
Georiviere

Release 1.0.3

Makina Corpus

Dec 15, 2022

USER MANUAL

1	Common features	3
1.1	Interface	3
2	Modules	5
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
3	Configuration	7
4	Import data	11
4.1	Import data	11
4.2	Import altimetry file	12
4.3	Import stations from Hub'Eau	12
4.4	Import data references from Sandre	12
4.5	Import zoning data from file	13
5	Install instructions	15
5.1	Requirements	15
5.2	Install	15
5.3	Update	16
6	Basic settings	17
6.1	Spatial reference identifier	17
6.2	Default Structure	17
6.3	Translations	17
6.4	Spatial Extent	17
7	Advanced settings	19
7.1	Georiviere settings	19
7.2	Based on Geotrek or Mapentity settings	19
8	Install local Environment	21
8.1	Installation	21
8.2	Tests	21

8.3	Dependencies	22
9	Documentation	23
10	Publication	25
10.1	CI	25
10.2	Release	25
10.3	Docker image	25
11	Authors	27
11.1	Authors & trademark	27
12	Changelog	29
12.1	CHANGELOG	29
13	Indices and tables	33

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

The screenshot displays the Georiviere interface with a list of stations on the left and a map on the right. Red arrows highlight key features:

- Liste 14**: Number of results.
- + Ajouter une station**: Add new object.
- Filtrer**: Filter.
- Châtilion**: Navigation.
- DCE, Sous DCE, La Valserine**: Zoom on a commune, sector, or watershed.
- Rechercher**: Search.
- Rechercher parmi les résultats**: Search among results.
- Parcourir les pages de résultats**: Browse result pages.
- Exporter la liste**: Export list.

The list of stations includes:

Code station	Libellé station
V242056001	Le Tacon à Saint-Claude
06084800	RUISSEAU DU CHAPY A SEPTMONCEL
06084830	FLUMEN A ST-CLAUDE
06084850	RU DE LA SOURCE DES MOULINS A SEPTM...
06084865	TACON A ST-CLAUDE 1
06084870	TACON A ST-CLAUDE 2
06084901	GROSDAR A ST-CLAUDE
06084920	TACON A ST-CLAUDE 3
06085230	FLUMEN A VILLARD ST SAUVEUR
06085240	GROSDAR A ST CLAUDE 1
06085250	ABIME A ST CLAUDE
06300067	BIENNE A ST-CLAUDE 1
06400012	BIENNE A ST-CLAUDE 3
06400105	TACON A SAINT CLAUDE

The map shows a topographic view of the region with various locations marked, including Saint-Claude, Chaumont, and Villard-Saint-Sauveur. The bottom right corner indicates '14 résultats'.

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 and flow type. Length is computed from geometry.

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. For this, first value in type_connaissance = végétation and second value = ouvrage.

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

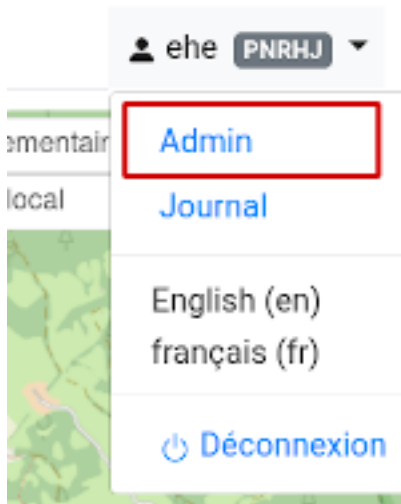
A proceeding can list all juridic events related to it.

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 data

Put your data file named CoursEau_FXX.shp in var/ folder, and run command

Custom your import file named import_bdtopage.py et put this file in georiviere/

```
#!/usr/bin/env python
import os
import django
import logging
from django.contrib.gis.gdal import DataSource

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'georiviere.settings')
django.setup()
from geotrek.authent.models import Structure
from django.contrib.gis.geos.collections import MultiLineString
from georiviere.river.models import Stream

# get first structure, adapt with correct structure
structure = Structure.objects.first()

ds = DataSource('var/CoursEau_FXX.shp')

for feat in ds[0][4:]:
    try:
        # get geos object
        geom = feat.geom.geos
        # force 2D
        geom.dim = 2
        name = feat.get('name') or 'No name'
        flow = feat.get('flow')
        if flow == 1:
            flow = 1
        elif flow == 2:
            flow = 2
        else:
            flow = 0
```

(continues on next page)

(continued from previous page)

```

        stream = Stream.objects.create(structure=structure,
                                        geom=feat.geom.
↪ geos,
                                        name=name,
                                        flow=flow)

    except Exception as exc:
        logging.warn(exc, feat.geom.geos.ewkt)

```

And run command : `docker compose run --rm web ./import_bdtotopage.py`

4.2 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`

4.3 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
```

4.4 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
```


4.5 Import zoning data from file

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

4.5.1 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
```

Example:

```
docker compose run --rm web ./manage.py loadcities /opt/georiviere-admin/var/commune.shp
↪--name-attribute nom --code-attribute insee_com
```

4.5.2 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
```

Example:

```
docker compose run --rm web ./manage.py loaddistricts /opt/georiviere-admin/var/epci.shp
↪--name-attribute nom --code-attribute code_siren
```

4.5.3 Load Restricted Area

Load Restricted Area from a file within the spatial extent `loadrestrictedareas <file_path>` and specify the name of the Area type

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
```

Example:

```
docker compose run --rm web ./manage.py loadrestrictedareas /opt/georiviere-admin/var/
↪ pnrhj.shp PNR --name-attribute nom
```

4.5.4 Import watershed

To import, use QGIS and edit `watershed_watershed` layer and specify “name” and “watershed_type_id” in attributes

INSTALL INSTRUCTIONS

5.1 Requirements

- **You need docker installed. Docker-compose is recommended in the configuration below.**
See [Docker](#) and [Docker compose](#) install documentations.
- **Optional** : if you want to use external database, prepare a postgresql 10+ postgres2.5+ database with postgres and postgres_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
- PGPORT
- POSTGRES_USER
- POSTGRES_PASSWORD
- POSTGRES_DB

Add local IPs in *pg_hba.conf* to allow connection from docker containers to your database.

- 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**

- Pull images

```
docker-compose pull
```

(Use ``docker compose`` **if** docker-compose version is `>=2.4.x`)

- 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 -d
```

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 -d
```

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.

After any change in `custom.py`, run:

```
docker compose restart
```

7.1 Georiviere settings

Base intersection margin

```
BASE_INTERSECTION_MARGIN = 2000
```

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

8.1 Installation

- 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
```

8.2 Tests

- 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
```

8.3 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.

```
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`

AUTHORS

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 1.0.3 (2022-12-15)

Enhancement

- Change secondary flow and habitat to multiselect field in description module
- Add fields to work : upstream and downstream bed impact, water impact
- Change vegetation strata field into a multiselect field

Bug fixes

- Display flow and source in stream detail
- Add flow filter in stream list
- Fix standalone intervention creation bug (#93)

Documentation

- Update doc installation
- Fix install doc with PostgreSQL not in Docker

12.1.2 1.0.2 (2022-05-22)

Bug fixes

- Fix ``./manage.py loadem ...`` command by including postgis libraries

12.1.3 1.0.1 (2022-03-30)

Enhancement

- Add data source and flow to stream

12.1.4 1.0.0 (2022-03-10)

Enhancement

- Add chosen multiselect on usage types
- Remove unused fields from Station form
- Get more data from Hubeau (start and end measure dates, measure type)
- Change base buffer width
- Change module picto colors
- Improve map color settings
- Display layers for all modules

Bug fixes

- Display missing unit
- Fix pip-tools / pip incompatibility

12.1.5 0.9.9 (2022-01-25)

Enhancement

- External link to station opened in new window
- Add unit on distance fields
- Remove secondary information from station detail
- Add chosen on some multiselect fields

Bug fixes

- Remove unwanted padding on lists
- Fix filter in service for stations
- Remove useless restricted area filter, replaced by zoning filter

Dependencies

- Update to django-mapentity 7.0.6 and Geotrek 2.75.0

12.1.6 0.9.8 (2022-01-20)

Features

- Display distance from object to stream source

Enhancement

- Improve morpho display

Bug fixes

- Fix translations

12.1.7 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

12.1.8 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

12.1.9 0.9.5 (2021-11-08)

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

12.1.10 0.9.4 (2021-11-05)

- First code publication

INDICES AND TABLES

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