well :

```
[api.endpoints]
lizmap_api=/lizmap

[api.enabled]
lizmap_api=yes
```

PostgreSQL

PostgreSQL can be used for two different purposes in Lizmap :

- To store GIS data. No configuration is needed on the Lizmap Web Client server side, **only** the PostgreSQL server must be accessible from the Lizmap Web Client server. It's possible to edit layers with Lizmap, but the layer **must** be stored in PostgreSQL. See *Prerequisites*.
- To store Lizmap Web Client users and user actions. Lizmap uses tables. This setting must be done when **installing** Lizmap.

5.2 Installing Lizmap Web Client on Linux Debian or Ubuntu

Note: If you want to quickly install and test Lizmap Web Client in a few steps, you can follow those instructions using Docker and Docker-Compose.

Note: In Debian distributions, you can work as administrator (log in with `root`), without using `sudo` on contrary to Ubuntu.

5.2.1 Generic Server Configuration with Nginx server

This documentation provides an example for configuring a server with the Debian 9 distribution. We assume you have base system installed and updated.

Warning: This page does not describe how to secure your Nginx server. It's just for a demonstration.

Configure Locales

For simplicity, it is interesting to configure the server with UTF-8 default encoding.

```
# configure locales
locale-gen fr_FR.UTF-8 #replace fr with your language
dpkg-reconfigure locales
# define your timezone [useful for logs]
dpkg-reconfigure tzdata
apt install ntp ntpdate
```

Note: It is also necessary configure the other software so that they are using this default encoding if this is not the case.

5.2.2 Nginx Server Configuration

This documentation provides an example for configuring a server with the Debian 9 distribution. We assume you have base system installed and updated.

> **Warning:** This page does not describe how to secure your Nginx server. It's just for a demonstration.

Installing necessary packages

> **Warning:** Lizmap web client is based on Jelix 1.6. You must install at least the **5.6** version of PHP. The **dom**, **simplexml**, **pcre**, **session**, **tokenizer** and **spl** extensions are required (they are generally turned on in a standard PHP 5.6/7.x installation)

```
sudo su # only necessary if you are not logged in as root
apt update # update packages list
apt-get install curl openssl libssl1.1 nginx-full nginx nginx-common
```

On debian 10, install these packages:

```
apt-get install php7.3-fpm php7.3-cli php7.3-bz2 php7.3-curl php7.3-gd php7.3-intl␣
↪php7.3-json php7.3-mbstring php7.3-pgsql php7.3-sqlite3 php7.3-xml php7.3-ldap
```

On Ubuntu 18.04 or later, install these packages:

```
apt-get install php7.3-fpm php7.3-cli php7.3-bz2 php7.3-curl php7.3-gd php7.3-intl␣
↪php7.3-json php7.3-mbstring php7.3-pgsql php7.3-sqlite3 php7.3-xml php7.3-ldap
```

Web configuration

Create a new file /etc/nginx/sites-available/lizmap.conf:

```
server {
    listen 80;

    server_name localhost;
    root /var/www/html/lizmap;
    index index.php index.html index.htm;

    # compression setting
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 5;
    gzip_min_length 100;
    gzip_http_version 1.1;
    gzip_types text/plain text/css application/json application/javascript text/xml␣
↪application/xml application/xml+rss text/javascript text/json;

    location / {
        try_files $uri $uri/ =404;
    }
```

(continues on next page)

(continued from previous page)

```
    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+\.php)(/.*)$;
        set $path_info $fastcgi_path_info; # because of bug http://trac.nginx.org/
↪nginx/ticket/321
        try_files $fastcgi_script_name =404;
        include fastcgi_params;

        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $path_info;
        fastcgi_param PATH_TRANSLATED $document_root$path_info;
        fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
        fastcgi_param SERVER_NAME $http_host;
    }
}
```

You should declare the lizmap.local domain name somewhere (in your /etc/hosts, or into your DNS..), or replace it by your own domain name.

Enable the virtual host you just created:

```
ln -s /etc/nginx/sites-available/lizmap.conf /etc/nginx/sites-enabled/lizmap.conf
```

Restart Nginx

You must restart the Nginx server to validate the configuration.

```
service nginx restart
```

5.2.3 Apache Server configuration

This documentation provides an example for configuring a server with the Debian 10 distribution. We assume you have base system installed and updated.

Warning: This page does not describe how to secure your Apache server. It's just for a demonstration.

Installing necessary packages

Firstly update the packages list, then install these packages:

```
sudo su # only necessary if you are not logged in as root
apt update
apt-get install xauth htop curl apache2 libapache2-mod-fcgid
apt-get install libapache2-mod-php7.3 php7.3-cgi php7.3-gd php7.3-sqlite php7.3-curl␣
↪php7.3-xmlrpc php7.3-xml python-simplejson software-properties-common
```

PHP 7.3 configuration

In this example, we use Apache mpm-worker. So we must manually configure the activation of PHP 7.3.

```
# Create the configuration file
nano /etc/apache2/conf-available/php.conf
# Copy the following text in it
<Directory /usr/share>
   AddHandler fcgid-script .php
   FCGIWrapper /usr/lib/cgi-bin/php7.3 .php
   Options ExecCGI FollowSymlinks Indexes
</Directory>

<Files ~ (\.php)>
   AddHandler fcgid-script .php
   FCGIWrapper /usr/lib/cgi-bin/php7.3 .php
   Options +ExecCGI
   allow from all
</Files>
```

Enable the configuration with the following command line:

```
a2enconf php
```

Web configuration

mpm-worker configuration

We modify the Apache configuration file to adapt the options to mpm_worker server configuration.

```
nano /etc/apache2/apache2.conf
<IfModule mpm_worker_module>
StartServers       4
MinSpareThreads    25
MaxSpareThreads    100
ThreadLimit          64
ThreadsPerChild      25
MaxClients        150
MaxRequestsPerChild   0
</IfModule>
```

mod_fcgid configuration

QGIS Server runs with the FastCGI protocole (a.k.a. fcgi). We must therefore configure the Apache mod_fcgid to suit to the server capabilities.

```
# Open the mod_fcgid configuration file
 nano /etc/apache2/mods-enabled/fcgid.conf
 # Paste the following content and adapt it
 <IfModule mod_fcgid.c>
    AddHandler    fcgid-script .fcgi
    FcgidConnectTimeout 300
    FcgidIOTimeout 300
    FcgidMaxProcessesPerClass 50
```

(continues on next page)

(continued from previous page)

```
    FcgidMinProcessesPerClass 20
    FcgidMaxRequestsPerProcess 500
    IdleTimeout   300
    BusyTimeout   300
 </IfModule>
```

Setting the compression

```
nano /etc/apache2/conf-available/mod_deflate.conf
# Add the bellow text in the file
<Location />
   # Insert filter
   SetOutputFilter DEFLATE
   # Netscape 4.x encounters some problems ...
   BrowserMatch ^Mozilla/4 gzip-only-text/html
   # Netscape 4.06-4.08 encounter even more problems
   BrowserMatch ^Mozilla/4\.0[678] no-gzip
   # MSIE pretends it is Netscape, but all is well
   BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
   # Do not compress images
   SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
   # Ensure that proxy servers deliver the right content
   Header append Vary User-Agent env=!dont-vary
</Location>
```

Restart Apache

You must restart the Apache server to validate the configuration.

```
service apache2 restart
# or
systemctl restart apache2
```

5.2.4 Enable geolocation

The automatic geolocation provided by Lizmap relies on Google services. To enable it, your webGIS must be placed under a secure protocol, like HTTPS. See for more details:

https://sites.google.com/a/chromium.org/dev/Home/chromium-security/deprecating-powerful-features-on-insecure-origins

https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04

5.2.5 Create directories for data

QGIS files and other cache files will be stored into these directories.

```
 mkdir /home/data
 mkdir /home/data/cache/
# optional
 mkdir /home/data/ftp
 mkdir /home/data/ftp/template/
 mkdir /home/data/ftp/template/qgis
```

5.2.6 Spatial Database: PostgreSQL

Note: This section is optional, but required if you want to enable editing capabilities on a layer. See *Prerequisites*.

PostgreSQL and PostGIS can be very useful to manage spatial data centralized manner on the server.

Install

```
# Install packages
apt-get install postgresql postgresql-contrib postgis pgtune

# A cluster is created in order to specify the storage directory
mkdir /home/data/postgresql
service postgresql stop
pg_dropcluster --stop 9.6 main
chown postgres:postgres /home/data/postgresql
pg_createcluster 9.6 main -d /home/data/postgresql --locale fr_FR.UTF8 -p 5678 --start

# Creating a "superuser" user
su - postgres
createuser myuser --superuser
# Modifying passwords
psql
ALTER USER postgres WITH ENCRYPTED PASSWORD '************';
ALTER USER myuser WITH ENCRYPTED PASSWORD '************';
\q
exit
```

Adapting the PostgreSQL configuration

We will use `pgtune`, an utility program that can automatically generate a PostgreSQL configuration file adapted to the properties of the server (memory, processors, etc.)

```
# PostgreSQL Tuning with pgtune
pgtune -i /etc/postgresql/9.6/main/postgresql.conf -o /etc/postgresql/9.6/main/
↪postgresql.conf.pgtune --type Web
cp /etc/postgresql/9.6/main/postgresql.conf /etc/postgresql/9.6/main/postgresql.conf.
↪backup
cp /etc/postgresql/9.6/main/postgresql.conf.pgtune /etc/postgresql/9.6/main/
↪postgresql.conf
```

(continues on next page)

(continued from previous page)

```
nano /etc/postgresql/9.6/main/postgresql.conf
# Restart to check any problems
service postgresql restart
# If error messages, increase the linux kernel configuration variables
echo "kernel.shmall = 4294967296" >> /etc/sysctl.conf # to increase shred buffer␣
↪param in kernel
echo "kernel.shmmax = 4294967296" >> /etc/sysctl.conf
echo 4294967296 > /proc/sys/kernel/shmall
echo 4294967296 > /proc/sys/kernel/shmmax
sysctl -a | sort | grep shm
# Restart PostgreSQL
service postgresql restart
```

5.2.7 FTP Server: pure-ftpd

Note: This section is optional

Install

```
apt-get install pure-ftpd pure-ftpd-common
```

Configure

```
# Creating an empty shell for users
ln /bin/false /bin/ftponly
# Configuring FTP server
echo "/bin/ftponly" >> /etc/shells
# Each user is locked in his home
echo "yes" > /etc/pure-ftpd/conf/ChrootEveryone
# Allow to use secure FTP over SSL
echo "1" > /etc/pure-ftpd/conf/TLS
# Configure the properties of directories and files created by users
echo "133 022" > /etc/pure-ftpd/conf/Umask
# The port range for passive mode (opening outwards)
echo "5400 5600" > /etc/pure-ftpd/conf/PassivePortRange
# Creating an SSL certificate for FTP
openssl req -x509 -nodes -newkey rsa:1024 -keyout /etc/ssl/private/pure-ftpd.pem -out␣
↪/etc/ssl/private/pure-ftpd.pem
chmod 400 /etc/ssl/private/pure-ftpd.pem
# Restart FTP server
service pure-ftpd restart
```

Creating a user account

```
# Creating a user account
MYUSER=demo
useradd -g client -d /home/data/ftp/$MYUSER -s /bin/ftponly -m $MYUSER -k /home/data/
↪ftp/template/
passwd $MYUSER
# Fix the user's FTP root
chmod a-w /home/data/ftp/$MYUSER
# Creating empty directories that will be the future Lizmap Web Client directories
mkdir /home/data/ftp/$MYUSER/qgis/rep1 && chown $MYUSER:client /home/data/ftp/$MYUSER/
↪qgis/rep1
mkdir /home/data/ftp/$MYUSER/qgis/rep2 && chown $MYUSER:client /home/data/ftp/$MYUSER/
↪qgis/rep2
mkdir /home/data/ftp/$MYUSER/qgis/rep3 && chown $MYUSER:client /home/data/ftp/$MYUSER/
↪qgis/rep3
mkdir /home/data/ftp/$MYUSER/qgis/rep4 && chown $MYUSER:client /home/data/ftp/$MYUSER/
↪qgis/rep4
mkdir /home/data/ftp/$MYUSER/qgis/rep5 && chown $MYUSER:client /home/data/ftp/$MYUSER/
↪qgis/rep5
# Create a directory to store the cached server
mkdir /home/data/cache/$MYUSER
chmod 700 /home/data/cache/$MYUSER -R
chown www-data:www-data /home/data/cache/$MYUSER -R
```

5.2.8 Installing sources of Lizmap Web Client

Retrieve the latest available stable version from our Github release page.

Warning: Do not use the automatic ZIP file created by GitHub on the website. Only use ZIP attached to a release.

```
# Options
# Check the latest version available, maybe it's not 3.5.1 anymore
# https://github.com/3liz/lizmap-web-client/releases
VERSION=3.5.1
# chose location where download your zip (e.g. /var/www or your home)
LOCATION=/var/www
# Archive recovery with wget
cd $LOCATION
wget https://github.com/3liz/lizmap-web-client/releases/download/$VERSION/lizmap-web-
↪client-$VERSION.zip
# Unzip archive
unzip $VERSION.zip

# virtual link for http://localhost/lizmap/
ln -s $LOCATION/lizmap-web-client-$VERSION/lizmap/www/ /var/www/html/lizmap
# Remove archive
rm $VERSION.zip
```

5.2.9 Configure Lizmap with the database support

Lizmap needs a database to store its own data and to access to data used in your Qgis projects, with its editing tool.

Create `profiles.ini.php` into `lizmap/var/config` by copying `profiles.ini.php.dist`.

```
cd lizmap/var/config
cp profiles.ini.php.dist profiles.ini.php
cd ../../..
```

PostgreSQL

For the editing of PostGIS layers in Web Client Lizmap operate, install PostgreSQL support for PHP.

```
sudo apt-get install php7.3-pgsql
sudo service nginx restart
```

For Lizmap logs, users and groups, it can be either stored in SqLite or PostgreSQL. To store these information in PostgreSQL, follow these instructions.

Into a fresh copy of `lizmap/var/config/profiles.ini.php`, you should have:

```
[jdb:jauth]
driver=sqlite3
database="var:db/jauth.db"

[jdb:lizlog]
driver=sqlite3
database="var:db/logs.db"
```

This is the configuration by default to use Sqlite. You should change these sections to use Postgresql, and indicate several parameters to access to your Postgresql database:

```
[jdb:jauth]
driver=pgsql
host=localhost
port=5432
database="your_database"
user=my_login
password=my_password
search_path=public

[jdb:lizlog]
driver=pgsql
host=localhost
port=5432
database="your_database"
user=my_login
password=my_password
search_path=public
```

You can use a specific schema to store lizmap tables. And you may want that lizmap could access to other schema. You then have to set search_path correctly. Example:

```
search_path=lizmap,my_schema,public
```

If you have setup a service file for postgresql onto your server, you may want to indicate a postgresql service instead of indicating login, password and so on. Use then the service parameter:

```
[jdb:jauth]
driver=pgsql
service=my_service
database="your_database"
search_path=lizmap,public

[jdb:lizlog]
driver=pgsql
service=my_service
database="your_database"
search_path=lizmap,public
```

Spatialite

Enable Spatialite extension

To use editing on layers spatialite,you have to add the spatialite extension in PHP. You can follow these instructions to do so: http://www.gaia-gis.it/gaia-sins/spatialite-cookbook-fr/html/php.html

Lizmap Web Client tests whether the spatialite support is enabled in PHP. If it is not, then spatialite layers will not be used in the editing tool. You can always use PostgreSQL data for editing.

Give the appropriate rights to the directory containing Spatialite databases

So that Lizmap Web Client can modify the data contained in databases Spatialite, we must ensure that **the webserver user (www-data) has well write access to the directory containing each Spatialite file**

For example, if a directory contains a QGIS project, which uses a Spatialite database placed in a **db** directory at the same level as the QGIS project:

```
/path/to/a/lizmap_directory
|--- mon_projet.qgs
|--- bdd
   |--- my_spatialite_file.sqlite
```

So you have to give the rights in this way:

```
chown :www-data /path/to/a/lizmap_directory -R
chmod 775 /path/to/a/lizmap_directory -R
```

Note: So if you want to install Lizmap to provide access to multiple map publishers, you should tell them to always create a **db** directory at the same level as the QGIS projects in the Lizmap Web Client directory. This will facilitate the admin work that just have to change the rights of this unique directory.

5.2.10 Configuring Lizmap and launching the installer

Give the appropriate rights to directories and files

Set rights for Nginx/Apache, so PHP scripts could write some temporary files or do changes.

```
cd /var/www/lizmap-web-client-$VERSION/
lizmap/install/set_rights.sh www-data www-data
chown :www-data lizmap/install/qgis/edition/ -R
chmod 775 lizmap/install/qgis/edition/ -R
```

Setup configuration

Create `lizmapConfig.ini.php`, `localconfig.ini.php` and edit them to set parameters specific to your installation. You can modify `lizmapConfig.ini.php` to set the url of qgis map server and other things.

```
cd lizmap/var/config
cp lizmapConfig.ini.php.dist lizmapConfig.ini.php
cp localconfig.ini.php.dist localconfig.ini.php
cd ../../..
```

In case you want to enable the demo repositories, just add to `localconfig.ini.php` the following:

```
[modules]
lizmap.installparam=demo
```

Launching the installer

After creating configuration files, you can launch the installer

```
php lizmap/install/installer.php
```

It will finished the installation, and will create all SQL tables needed by Lizmap.

First test

For testing launch: `http://localhost/lizmap` in your browser.

In case you get a `500 - internal server error`, run again:

```
cd /var/www/lizmap-web-client-$VERSION/
lizmap/install/set_rights.sh www-data www-data
```

Note: Replace `localhost` with the address or IP number of your server.

In the administration panel, you should check the *QGIS server version* and the *WMS server URL* with the URL of QGIS Server.

If you didn't install the demo, you can check that you have well installed Lizmap and configured QGIS Server within Lizmap by checking the `qgis_server` section in this URL: http://localhost/lizmap/index.php/view/app/metadata

```
{
    "qgis_server":{
        "test":"OK",
        "mime_type":"text\/xml; charset=utf-8"
    }
}
```

Lizmap is accessible, without further configurations, also as WMS and WFS server from a browser:

http://localhost/lizmap/index.php/lizmap/service/?repository=montpellier&project=montpellier&VERSION=1.3.0&SERVICE=WMS&REQUEST=GetCapabilities

http://localhost/lizmap/index.php/lizmap/service/?repository=montpellier&project=montpellier&SERVICE=WFS&REQUEST=GetCapabilities

and from QGIS:

http://localhost/lizmap/index.php/lizmap/service/?repository=montpellier&project=montpellier&VERSION=1.3.0&

http://localhost/lizmap/index.php/lizmap/service/?repository=montpellier&project=montpellier&

Note: Access to the WMS and WFS servers can be limited by assigning privileges to specific repositories, see the administration section.

5.2.11 Lizmap modules

Previously, we explained how we could add QGIS Server plugins to add more features to QGIS Server. Now that we have Lizmap Web Client up and running, we can add some Lizmap modules to add again some features.

The list is available in the Lizmap *introduction*. On their GitHub repository, their is usually their install instructions. You should follow them. However here are the main instructions to install a module.

Installing modules with Composer

You can install modules with Composer, the package manager for PHP. Of course it is possible only if the author of the module has created a package of his module. A such package has a name, for example *lizmap/lizmap-pgmetadata-module*`. The documentation of the module should indicate it.

You must install Composer. See instructions on its web site http://getcomposer.org.

You must create a `composer.json` file into `lizmap/my-packages/` by copying the `composer.json.dist` from this directory. And launching a first time Compose

```
cp -n lizmap/my-packages/composer.json.dist lizmap/my-packages/composer.json
composer install --working-dir=lizmap/my-packages
```

Then you can install the package of the module

```
composer require --working-dir=lizmap/my-packages "lizmap/lizmap-pgmetadata-module"
```

If you want to install a new version of the module, just execute:

```
composer update --working-dir=lizmap/my-packages
```

Read the documentation of the module to know if there are additional steps to configure it.

To finish the installation, run again the installer of Lizmap:

```
php lizmap/install/installer.php
lizmap/install/clean_vartmp.sh
lizmap/install/set_rights.sh
```

installing modules without Composer

To install a module without Composer, retrieve the zip file of the module.

- Extract the module into `lizmap/lizmap-modules/`. For instance, for the module `PgMetadata` :

```
$ ls -hl lizmap/lizmap-modules/pgmetadata/
total 44K
drwxrwxr-x 2 etienne etienne 4,0K nov.  17 12:38 classes
drwxrwxr-x 2 etienne etienne 4,0K nov.   4 12:50 controllers
drwxrwxr-x 2 etienne etienne 4,0K nov.   4 10:09 daos
-rw-rw-r-- 1 etienne etienne  146 nov.   4 10:38 events.xml
drwxrwxr-x 2 etienne etienne 4,0K nov.   4 10:09 forms
drwxrwxr-x 2 etienne etienne 4,0K nov.   4 12:50 install
drwxrwxr-x 4 etienne etienne 4,0K nov.   4 10:09 locales
-rw-rw-r-- 1 etienne etienne  789 nov.  19 16:02 module.xml
drwxrwxr-x 2 etienne etienne 4,0K nov.   4 10:09 templates
-rw-rw-r-- 1 etienne etienne  106 nov.   4 10:39 urls.xml
drwxrwxr-x 2 etienne etienne 4,0K nov.  17 12:38 www
```

- Edit the `lizmap/var/config/localconfig.ini.php`, in the `modules` section, add a new line for the given module (check the documentation of the module to setup the correct values):

```
[modules]
pgmetadata.access=2
```

- Read the documentation of the module to know if there are additional steps to configure it.
- Run the installation :

```
php lizmap/install/installer.php
lizmap/install/clean_vartmp.sh
lizmap/install/set_rights.sh
```

5.3 Installing Lizmap Web Client on Windows

Lizmap Web Client is not officially supported on a Windows server.

We recommend either :

- using the tutorial and configuration from NaturalGIS.
- using Docker and Docker-Compose to quickly test Lizmap Web Client.

5.4 Configuration of Lizmap

In some server, additionnal settings should be done into Lizmap. Settings should be set into the `lizmap/var/config/` directory.

5.4.1 Configuration files

There are several configuration files for Lizmap. There are into lizmap/var/config/.

Framework configuration files:

- `mainconfig.ini.php` contains many configuration parameters, mainly for the framework used by Lizmap. You may want to modify some of them, like the available languages etc. **DON'T MODIFY mainconfig.ini.php**. Put parameters with their new values into `localconfig.ini.php` instead!
- `localconfig.ini.php` contains configuration parameters that are specific to your installation. So you put into it any parameters you can find into `mainconfig.ini.php` and you want to change.
- `liveconfig.ini.php` is containing parameters from `mainconfig.ini.php` that are changed by the application itself.

During the execution of Lizmap, `mainconfig.ini.php`, `localconfig.ini.php`, and `liveconfig.ini.php` are merged in this order. So parameters into `liveconfig.ini.php` have higher priority that those into `localconfig.ini.php`, which in turn, have parameters having higher priority over parameters of `mainconfig.ini.php`.

Note: `liveconfig.ini.php` and `localconfig.ini.php` may not exists on old releases.

Other framework configuration files:

- `profiles.ini.php` contains all credentials to access to databases, smtp servers, ldap etc. You should modify it to set these access parameters.
- `installer.ini.php` contains informations about Lizmap modules and their state. Don't touch it, and don't erase it between upgrades. You can delete it only if you want to reinstall Lizmap.

Lizmap configuration files:

- `lizmapConfig.ini.php` contains configuration parameters specific to Lizmap.
- `lizmapLogConfig.ini.php` contains configuration for the Lizmap logger

For backups and upgrades, you should keep `localconfig.ini.php`, `liveconfig.ini.php`, `lizmapConfig.ini.php`, `profiles.ini.php`, `lizmapLogConfig.ini.php` and `installer.ini.php`. They are all modified during the life or the installation of the application.

5.4.2 Setting language

Lizmap detects automatically the language of the user (given by his browser), and it supports many languages.

Available language into Lizmap are: fr_FR, en_US, it_IT, es_ES, eu_ES, pt_PT, el_GR, de_DE, pl_PL, ru_RU, fi_FI, gl_ES, sv_SE, nl_NL, ro_RO, hu_HU.

If the browser of the user indicates an unsupported language code, the default language of Lizmap is used, and is en_US.

You can change the default language of Lizmap by setting the parameter `locale` Into localconfig.ini.php, with your prefered language code.

You can also limit available language by changing the `availableLocales` option.

In this example, only 3 languages are available and the default language is italian:

```
locale = it_IT
availableLocales = en_US,fr_FR,it_IT
```

There is also a `fallbackLocale` option but it is not recommended to change it.

5.5 LDAP authentication

The advantage of using LDAP is that all the users and groups information can be held on one server which is centrally managed.

The first thing to do is to install the php ldap extension on your linux system:

```
apt-get install php5.6-ldap
```

or

```
apt-get install php7.x-ldap
```

Then, in order to enable the LDAP support Lizmap, you have to change the authentication method in the configuration as follows.

5.5.1 Enabling LDAP

In the `localconfig.ini.php`, you have to set these parameters:

```
[modules]
ldapdao.access=1

[coordplugin_auth]
driver=ldapdao
```

These parameters enable the ldapdao module of Lizmap. If you copied localconfig.ini.php from localconfig.ini.php.dist, you have already this parameters but they are commented.

Then you have to add a section *[ldap:lizmapldap]* into your profiles.ini.php, with some settings that allow to connect to your ldap server, and to search users into the ldap. The section may already exist if you copied profiles.ini.php from the profiles.ini.php.dist.

```
[ldap:lizmapldap]
hostname=localhost
port=389
adminUserDn="cn=admin,ou=lizmap,dc=com"
adminPassword=""
searchUserBaseDN="dc=XY,dc=fr"
searchUserFilter="(&(objectClass=posixAccount)(uid=%%LOGIN%%))"
bindUserDN="uid=%?%,ou=users,dc=XY,dc=fr"
searchAttributes="uid:login,givenName:firstname,sn:lastname,mail:email,
↪organization:organization,street:street,postcode:postcode,city:city"
searchGroupFilter=
searchGroupProperty="cn"
searchGroupBaseDN=""
```

See details about how to set these parameters in a section below.

Note: these parameters could be in the ldapdao section of auth.coord.ini.php configuration files, but it is not recommended to modify these files, as when you do upgrade, you should do modification again in them.

To finish to enable the module, run *php lizmap/install/installer.php*

5.5.2 ldap settings

Configuration properties for user data

To verify password, or to register the user into Lizmap the first time he authenticate himself, the plugin needs some data about the user.

You should indicate to it which ldap attributes it can retrieve, and which database fields that will receive the ldap attributes values.

You indicate such informations into the *searchAttributes* property. It is a pair of names, `<ldap attribute>:<table field>`, separated by a comma.

In this example, `searchAttributes="uid:login,firstname,sn:lastname,mail:email,dn:"`:

- the value of the *uid* ldap attribute will be stored into the *login* field
- the value of the *sn* ldap attribute will be stored into the *lastname* field
- the value of the *firstname* ldap attribute will be stored into a field that have the same name, as there is no field name nor `:`.
- there will not be mapping for the *dn* property. There is a `:` without field name. It will be readed from ldap, and can be used into the *bindUserDN* DN template. (see below).

The list of possible fields in Lizmap are: *login*, *email*, *firstname*, *lastname*, *organization*, *phonenumber*, *street*, *postcode*, *city*, *country*. Only *login* and *email* are required. Others are optional.

Configuration properties for authentication

Before to try to authenticate the user against the ldap, the plugin retrieves user properties. It uses two configuration parameters : *searchUserFilter* and *searchAttributes*.

The *searchUserFilter* should contain the ldap query, and a `%%LOGIN%%` placeholder that will be replaced by the login given by the user.

Example: `searchUserFilter="(&(objectClass=posixAccount)(uid=%%LOGIN%%))"`

You may also indicate the base DN for the search, into *searchUserBaseDN*. Example: `searchUserBaseDN="ou=ADAM users,o=Microsoft,c=US"`.

Note that you can indicate several search filters, if you have complex ldap structure. Use `[]` to indicate an item list:

```
searchUserFilter[]="(&(objectClass=posixAccount)(uid=%%LOGIN%%))"
searchUserFilter[]="(&(objectClass=posixAccount)(cn=%%LOGIN%%))"
```

To verify the password, the plugin needs the DN (Distinguished Name) corresponding to the user. It builds the DN from a "template" indicated into the *bindUserDN* property, and from various data. These data can be the given login or one of the ldap attributes of the user.

- *Building the DN from the login given by the user*: bindUserDN should contain a DN, with a `%%LOGIN%%` placeholder that will be replaced by the login.

  Example: `bindUserDN="uid=%%LOGIN%%,ou=users,dc=XY,dc=fr"`. If the user give *john.smith* as a login, the authentication will be made with the DN `bindUserDN="uid=john.smith,ou=users,dc=XY,dc=fr"`.

  For some LDAP, the DN could be a simple string, for example an email. You could then set `bindUserDN="%%LOGIN%%@company.local"`. Or even `bindUserDN="%%LOGIN%%"` if the login can type the full value of the DN or an email or else.. (Probably it's not recommended to allow a user to type himself its full DN, it can be a security issue)

- *Building the DN from one of the ldap attributes of the user*. In this case, the plugin will first query the ldap directory with the *searchUserFilter* filter, to retrieve the user's ldap attributes. Then, in bindUserDN, you can indicate a DN where some values will be replaced by some attributes values, or you can indicate a single attribute name, corresponding to an attribute that contain the full DN of the user.

  For the first case, bindUserDn should contain a DN, with some `%?%` placeholders that will be replaced by corresponding attributes value. Example: `bindUserDN="uid=%?%,ou=users,dc=XY,dc=fr"`. Here it replaces the `%?%` by the value of the *uid* attribute readed from the user's attributes. The attribute name should be present into the *searchAttributes* configuration property, even with no field mapping. Ex: `...,uid:,...`. See above.

  For the second case, just indicate the attribute name, prefixed with a *$*. Example: `bindUserDN="$dn"`. Here it takes the *dn* attribute readed from the search, and use its full value as the DN to login against the ldap server. It is useful for some LDAP server like sometimes Active Directory that need a full DN specific for each user. The attribute name should be present into the *searchAttributes* configuration property, even with no field mapping. Ex: `...,dn:,...`. See above.

Note that you can indicate several dn templates, if you have complex ldap structure. Use `[]` to indicate an item list:

```
bindUserDN[]="uid=%?%,ou=users,dc=XY,dc=fr"
bindUserDN[]="cn=%?%,ou=users,dc=XY,dc=fr"
```

Configuration properties for user rights

If you have configured groups rights into Lizmap, and if these groups match your ldap groups, you can indicate to the plugin to automatically put the user into the application groups, according to the user ldap groups.

You should then indicate into *searchGroupFilter* the ldap query that will retrieve the groups of the user.

Example: `searchGroupFilter="(&(objectClass=posixGroup)(member=%%USERDN%%))"`

`%%USERDN%%` is replaced by the user dn. `%%LOGIN%%` is replaced by the login. You can also use any ldap attributes you indicate into *searchAttributes*, between *%%*. Example: `searchGroupFilter="(&(objectClass=posixGroup)(member=%%givenName%%))"`

Warning : setting *searchGroupFilter* will remove the user from any other application groups that don't match the ldap group. If you don't want a groups synchronization, leave *searchGroupFilter* empty.

With *searchGroupProperty*, you must indicate the ldap attribute that contains the group name. Ex: `searchGroupProperty="cn"`.

You may also indicate the base DN for the search, into *searchGroupBaseDN*. Example: `searchGroupBaseDN="ou=Groups,dc=Acme,dc=pt"`.

Debugging

If the authentication does not working, you can have more details on what is wrong. To see these details, you should activate the traces for ldapdao.

In your var/config/localconfig.ini.php, set these parameters

```
[logger]
auth=file

[fileLogger]
auth=auth.log
```

Then, in var/log/auth.log, you will have some messages from the ldap connector. Remove these settings when you don't need them, to avoid a huge auth.log file.

5.6 Advanced installation configuration

5.6.1 Lizmap behind a proxy/reverse proxy

Sometimes, some URL in Lizmap may not what you wanted. For example, there are using the "http" protocol instead of the "https" protocol. Or it may content an unwanted port (`http://mydomain:5468/` instead of `http://mydomain/`). Or url may not contain the real domain name.

In most of case, this is because the web server or PHP-fpm is behind a master web server (which act as a proxy or reverse proxy), and then environment parameters given by PHP or the backend web server to Lizmap are wrong.

You can tell to Lizmap to force to HTTPS, to use the right domain or to use the right port.

In `localconfig.ini.php`, you can use these following configuration parameters.

To force to use the port 80 (true) or the 8080 for example with the http protocol:

```
forceHTTPPort = true
# or
forceHTTPPort = 8080
```

To force to use the port 443 (true) or the 8443 for example with the https protocol:

```
forceHTTPSPort = true
# or
forceHTTPSPort = 8443
```

To specify the domain name of your Lizmap application, if Lizmap cannot guess it:

```
domainName = www.example.com
```

If the URL path of the backend web server does not correspond to the URL path of the frontal web server (ex: the proxy redirects urls like `http://example.com/index.php` to your web server `http://backend.example.com/foo/bar/index.php`, You have to indicate the "public" URL path (basePath) and the backend URL Path (backendBasePath):

```
[urlengine]
basePath= /
backendBasePath = /foo/bar
```

If the reverse proxy redirect HTTPS request to HTTP, you must deactivate the HTTPS check:

```
[urlengine]
checkHttpsOnParsing = off
```

Starting with Lizmap 3.0.18 and 3.1.6, you can indicate to force all generated URL with the HTTPS protocol, when Lizmap doesn't know what is the protocol used by requests on the reverse proxy:

```
[urlengine]
checkHttpsOnParsing = off
forceProxyProtocol = https
```

5.6.2 Using environment variables

If some credentials or parameters are available in environment variables, you can indicate use them into the `localconfig.ini.php` or `profiles.ini.php` files. Be sure the environment variables are available to the PHP-FPM process or the APACHE/NGINX process.

Into this files, use the syntax `${VARIABLE_NAME}`.

For example, to indicate postgresql credentials stored into these variables :

```
LIZMAP_PGSQL_HOST=localhost
LIZMAP_PGSQL_DATABASE=lizmap
LIZMAP_PGSQL_LOGIN=admin
LIZMAP_PGSQL_PASSWORD="Sup3Rp4ssw0rd!"
```

You write this configuration into profiles.ini.php:

```
[jdb:jauth]
driver="pgsql"
database=${LIZMAP_PGSQL_DATABASE}
host=${LIZMAP_PGSQL_HOST}
user=${LIZMAP_PGSQL_LOGIN}
password=${LIZMAP_PGSQL_PASSWORD}
```

5.7 Upgrade Lizmap Web Client

5.7.1 Upgrading from Lizmap 2.x

See documentation of Lizmap 3.1.

5.7.2 Upgrading from Lizmap 3.x versions

From 3.0 versions to upper, here is how to upgrade.

Data backup

Backup your data into a directory (ex: /tmp) with the lizmap/install/backup.sh script, so you could reinstall them if the installation failed.

```
lizmap/install/backup.sh /tmp
```

If you want to backup by hand, you should backup at least these files:

- var/db/jauth.db
- var/db/logs.db
- var/config/installer.ini.php
- var/config/liveConfig.ini.php (if it exists)
- var/config/lizmapConfig.ini.php
- var/config/localconfig.ini.php
- var/config/profiles.ini.php

Replace Lizmap files

Get the Lizmap archive (by downloading an archive or by doing a git clone/pull)

You should

- replace the lib/ directory by the new lib/ directory
- replace files into lizmap/ directory by the new lizmap/ files
- If the replacement has erased some files that you've been backuped, restore them with `lizmap/install/restore.sh /tmp`

Launch the installer

You have to launch the installer, it will upgrade some stuff: database tables, configuration etc..

```
sudo lizmap/install/clean_vartmp.sh
php lizmap/install/installer.php
sudo lizmap/install/clean_vartmp.sh
```

Note: if you upgrade from 3.0 or 3.1 to Lizmap 3.2/3.3, and if you are using the ldap authentication with the ldapdao module, you have to know that this module is included into Lizmap 3.2/3.3 and is pre-configured. So, before launching the installer, you have to remove the ldapdao module you've installed, and you have to configure the ldapdao module in a little different manner than when installing it by hand. See the ldap configuration section in this manual.

Delete Jelix temporary files

```
rm -rf /var/www/$MYAPP-$VERSION/temp/lizmap/*
```

Redefine the rights to the application files

```
cd /var/www/$MYAPP-$VERSION
chown :www-data temp/ lizmap/var/ lizmap/www lizmap/install/qgis/edition/ -R
chmod 775 temp/ lizmap/var/ lizmap/www lizmap/install/qgis/edition/ -R
```

Migrating from Sqlite to Postgresql

You may have installed Lizmap with Sqlite. You should then have these files `lizmap/var/db/jauth.db` and `lizmap/var/db/logs.db`, where some data like users, permissions and logs are stored. And you should have this configuration into `lizmap/var/config/profiles.ini.php`:

```
[jdb:jauth]
driver=sqlite3
database="var:db/jauth.db"

[jdb:lizlog]
driver=sqlite3
database="var:db/logs.db"
```

It you have a such configuration, you can migrate data to a Postgresql database.

First, create a Postgresql database, and then change the configuration into `lizmap/var/config/profiles.ini.php`, by setting access parameters to the Postgresql database. It is recommended to create a schema into the database, for example `lizmap` , if it contains already some tables.

For example :

```
[jdb:jauth]
driver=pgsql
host=localhost
port=5432
database="your_database"
user=my_login
password=my_password
search_path=lizmap,public

[jdb:lizlog]
driver=pgsql
host=localhost
port=5432
database="your_database"
user=my_login
password=my_password
search_path=lizmap,public
```

See the chapter about installation to know more about these parameters.

Then you can launch these scripts which will migrate the data.

```
php lizmap/scripts/script.php lizmap~database:migrateusers
php lizmap/scripts/script.php lizmap~database:migratelog
```

If there are no errors, you can then go onto lizmap with your browser, and check that you can authenticate yourself. You should see also the list of users into the administration panel. If this is the case, you can backup files jauth.db and logs.db and you can delete them.

If something goes wrong and you cannot fix the issue, revert the database access into `lizmap/var/config/profiles.ini.php` as before, like this :

```
[jdb:jauth]
driver=sqlite3
database="var:db/jauth.db"

[jdb:lizlog]
driver=sqlite3
database="var:db/logs.db"
```

And Lizmap should work well, but still with Sqlite.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search