

---

# **Georiviere**

***Release 0.9.8***

**Makina Corpus**

**Jan 20, 2022**



# USER MANUAL

<b>1</b>	<b>Common features</b>	<b>3</b>
1.1	Interface . . . . .	3
<b>2</b>	<b>Modules</b>	<b>5</b>
2.1	Streams . . . . .	5
2.2	Descriptions . . . . .	5
2.3	Knowledges . . . . .	6
2.4	Follow-ups . . . . .	6
2.5	Stations . . . . .	6
2.6	Studies . . . . .	6
2.7	Finance and administrative files . . . . .	6
2.8	Proceedings . . . . .	6
2.9	Interventions . . . . .	6
<b>3</b>	<b>Configuration</b>	<b>7</b>
<b>4</b>	<b>Import data</b>	<b>11</b>
4.1	Import altimetry file . . . . .	11
4.2	Import stations from Hub'Eau . . . . .	11
4.3	Import data references from Sandre . . . . .	12
4.4	Import zoning data from file . . . . .	12
<b>5</b>	<b>Install instructions</b>	<b>15</b>
5.1	Requirements . . . . .	15
5.2	Install . . . . .	15
5.3	Update . . . . .	16
<b>6</b>	<b>Basic settings</b>	<b>17</b>
6.1	Spatial reference identifier . . . . .	17
6.2	Default Structure . . . . .	17
6.3	Translations . . . . .	17
6.4	Spatial Extent . . . . .	17
<b>7</b>	<b>Advanced settings</b>	<b>19</b>
7.1	Georiviere settings . . . . .	19
7.2	Based on Geotrek or Mapentity settings . . . . .	19
<b>8</b>	<b>Install local Environment</b>	<b>21</b>
<b>9</b>	<b>Documentation</b>	<b>23</b>

<b>10 Publication</b>	<b>25</b>
10.1 CI . . . . .	25
10.2 Release . . . . .	25
10.3 Docker image . . . . .	25
<b>11 Authors</b>	<b>27</b>
11.1 Authors & trademark . . . . .	27
<b>12 Changelog</b>	<b>29</b>
12.1 CHANGELOG . . . . .	29
<b>13 Indices and tables</b>	<b>31</b>

Georiviere-admin is an application to handle aquatic environment (stream, river, watersheds...) in a given territory. It is a [Django](#) application, based on [Geotrek-admin](#) and [Mapentity](#).



## COMMON FEATURES

Since Georiviere is based on Geotrek-admin, it has the same features:

- display and edit contents with their geom
- export contents in ODT/DOC, PDF, GPX or shape files
- add linked files
- history

### 1.1 Interface

**Modules**

Nombre de résultats: Liste 12

Ajouter un nouvel objet: + Ajouter une station

Filtrer: Filtre

Navigation: Commune, DCE, Secteur, Sous DCE, Zone réglementaire, Bassin versant local

Zoomer sur une commune, un secteur ou un bassin versant

Code station	Libellé station	Profil de station
V101581001	La Semine à Châtillon-en-Mi...	Hydrométrie (HYDRO)
V101582401	La Valserine à Lancrans [Mo...	Hydrométrie (HYDRO)
V101582501	La Valserine à Lancrans [Le ...	Hydrométrie (HYDRO)
06067834	RU DE VAUCHENY A CONFO...	Qualité Physico-chimique (P...
06067838	RU DE VAUCHENY A CONFO...	Qualité Physico-chimique (P...
06067845	SEMINE A CHATILLON-EN-M...	Qualité Physico-chimique (P...
06067860	VALSERINE A BELLEGARDE...	Qualité Physico-chimique (P...
06067900	VALSARINE A CHATILLON-E...	Qualité Physico-chimique (P...
06068000	VALSERINE A BELLEGARDE...	Qualité Physico-chimique (P...
06580128	VALSERINE A BELLEGARDE...	Qualité Physico-chimique (P...
06580129	VALSERINE A CHATILLON-E...	Qualité Physico-chimique (P...
06580130	VALSERINE A MONTANGES	Qualité Physico-chimique (P...

Rechercher

Rechercher parmi les résultats

Parcourir les pages de résultats

Exporter la liste

Carte

To get more detail on how add and edit contents, have a look on [Geotrek-admin documentation](#) (in french).





## MODULES

Several modules are available in Georiviere, to display and edit professional content:

- Streams
- Descriptions (usage, land, morphology, status)
- Knowledges (vegetation, work or other knowledges)
- Follow-ups for a given knowledge
- Stations (hydrometric, temperature, physico-chemical quality or other types)
- Studies
- Finance and administrative files
- Proceedings
- Interventions (on knowledges)

In a content detail page, nearby other contents are displayed.

### 2.1 Streams

Streams is a line with a source location.

### 2.2 Descriptions

Four description models are available:

- usage
- land
- morphology
- status

Land and morphology have both a geom relative to a stream, and are created along to a stream on its creation. They can be cut to edit more precisely their attributes.

Usage and status are both standalone geometry and can be whatever point, line or polygon.

## 2.3 Knowledges

Knowledges can be a point, line or polygon about a stream, of vegetation, work or other type.

Vegetation and work type knowledges have specific fields.

## 2.4 Follow-ups

Follow-ups can be added to a knowledge, to take regular readings related to this knowledge.

## 2.5 Stations

Stations are points of measures, but they can be line or polygon too. Tracked parameters can be added to a station, with their name, measure and transmission frequency, etc.

Station of hydrometric, temperature, physico-chemical quality can be imported from Hub'Eau API.

## 2.6 Studies

A study is just a content with authors and year.

## 2.7 Finance and administrative files

Estimated or actual costs, fundings, or organisations involved in a project can be filled in an administrative file.

Every content in georiviere can be linked to an administrative file with operation, and for each you can edit its estimated, material, sub-contracting or man-days costs.

## 2.8 Proceedings

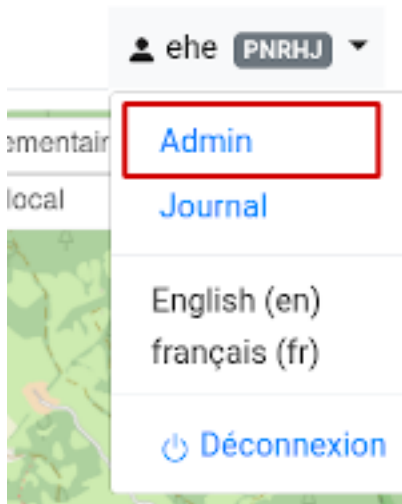
A proceeding can list all juridic events related to it.

## 2.9 Interventions

Intervention is a maintenance intervention related to a follow-up.

## CONFIGURATION

To customize lists for each module, go to django administration page.



- **Description**

- Bank states
- Facies diversities
- Flow types
- Granulometric diversities
- Habitat types
- Habitats diversities
- Land types
- Plan layout types
- Sediment dynamics
- Status types
- Usage types
- Working space types

- **Finances and administration**

- Admin file domains

- Admin file types
  - Administrative operations
  - Job categories
  - Organisms
- **Knowledge**
  - Age class diversities
  - Follow-up types
  - Knowledge types
  - **Vegetations:**
    - \* Specific diversities
    - \* Vegetation states
    - \* Vegetation stratas
    - \* Vegetation thickness types
    - \* Vegetation types
  - **Work:**
    - \* Work bank effects
    - \* Work fish continuity effects
    - \* Work materials
    - \* Work sediment effects
    - \* Work states
    - \* Work stream influences
    - \* Work types
- Main: File types
- **Maintenance**
  - Intervention's disorders
  - Intervention's stakes
  - Intervention's statuses
  - Intervention's types
  - Interventions
- **Observations**
  - Parameter categories
  - Parameters
  - Station profiles
  - Units
- Proceeding: Event types
- Studies Study types

- **Watershed**

- Watershed types
- Watersheds

- **Zoning**

- Cities
- Districts
- Restricted area types
- Restricted areas



## IMPORT DATA

To import data, you have to run these commands from the server where Georiviere-admin is hosted.

### 4.1 Import altimetry file

Put your altimetry file in `var/` folder, and run command

```
docker-compose run --rm web ./manage.py loaddem <dem_path>
```

where `<dem_path>` is `/opt/georiviere-admin/var/my_dem_file.tiff`

### 4.2 Import stations from Hub'Eau

Stations can be imported from french Hub'Eau APIs :

- Temperature stations with `import_temperature_stations`
- Hydrometry with `import_hydrometric_stations`
- Physico-chemical quality with `import_pcquality_stations`

Optional arguments:

```
--department DEPARTMENT [DEPARTMENT ...]
                        Department code
-p, --with-parameters  Get also parameter tracked by the station
--size SIZE            Results per page
```

Example:

```
docker-compose run --rm web ./manage.py import_pcquality_stations --department 39,25
```

## 4.3 Import data references from Sandre

Some data references can be imported from Sandre, for now only units are imported.

Usage:

```
docker-compose run --rm web ./manage.py import_reference_data
```

## 4.4 Import zoning data from file

Put your files into var/ folder as for altimetry profile import.

### 4.4.1 Load cities

Load Cities from a file within the spatial extent : `loadcities <file_path>`

Optional arguments::

```
--code-attribute CODE, -c CODE      Name of the code's attribute inside the file
--name-attribute NAME, -n NAME      Name of the name's attribute inside the file
--encoding ENCODING, -e ENCODING    File encoding, default utf-8
--srid SRID, -s SRID               File's SRID
--intersect, -i                   Check features intersect spatial extent and not only within
```

### 4.4.2 Load districts

Load Districts from a file within the spatial extent `loaddistricts <file_path>`

Optional arguments::

```
-h, --help                        show this help message and exit
--name-attribute NAME, -n NAME    Name of the name's attribute inside the file
--encoding ENCODING, -e ENCODING  File encoding, default utf-8
--srid SRID, -s SRID             File's SRID
--intersect, -i                  Check features intersect spatial extent and not only within
```

Load Restricted Area from a file within the spatial extent `loadrestrictedareas <file_path>`

Optional arguments::

```
-h, --help                        show this help message and exit
--name-attribute NAME, -n NAME    Name of the name's attribute inside the file
--encoding ENCODING, -e ENCODING  File encoding, default utf-8
```

(continues on next page)



(continued from previous page)

<code>--srid SRID, -s SRID</code>	File's <code>SRID</code>
<code>--intersect, -i</code>	Check features intersect spatial extent <b>and not</b> only within



## INSTALL INSTRUCTIONS

### 5.1 Requirements

- You need docker installed. Docker-compose is recommended in the configuration below.
- **Optional** : if you want to use external database, prepare a postgresql 10+ postgis2.5+ database with postgis and postgis\_raster enabled, and a dedicated user.

You can use external database by commenting postgres container and volume references in docker-compose.yml, and set variables

- POSTGRES\_HOST
- POSTGRES\_PORT
- POSTGRES\_USER
- POSTGRES\_PASSWORD
- POSTGRES\_DB
- You can use external nginx proxy. Edit provided nginx conf file and comment nginx references in docker-compose.yml. Fix web:8000 to 127.0.0.1:8000 in nginx.conf.

### 5.2 Install

- Download [zip package](#)
- Unzip it where you want

```
unzip install.zip  
cd georiviere
```

- Prepare environment variables

```
mv .env.dist .env
```

-> Set all required values

- Init default var folder

```
docker-compose run --rm web bash -c "exit"
```

- Set at least these variables in var/conf/custom.py:
  - SRID

- DEFAULT\_STRUCTURE\_NAME
- SPATIAL\_EXTENT

As geotrek overlay, these settings should be set BEFORE database initialization. See *Basic settings* for details

- Init database and project config

```
docker-compose run --rm web update.sh
```

- Create your super user

```
docker-compose run --rm web ./manage.py createsuperuser
```

- Load initial data

```
docker-compose run --rm web ./manage.py loaddata georiviere/**/fixtures/basic.json
```

- Launch stack

```
docker-compose up
```

## 5.3 Update

- Read [release notes](#) about bugfix, news and breaking changes.
- Backup your data (database and var folder)
- Pull latest image

```
docker-compose pull
```

- Run post update script

```
docker-compose run --rm web update.sh
```

- Relaunch you docker-compose stack

```
docker-compose down  
docker-compose up
```

## BASIC SETTINGS

Settings can be overridden in `var/conf/custom.py` file.

Basic settings should be defined on installation. See *Geotrek-admin documentation* <<https://geotrek.readthedocs.io/en/master/advanced-configuration.html#basic-settings>> for details.

### 6.1 Spatial reference identifier

```
SRID = 2154
```

Spatial reference identifier of your database. Default 2154 is RGF93 / Lambert-93 - France

### 6.2 Default Structure

```
DEFAULT_STRUCTURE_NAME = "GEOTEAM"
```

Name for your default structure.

### 6.3 Translations

```
MODELTRANSLATION_LANGUAGES = ('en', 'fr', 'it', 'es')
```

Languages of your project. It will be used to generate fields for translations. (ex: `description_fr`, `description_en`)

### 6.4 Spatial Extent

```
SPATIAL_EXTENT = (105000, 6150000, 1100000, 7150000)
```

Boundingbox of your project : x minimum , y minimum , x max, y max



## ADVANCED SETTINGS

More settings can be overridden in `var/conf/custom.py` file.

### 7.1 Georiviere settings

Base intersection margin

<code>BASE_INTERSECTION_MARGIN = 2000</code>
--

### 7.2 Based on Geotrek or Mapentity settings

Some settings come from Geotrek-admin or Mapentity, on which Georiviere is based:

- [Email settings](#)
- [Change or add WMTS tiles layers](#)
- [Map layers colors and style](#)

See [Geotrek-admin documentation](#) for further information.





## INSTALL LOCAL ENVIRONMENT

- Configuration

To get local environment working, we recommend to use a custom domain, as 'georiviere.local'. Define it in your /etc/hosts.

Copy the env dist file

```
cp .env.dist .env
```

Set required values, for postgres database access

- Init database:

```
docker-compose run --rm web ./manage.py migrate
```

- Create user:

```
docker-compose run --rm web ./manage.py createsuperuser
```

- Run:

```
docker-compose up
```

- Launch tests :

```
docker-compose run --rm web ./manage.py test
```

- With coverage :

```
docker-compose run --rm web coverage run ./manage.py test
docker-compose run --rm web coverage report -m
```

- Dependencies :

- Manage all project dependencies with pip-tools
- Use included pip-tools to generate requirements (python version should match georiviere version)

- Global dependencies :

- set global dependency in requirements.in

```
docker-compose run --rm web pip-compile
```

**pip-tools** does not upgrade any package by default. Package is upgrade only if new dependency require another version that already fixed in requirements.txt file.

To upgrade a package, run :

```
docker-compose run --rm web pip-compile --upgrade-package django==3.1.*
```

Development packages are separated in dev-requirements.in. dev-requirements.txt depends on requirements.txt. Each time you run pip-compile, dev-requirements.txt is updated. Each time you run pip-compile for requirements.txt, you should run pip-compile for dev-requirements.txt.

```
docker-compose run --rm web pip-compile dev-requirements.in
```

**Warning:** Geotrek is used as main library of this project. Sub-dependencies are not yet managed in geotrek setup.py. When you update geotrek, you should update requirements according geotrek dependencies versions.

## DOCUMENTATION

We use sphinx doc and sphinx-rtd-theme.

Requirements are included.

To compile and test documentation on local environment, run :

```
docker-compose run --workdir /opt/georiviere-admin/docs --rm web make html
```



## PUBLICATION

### 10.1 CI

- Each edition runs a CI build.
- All Georiviere-admin maintainers can review or merge Pull Requests.
- First time contributor not in maintainer team can request to be added. Maintainers can accept its first pull request to allow CI build.

### 10.2 Release

To release a new version :

- set version to georiviere/VERSION file.
- set changelog infos in docs/changelog.rst
- push or merge to master
- Go to <https://github.com/Georiviere/Georiviere-admin/releases>.
- Click on “Draft a new release”
- set new tag according older ones.
- Copy / paste changelog for version in release notes.
- In the end, CI publish publish new docker image to github packages.

### 10.3 Docker image

- Docker image is published after each release in Georiviere github repository: `ghcr.io/georiviere/georiviere-admin:latest`



## 11.1 Authors & trademark



- **Georiviere**, community software available under Open Source licence.
- Trademark is registered to **Institut National de la Propriété Intellectuelle** as a word mark under n°4691040.
- Brand policy

### 11.1.1 Makina Corpus

- Emmanuelle Helly
- Timothée de Montety
- Jean-Etienne Castagnede

### 11.1.2 Credits

#### Icons

- Stream : Creative Commons, Adrien Coquet, FR
- Work : construction sign work from [SVG Repo](<https://www.svgrepo.com/svg/307735/construction-sign-work>) - CC0
- Observation : observation from [SVG Repo](<https://www.svgrepo.com/svg/293710/observation>) - CC0
- Knowledge : idea from [SVG Repo](<https://www.svgrepo.com/svg/293713/idea>) - CC0
- Finance & administration : report from [SVG Repo](<https://www.svgrepo.com/svg/58321/report>) - CC0
- Studies : research from [SVG Repo](<https://www.svgrepo.com/svg/109479/research>)

- Description : Makina Corpus, derivated from [SVG Repo](<https://www.svgrepo.com/svg/258092/route-start>) - CC0
- Scissors from [SVG Repo](<https://www.svgrepo.com/svg/82088/opened-scissors>)



## CHANGELOG

### 12.1 CHANGELOG

#### 12.1.1 0.9.8 (2022-01-20)

##### Features

- Display distance from object to stream source

##### Enhancement

- Improve morpho display

##### Bug fixes

- Fix translations

#### 12.1.2 0.9.7 (2021-12-23)

##### Enhancement

- Change module order
- Add help text for multiselect

##### Bug fixes

- Fix logo header for PDF
- Fix man-days and costs display
- Fix translations

##### Dependencies

- Update to django-mapentity 7.0.5 and Geotrek 2.74.1

### **12.1.3 0.9.6 (2021-12-09)**

- Use mapentity standalone release
- Improve documentation
- Add source location on a stream
- Make cut topology simpler
- Add help message on how edit man-days cost
- Fix filters on intervention and follow-ups

### **12.1.4 0.9.5 (2021-11-08)**

- Improve documentation
- Improve README, maintainers and brand mark policy

### **12.1.5 0.9.4 (2021-11-05)**

- First code publication

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`