



Documentation TerriSTORY®
docs.terristory.fr

Guide d'installation de TerriSTORY

Guide d'installation de TerriSTORY

Guide d'installation de TerriSTORY pour une machine Linux, ici avec l'OS Ubuntu 22.04.2. Ce guide décrit l'installation complète d'un environnement de développement local.

Certaines informations de *troubleshooting* sont disponibles en [fin de page ici](#).

Téléchargement du projet TerriSTORY

Le projet TerriSTORY® est partagé en ligne sur un [dépôt GitLab](#). La première étape est de télécharger ce projet en local.

Si git n'est pas installé, l'installer :

```
sudo apt install git
```

Cloner le dépôt Gitlab du projet vers un dossier local (par exemple dans `/home/<user>`) :

```
git clone https://gitlab.com/terristory/terristory.git
```

Installation de la base de données

Le projet utilise une base de donnée PostgreSQL avec PostGIS pour les données géographiques.

Installer PostgreSQL et PostGIS :

```
sudo apt install postgresql-12-postgis-3  
sudo apt install postgis
```

Par défaut l'utilisateur postgres n'a pas de mot de passe. Dans l'invite de commande Postgres (accessible avec la commande `psql`), ajouter un mot de passe :

```
ALTER USER postgres PASSWORD 'myPassword';
```

Création de la base de données depuis une archive

Si vous avez à disposition une archive d'une base de données existante obtenue via la commande `pg_dump`, basculer sur l'utilisateur `postgres` et créer la nouvelle base de donnée " `api` " à partir de cette archive :

```
sudo -i -u postgres  
pg_restore -U postgres -C -d api < [path to archive]
```

Création de la base de données depuis les fixtures

Si jamais vous n'avez pas accès à une archive d'une base de données, vous pouvez utiliser les *fixtures* prévues dans le projet pour créer une base de données partielle remplie de données aléatoires.

La création de cette base de donnée suppose l'existence d'un utilisateur "terristory". Pour créer cet utilisateur :

```
sudo -u postgres createuser -e terristory
```

Se placer ensuite dans le dossier `terristory/terriapi/fixtures` et exécuter les commandes suivantes pour créer une base de données *fixture* et sa structure :

```
psql -h localhost -U postgres < init.sql
psql -d fixture -h localhost -U postgres < public.sql
psql -d fixture -h localhost -U postgres < meta.sql
psql -d fixture -h localhost -U postgres < table_passage.sql
psql -d fixture -h localhost -U postgres < consultations.sql
psql -d fixture -h localhost -U postgres < strategie_territoire.sql
psql -d fixture -h localhost -U postgres < auvergne_rhone_alpes_station_mesure.sql
psql -d fixture -h localhost -U postgres < auvergne_rhone_alpes.sql
psql -d fixture -h localhost -U postgres < auvergne_rhone_alpes_poi.sql
psql -d fixture -h localhost -U postgres < static_fixtures.sql
psql -d fixture -h localhost -U postgres < boundaries_cities.sql
```

Un script Python `import_fixtures.py` permet de peupler cette base de données avec des données aléatoires. L'exécution de ce script doit se faire depuis l'environnement virtuel Python de terriSTORY créé dans la partie "[Installation de l'API](#)". Se placer dans `terristory/terriapi/fixtures` avec l'environnement activé et lancer le script :

```
python import_fixtures.py
```

Installation de Postile

Point d'attention

L'image Docker `oslandia/postile` n'est plus maintenue. Une version maintenue à jour pour TerriStory est accessible [ici](#) et son utilisation (décrite sur le dépôt) est préférable à la version explicitée dans cette section.

Docker est utilisé pour **Postile**, l'application développée par Oslandia qui affiche les tuiles OSM et les géométries des territoires.

Pour installer Docker, suivre les instructions d'installation de la documentation [ici](#).

Optionnellement compléter l'installation en ajoutant l'utilisateur Linux au groupe docker en suivant les instructions [ici](#).

Télécharger l'image Postile et lancer le container avec la commande suivante en adaptant le nom et les identifiants de la base de données :

```
docker run --net host oslandia/postile postile --pguser postgres --pgpassword myPassword --pgdatabase api --pghost localhost --pgport 5432 --listen-port 8081 --cors
```

Installation de l'API

Installation de l'environnement Python

L'API de Terristory utilise **Python 3.9** qui n'est pas forcément la version de base du système d'exploitation. Si nécessaire, il faut donc installer cette version de Python :

```
sudo apt install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt install python3.9
sudo apt install python3.9-venv
```

Créer un environnement virtuel Python 3.9 pour TerriSTORY, par exemple dans `/home/<user>` :

```
python3.9 -m venv env_terristory
```

Activer l'environnement :

```
. ~/env_terristory/bin/activate
```

Avec l'environnement activé, installer l'ensemble des dépendances Python pour le projet (avec le paramètre `.[dev]` pour installer également les dépendances spécifiques à l'environnement de développement) :

```
cd ~/terristory/terriapi
pip install -e .[dev]
```

Configuration de l'API

Un fichier de configuration `terriapi.ini.sample` dans `terristory/terriapi/terriapi` est à recopier en `terriapi.ini` et à compléter. Ce fichier contient tous les paramètres nécessaires pour que l'API et ses scripts fonctionnent (accès à la base de donnée, envoi de mail...).

Dans ce fichier `terriapi.ini`, configurer les informations relatives à la base de données créée précédemment :

```
pg_name = api
pg_user = postgres
pg_password = myPassword
```

Renseigner des chemins pour les différents dossiers et les créer si nécessaire. Par exemple :

```
upload=/home/<user>/terriStory/uploads/
upload_other_files=/home/<user>/uploads/files/
changelog_path=/home/<user>/terristory/
pcaet_path=/home/<user>/terristory/terriapi/terriapi/controller/temp_pcaet/
chemin_telechargement_donnees_shp=/home/<user>/terristory/terriapi/terriapi/data/donnees_ign/
upload_methodo=/home/<user>/terristory/front/public/pdf/
upload_icones_svg=/home/<user>/terristory/front/public/svg/
img_logo_source_fiches=/home/<user>/terristory/front/public/img/logo_source_fiches/
img_pop_up_accueil_path=/home/<user>/terristory/front/public/img/img_carousel/
```

Les icônes `.svg` et les pdfs méthodologiques doivent être placés dans les dossiers définis par `upload_icones_svg` et `upload_methodo`. Pour ne pas interférer avec le rechargement automatique du front en mode développement, il

est préférable de stocker ces documents dans un autre dossier et de les ajouter dans les dossiers `upload_icone_svg` et `upload_methodo` via des liens symboliques :

```
ln -s [Source_File_Path] [Symbolic_Link_Path]
```

De la sorte, les icônes et PDFs restent accessibles depuis la version locale (car présents dans l'architecture du front via les liens symboliques) sans conduire à des rechargements intempestifs de l'application lorsqu'un fichier est modifié (par exemple, lors de la création d'équipements). Cela est dû à la configuration suivante dans `webpack.config.dev.js` : `followSymlinks: false`.

Vérification de l'installation du backend

Vérifier l'installation en lançant l'API :

```
terriapi-serve --cors
```

Pour pouvoir utiliser le système de rechargement automatique pour le développement il faut être en mode debug (`debug = true` dans `terriapi.ini`) et lancer l'API avec :

```
terriapi-serve --cors --reload
```

L'ensemble des autres commandes liées à l'API (chargement de données, créations d'utilisateurs, tests...) est documentée dans le [README](#) de l'API.

Installation du front

Installer nodejs et le gestionnaire de paquets npm:

```
sudo apt install nodejs npm
```

En se plaçant dans `terristory/front`, installer l'ensemble des dépendances pour le front :

```
npm install --force
```

Configuration

Faire une copie du fichier de configuration locale `terristory/front/src/settings-local.js.sample` et le renommer `settings-local.js`. Ce fichier définit une configuration pour le développement local et offre des paramètres qui facilitent le développement. Le paramètre `HIDE_SPLASH_SCREEN` par exemple permet de ne pas afficher la fenêtre d'accueil à chaque chargement. Le paramètre `DEFAULT_REGION` permet de définir une région par défaut pour ne pas avoir à resélectionner une région à chaque fois.

Vérification de l'installation du front

Pour vérifier l'installation, se placer dans `terristory/front` et lancer le front avec :

```
npm start
```

TerriSTORY est alors accessible depuis le navigateur à l'adresse <http://localhost:3000/>.

Ressources complémentaires

L'environnement de développement local est désormais installé. Il est conseillé de parcourir le reste de la documentation pour en apprendre plus sur l'application TerriSTORY et son fonctionnement. Notamment, la [notice de contribution](#) donne des informations sur les bonnes pratiques et les conventions à suivre pour contribuer au développement de l'application, et la page "[Tester votre version de l'application](#)" explique comment configurer et lancer les tests de l'API et du front.

Autres paquets potentiellement nécessaires

Selon votre configuration, vous aurez besoin d'installer d'autres paquets utiles :

- lorsque `pkg-config` manque : `sudo apt install pkg-config`
- pour `node-canvas` : `sudo apt install libpixmap-1-dev libcairo2-dev libpango1.0-dev libjpeg8-dev libgif-dev`
- pour `psycopg2` : `sudo apt install libpq-dev python3.9-dev`