



Documentation TerriSTORY®
docs.terristory.fr

Tester votre version de l'application

Tester votre version de l'application

Tester le backend de TerriSTORY®

L'ensemble des tests est stocké dans le dossier `terriapi/tests/`. Pour les exécuter, une fois dans l'environnement virtuel dans lequel sont installées les dépendances, il suffit de se placer dans le dossier `/terriapi/` et d'exécuter la commande suivante :

```
pytest
```

Cela devrait exécuter tous les tests du backend et afficher les tests qui échoueraient sur votre version actuelle.

Tester le frontend de TerriSTORY®

La configuration de babel utilisée pour les tests est différente de celle pour le lancement du serveur local ou la compilation du code final. Il faut donc copier un fichier de configuration pour qu'il soit lu en priorité sur la config disponible dans le `package.json` (qui n'est donc utilisée que pour les status *development* - correspondant à la tâche `npm start` - et *production* - correspondant à la tâche `npm run build`).

Cela revient, en local, à exécuter les commandes suivantes depuis le dossier `front/` :

```
cp .babel.test-config.js babel.config.js
npm test -- --coverage
rm babel.config.js
```

Il est possible de créer un alias dans son `.bashrc` pour facilement exécuter les tests.

```
runtests() {
  cd ~/Bureau/TerriStory/front
  cp .babel.test-config.js babel.config.js
  npm test -- --coverage
  rm babel.config.js
}
```

Il ne reste alors plus qu'à exécuter `runtests` depuis n'importe où dans la console.

Les tests sont stockés dans le dossier `front/src/tests/`.

Tests fonctionnels ou transverses

Deux catégories de tests sont proposés ici :

- les récupérations de données d'API pour comparaison avant et après MAJ
- les captures d'écran automatiques pour comparaison avant et après MAJ

Comparaisons des résultats de l'API

Une étape intégrée au procédé de déploiement est celle qui teste les résultats obtenus par l'API avant et après une mise à jour. Cela permet de s'assurer que les modifications apportées entre deux versions ne changent pas ou ne brisent pas un point d'entrée, et qu'elles ne changent pas non plus le calcul des indicateurs (sauf modification explicite).

Pour voir un exemple, on pourra se référer au procédé de déploiement de TerriSTORY® qui intègre déjà ces outils. Par exemple, dans le fichier `/.gitlab-ci.yml` avec les étapes **test_data_api_dev** et **check_data_api_dev**.

Le principe général est d'exécuter, avant la modification, dans le dossier `terriapi/tests_fonctionnels/`, la commande suivante :

```
python save_api_outputs.py --state before
```

Puis, après la modification, les commandes :

```
python save_api_outputs.py --state after
python compare_api_outputs.py
```

Captures d'écran pour contrôle visuel des résultats

L'idée cette fois est, à partir d'une liste d'URLs et d'actions associées, de prendre automatiquement, grâce à la librairie [selenium](#), des captures d'écran de différentes pages de TerriSTORY® afin de s'assurer que les visuels sont bien restés stables et qu'il n'y a pas de régression.

Pour ce faire, il faut se placer dans le dossier `terriapi/tests_fonctionnels/` et exécuter les commandes suivantes, en se plaçant sur une machine avec une interface graphique et dans un environnement virtuel avec les librairies de développement installées en Python (selenium notamment) :

- Avant la MAJ (en ne touchant pas à la fenêtre firefox qui s'ouvre automatiquement) :

```
python screenshots.py --state before
```

- Après la MAJ :

```
python screenshots.py --state after
python compare_screenshots.py
```

Cela va créer un fichier **comparaison.html** dans le dossier `output/` avec une visualisation des captures d'écran avant et après la MAJ.

Les urls utilisées sont configurables dans le fichier `urls.py` dans ce même dossier.