



Documentation TerriSTORY®
docs.terristory.fr

Structure de code

Structure de code

La structure de l'application TerriSTORY®

L'application TerriSTORY® est organisée suivant l'architecture **Modèle-Vue-Contrôleur** qui sépare le modèle de données, l'interface utilisateur et la logique de contrôle.

Qu'est-ce qu'une architecture Modèle-Vue-Contrôleur ?

Modèle-vue-contrôleur (MVC) est un modèle de conception permettant de relier l'interface utilisateur aux modèles de données. Il consiste à distinguer trois entités : le modèle, la vue et le contrôleur.

Rôle de modèle

Le modèle assure la gestion des données et garantit leur intégrité. Il offre des méthodes pour récupérer les données et les mettre à jour (insertion, suppression, modification, etc.).

Rôle de vue

La vue fait l'interface avec l'utilisateur, elle assure l'affichage des données qu'elle récupère auprès du contrôleur et reçoit toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, ...). Ses différents événements sont envoyés au contrôleur.

Rôle de contrôleur

Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Si certains événements de l'utilisateur ne concernent pas les données, le contrôleur demande à la vue de se modifier directement.

Interactions entre le modèle, la vue et le contrôleur

Le schéma suivant modélise les différentes interactions entre le modèle, la vue et le contrôleur.

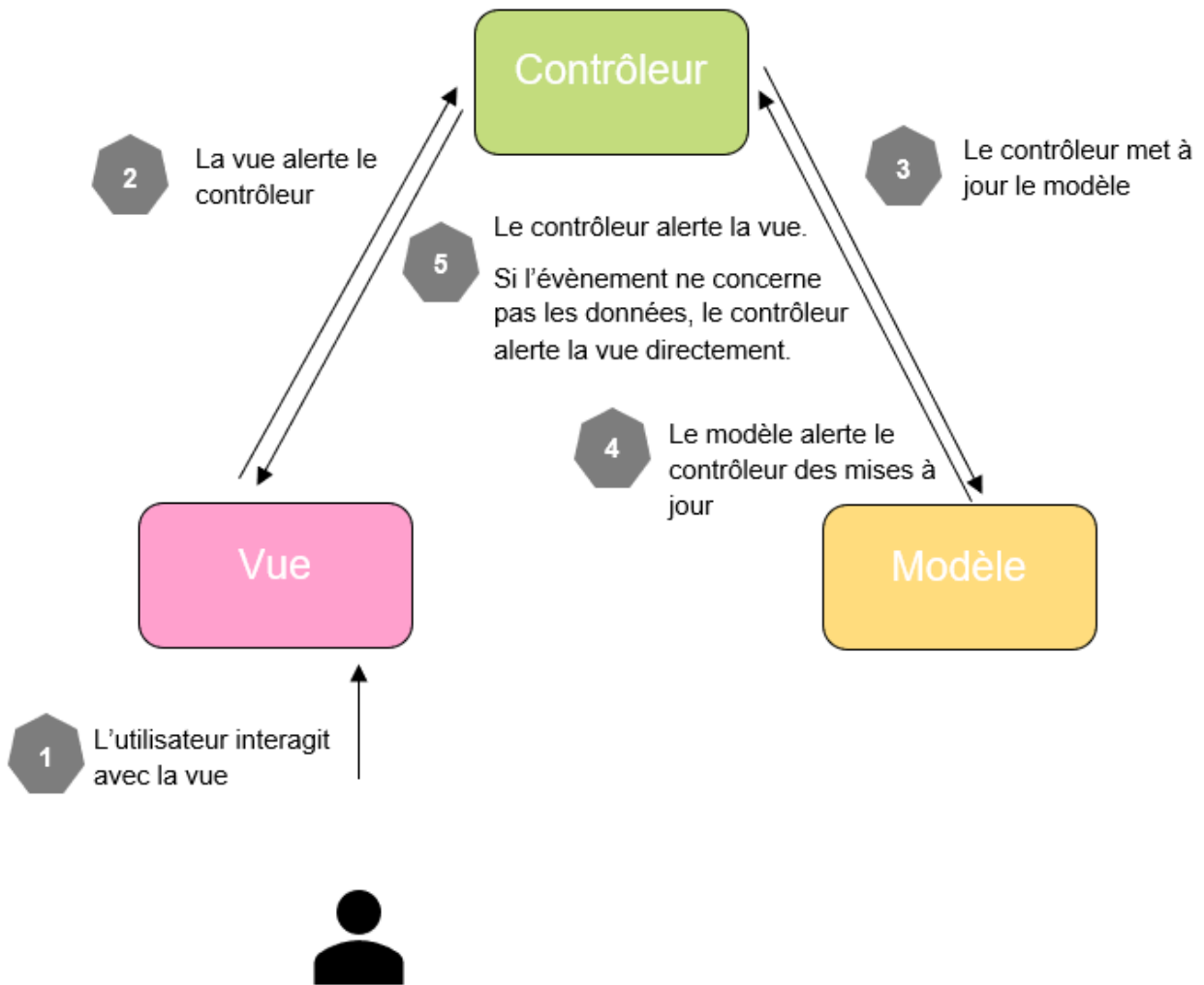
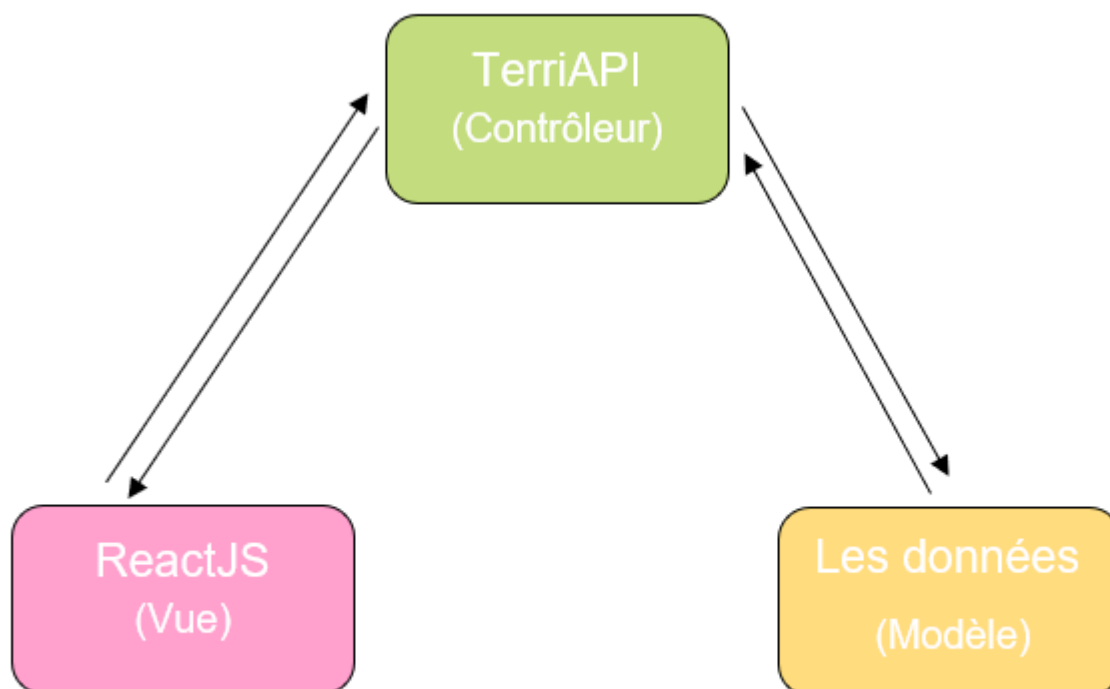


Schéma MVC

L'architecture MVC et l'application TerriSTORY®

Le schéma suivant modélise l'architecture MVC dans l'application TerriSTORY®



Modèle MVC TerriSTORY

Structure du code source de l'application TerriSTORY®

L'architecture de code source

L'application se compose de deux répertoires principaux :

front : représente la vue, il contient tous les fichiers Javascript qui permettent de faire l'interface utilisateur.

terriapi : représente le contrôleur et le lien avec la base de données de l'application, il contient tous les fichiers Python qui permet de manipuler les données.

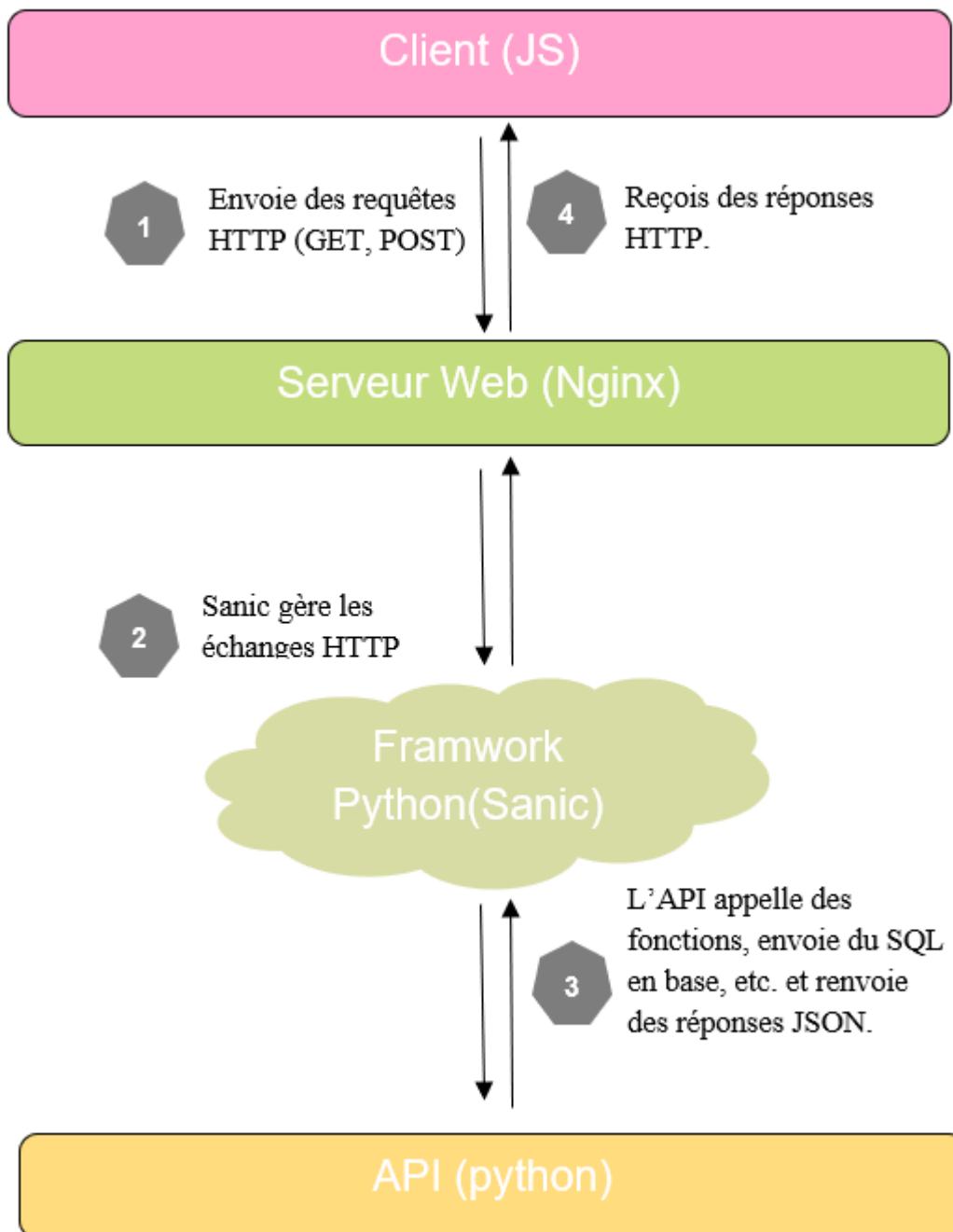
Le Javascript, côté front réalise des requêtes http. Il appelle l'**API** en utilisant **AJAX** en **POST** ou **GET** avec des **paramètres** que l'on retrouver dans la variable **Request**.

Les fonctions python sont accessible via des routes (URLs) définis dans le fichier **front/src/settings.js** si elles contiennent la ligne `\@bp.route` ou `\@pp.route` au-dessus de sa définition.

Exemple des fichiers qui contient des fonctions accessibles via des routes :

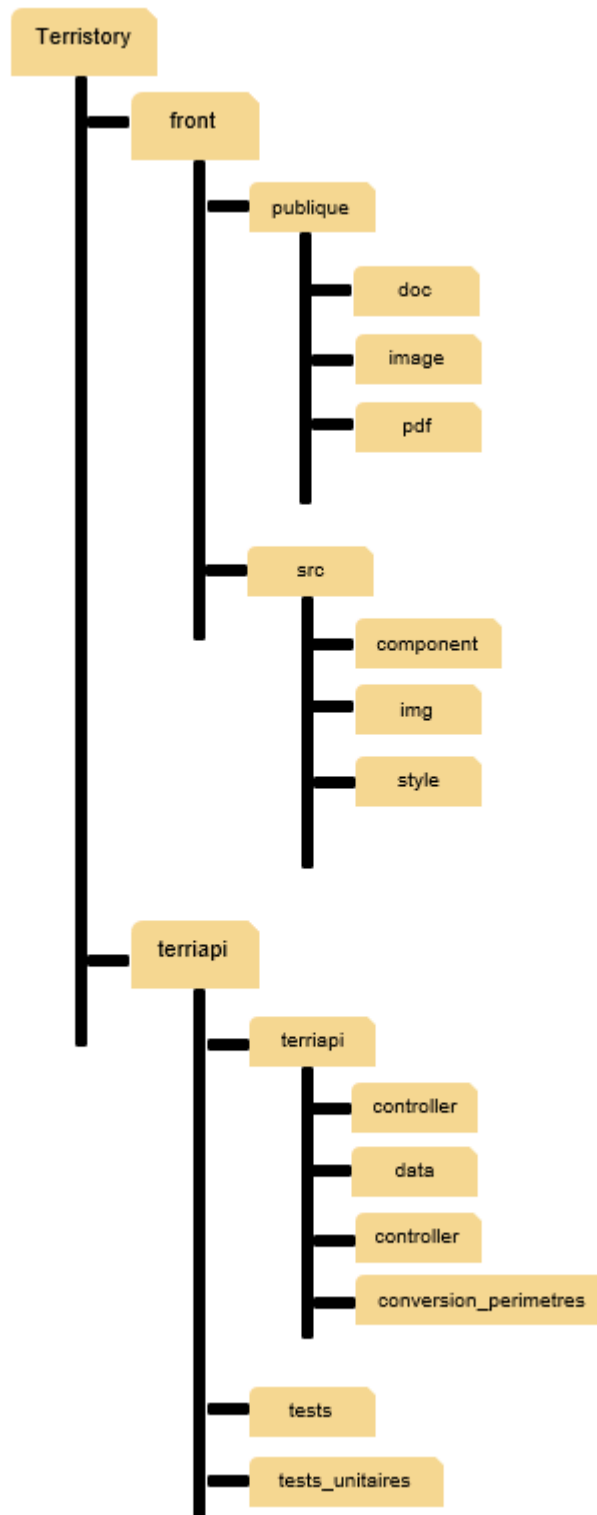
- **terriapi/terriapi/api.py**
- **terriapi/terriapi/user.py**

Le schéma suivant illustre les échanges HTTP entre le client et le serveur.



Les échanges HTTP entre le client et le serveur

Le schéma suivant illustre les répertoires principaux de l'application TerriSTORY®

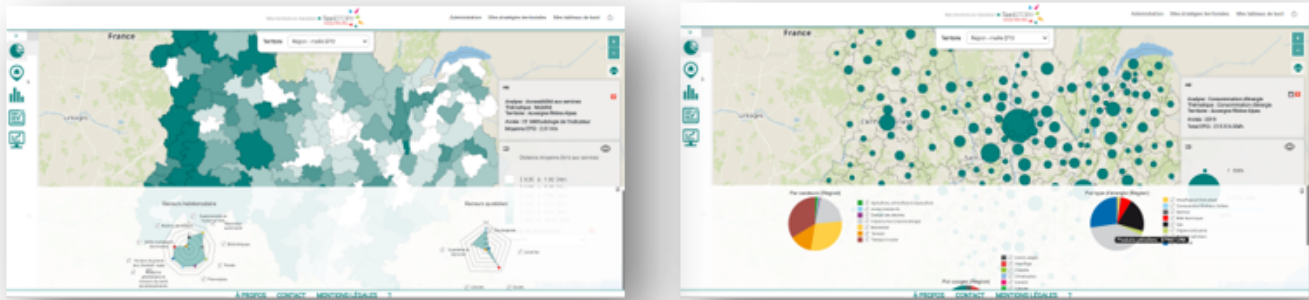


Arborescence de l'application

Les composantes de l'application

Le module de visualisation des indicateurs

Ce module permet de visualiser des indicateurs représentés de manière cartographique par des aplats de couleurs ou cercles proportionnels selon leur définition, et de les accompagner de diagrammes circulaires ou radar.



Module d'Indicateurs

Cette fonctionnalité repose sur une fonction principale dont le rôle est d'associer chacune des mailles territoriales à une valeur et de calculer le total, la moyenne ou la moyenne pondérée selon le type de l'indicateur.

Celle-ci fait appel à une architecture de classe dont la responsabilité est de générer les requêtes SQL nécessaires à l'obtention et la structuration des données par territoires et catégories, parfois filtrées selon des modalités ou des valeurs.

Les fichiers liés à ce composant sont :

- Pour la récupération de données coté serveur (back) : **terriapi/terriapi/api.py**
terriapi/terriapi/controller/ analyse.py
terriapi/terriapi/controller/ donnees_pour_representations.py
- Pour la restitution de données coté client (front) : **front/src/components/OIMap.js**
front/src/components/Filters.js

Le schéma suivant modélise l'architecture classe des différents types d'indicateurs.

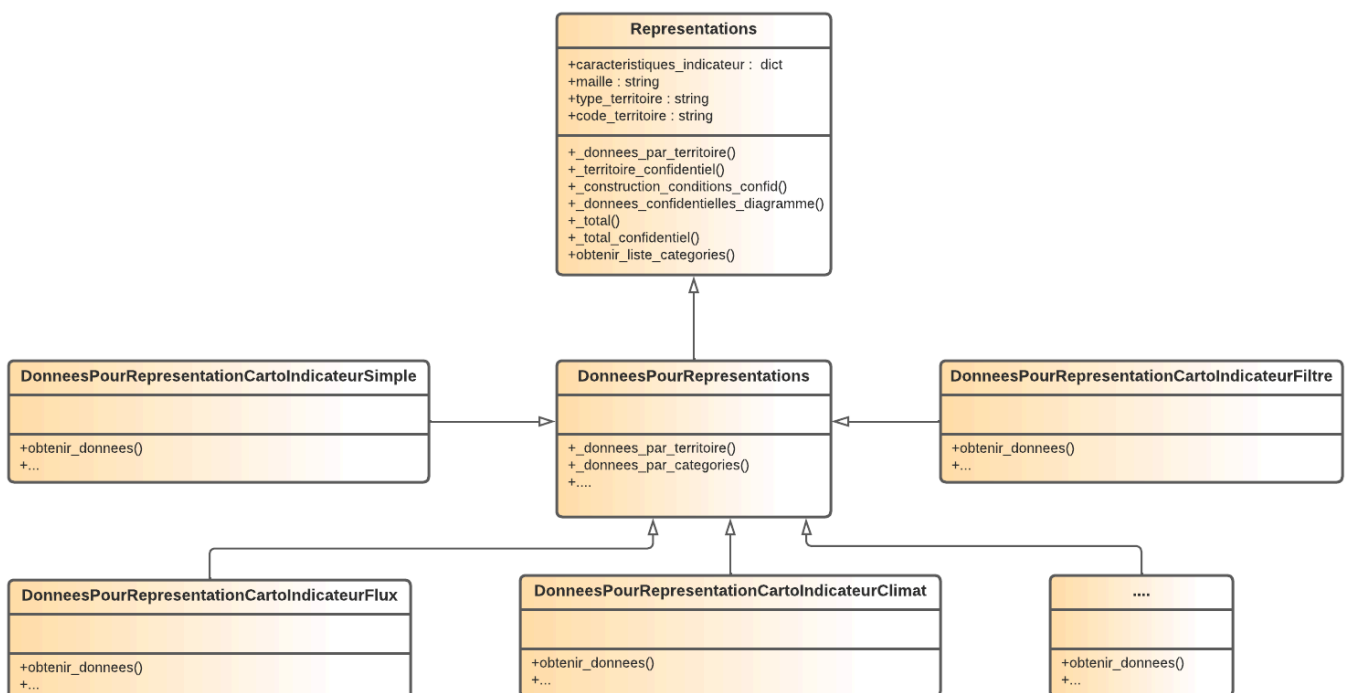
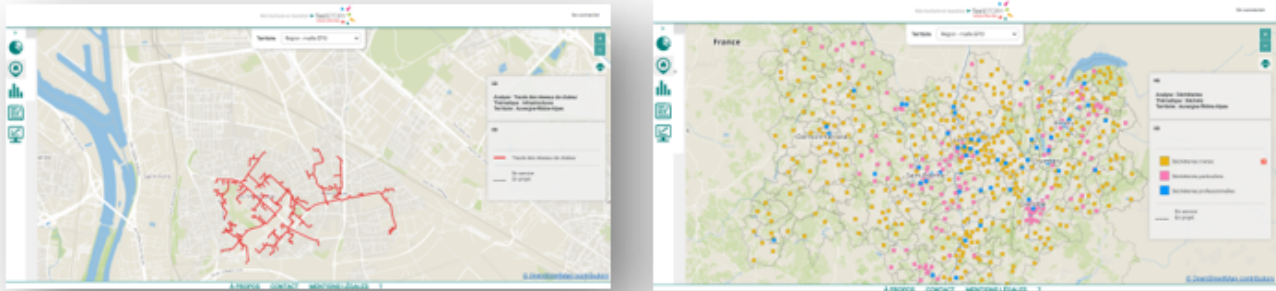


Diagramme de classe des indicateurs

Le module de visualisation des équipements

Ce module permet de visualiser des équipements représentés de manière cartographique par des lignes ou des points.

*Module d'Equipements*

Ce composant repose sur une fonction simple qui permet de récupérer un jeu de données qui associe pour chaque équipement, des points avec des coordonnées (X, Y) à une liste de données qu'on souhaite afficher sur l'interface cartographique.

Les fichiers liés à ce composant sont :

- Pour la récupération de données coté serveur (back) : **terriapi/terriapi/api.py**
terriapi/terriapi/controller/poi.py
- Pour la restitution de données coté client (front) : **front/src/components/OIMap.js**

front/src/components/Filters.js

La pop-up d'accueil

Le pop-up s'affiche au lancement de site régional de **terrstory.fr**, il permet d'afficher un message d'accueil personnalisable depuis les interfaces d'administrations.



Bienvenue sur le site de TerriSTORY®, outil d'aide au pilotage de la transition des territoires en Auvergne-Rhône-Alpes, proposé par l'Agence régionale Auvergne-Rhône-Alpes Énergie-Environnement.

Nouveautés

Évolution du module Stratégie territoriale :

- Intégration des objectifs PCAET dans la partie saisie de trajectoires cibles
- Ajout d'actions de mobilité durable, d'actions efficacité énergétique dans l'industrie et l'agriculture

Ajout de nouveaux indicateurs relatifs à la mobilité :

• Accessibilité à l'emploi
Veuillez sélectionner une maille d'analyse et un territoire dans les listes déroulantes situées en haut de la page.

J'ai compris

Pop-up d'accueil

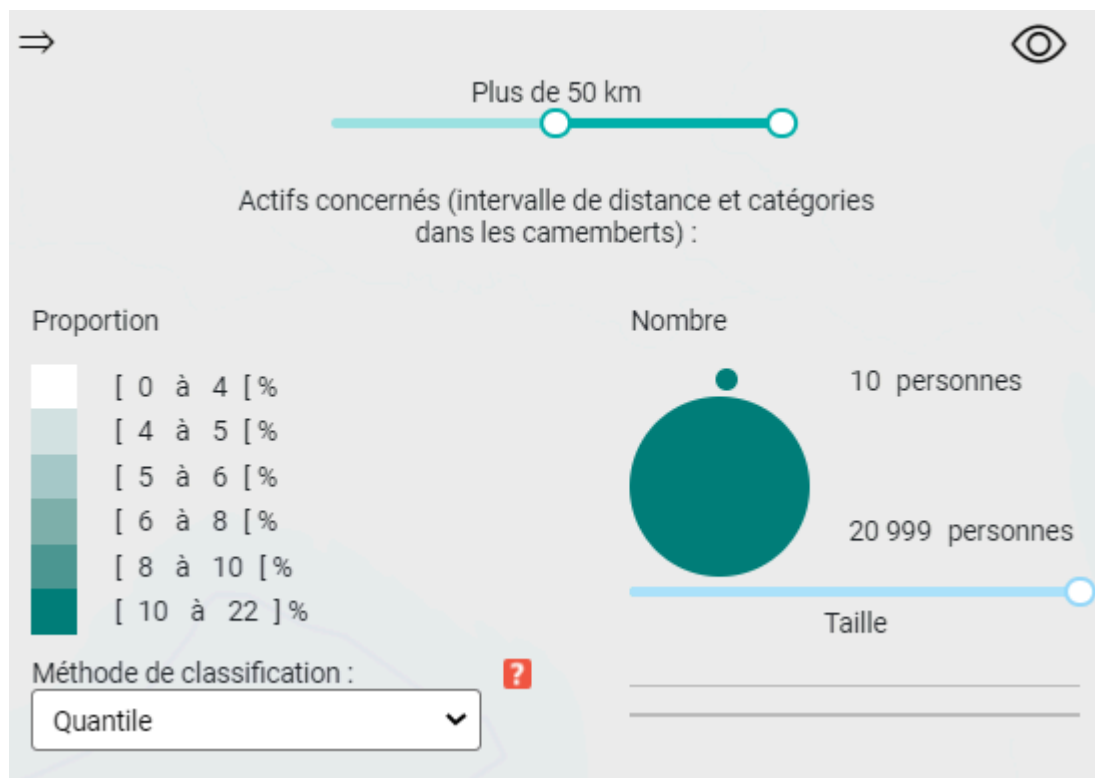
Les fichiers liés à ce composant sont :

- Fichier back : **terriapi/terriapi/controller/pop_up_accueil.py**
terriapi/terriapi/api.py
- Fichier front principal : **front/src/components/SplashScreen.js**
- Fichiers front liée au carrousel : **front/src/components/Carrousel.js**
front/src/components/PopUpEditor.js
front/src/components/ EditeurMarkdown.js
front/src/components/GestionnaireImagesCarrousel.js

La légende

Situé dans l'interface cartographique, ce composant permet d'afficher un certain nombre d'informations parmi lesquelles :

- Une légende accompagnée d'un formulaire de sélection de méthode de classification des données dans le cas où les données cartographiques sont représentées sous forme d'aplats de couleurs ;
- D'une entrée spécifique pour les données confidentielles ;
- D'une jauge qui permet de définir un seuil en dessous duquel on filtre les flux migratoires ;



Légende

Les fichiers liés à ce composant sont :

Fichier côté client (Front) : **front/src/components/Legend.js**

front/src/components/Style.js

La légende d'information

Situé dans l'interface cartographique, ce composant permet d'afficher un certain nombre d'informations parmi lesquelles :

- Afficher le nom de territoire sélectionné
- Afficher le nom et la thématique de l'indicateur ou l'équipement sélectionné
- Afficher la dernière année de disponibilité des données.
- Le total territoire
- Consulter la fiche méthodologique
- Exporter les données



Légende Informations

Le fichier lié à ce composant est : **front/src/components/Infos.js**

Le module d'analyses territoriales

Synthèse des indicateurs territoriaux

Ce module a été développé dans le cadre du projet européen **CESBA_Alps**, il permet d'évaluer la performance du territoire sur de nombreux sujets par rapport à d'autres territoires du même type (consommation et production d'énergie, émissions de gaz à effets de serre etc.).

Des indicateurs de performance sont calculés au moment de la procédure d'intégration des données de l'application et leur valeur figure donc en base de données, puis sont obtenues au moyen d'une requête simple écrite dans un module (un fichier Python) de la partie « controller ».

Cette partie de l'application est un peu à part, car elle est la seule qui nécessite d'intégrer des données au préalable en base de données pour la faire fonctionner.

Les fichiers liés à ce composant sont :

- Fichier côté serveur (Back) : **terriapi/terriapi/controller/territorialsynthesis.py**
- Fichier côté client (Front) : **front/src/components/TerritorialSynthesis.js**

Le module de suivi de la trajectoire énergétique

Ce module permet de suivre :

- L'évolution de la consommation énergétique par secteur, par énergie, et par usage.
- L'évolution de la production énergétique

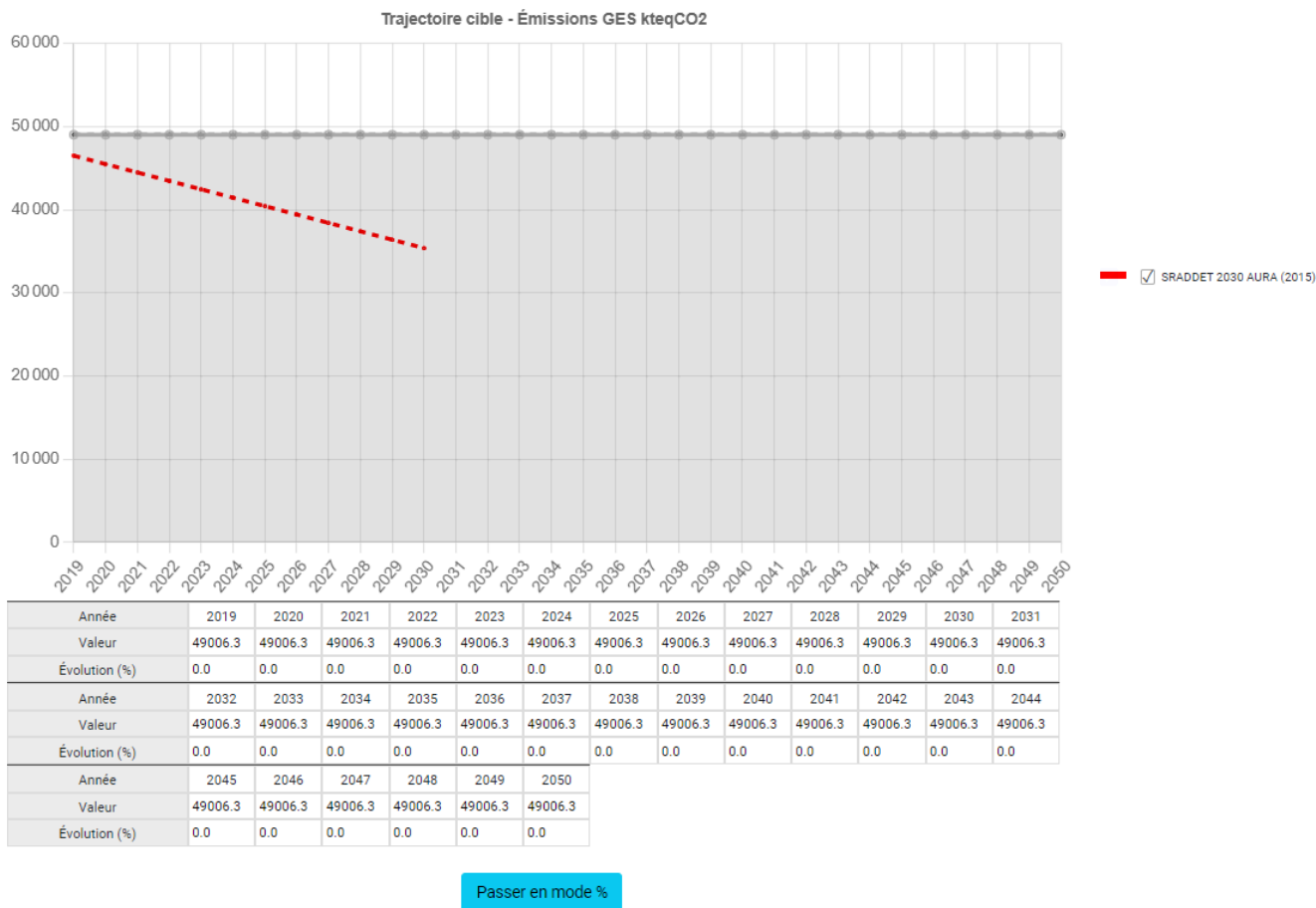
Les fichiers liés à ce composant sont :

- Fichier côté client (Front) : **front/src/components/SuiviEnergetique.js**
- Fichier côté serveur (Back) : **terriapi/terriapi/controller/suivi_trajectoire.py** et **terriapi/terriapi/api.py**

Le module de stratégie territoriale

Les trajectoires cibles

Ce composant permet de visualiser les trajectoires cibles énergie et GES jusqu'à 2050.



Module stratégie territoriale - Trajectoires cibles

Les fichiers liés à ce composant sont :

- Pour la récupération des données énergétiques coté serveur (back) :

`terriapi/terriapi/user.py`

`terriapi/terriapi/controller/scenario.py`

- Pour la restitution de données coté client (front) :

`front/src/components/PlanActionsTrajectoiresCibles.js`

`front/src/components/PlanActionsScenarii.js`

Les plans d'actions

Ce composant permet de simuler les résultats d'impacts d'actions modélisées par des calculs, paramètres et coefficients dont certains sont paramétrables et / ou modifiables depuis l'interface.

Mon territoire en transition **TerriSTORY**
Europe 2014-2020

Se connecter

réduction consommation énergétique dans l'industrie

Agriculture

Réduction consommation énergétique dans l'agriculture

Actions Production renouvelable

Centrale hydroélectrique

Centrale photovoltaïque au sol

Chaudière bois

Eolien

	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
Puissance installée (MW)	0	0	0	0	0	0	0	0	0	0	0	0	0

	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

> Modifier les paramètres économiques

> Modifier les autres paramètres

Installation solaire thermique Combinée (Chauffage et Eau Chaude Sanitaire)

Méthanisation avec injection sur le réseau de gaz

Méthanisation en cogénération

Modification du mix énergétique d'un réseau de chaleur

Réseau de chaleur ENR/gaz

Toiture solaire photovoltaïque – résidentiel petites toitures

Toiture solaire photovoltaïque – tertiaire industrie grandes toitures

Toiture solaire photovoltaïque – Ombrières de parking

Installation solaire thermique Eau Chaude Sanitaire sur logement collectif

Lancer le calcul Export Excel ADEME Réinitialiser les valeurs Réinitialiser les trajectoires cibles

[À PROPOS](#) [CONTACT](#) [MENTIONS LÉGALES](#) [?](#)

Module stratégie territoriale -- Plan d'actions

Cette partie du code peut être divisée en deux scripts de calculs distincts.

- L'un concerne principalement les impacts énergétiques (réduction de consommation et de la facture énergétique, augmentation de la production) et les retombées fiscales. Il repose sur un calcul d'impacts par action d'une part, puis d'une fusion entre l'historique des données et les prévisions jusqu'en 2050 d'autre part. Les calculs d'impacts mobilisent les données des utilisateurs, certaines situées en base de données (nécessaire pour obtenir l'historique sur plusieurs années), et des paramètres indiqués dans des fichiers csv qui font partie intégrante de l'application.
- L'autre concerne les impacts économiques et sociaux (investissement, création de valeur ajoutée et emplois créés). Un seul fichier Python est chargé de construire la requête SQL qui mobilise des paramètres spécifiques construits de toute pièce, la matrice de Leontief d'entrées / sorties, ainsi que les données INSEE sur la nomenclature de l'activité française en 88 branches et la connaissance local de l'appareil productif. Ce dernier jeu de données est utile pour calculer les parts économiques captées par les territoires et la région des différentes phases des projets. Ce pourcentage de part captée est modifiable par les utilisateurs directement depuis l'interface.

Les fichiers liés à ce composant sont :

- Pour la mise en forme de données des impacts énergétiques coté serveur (back) :
terriapi/terriapi/controller/actions.py
- Pour la mise en forme de données des impacts emplois coté serveur (back) :
terriapi/terriapi/actions.py
terriapi/terriapi/controller/suivi_trajectoire.py
terriapi/terriapi/impact_emploi.py
- Pour la restitution de données coté client (front) :

front/src/components/PlanActions.js

Le schéma suivant modélise l'architecture de classe de la partie économique du module stratégie territoriale :

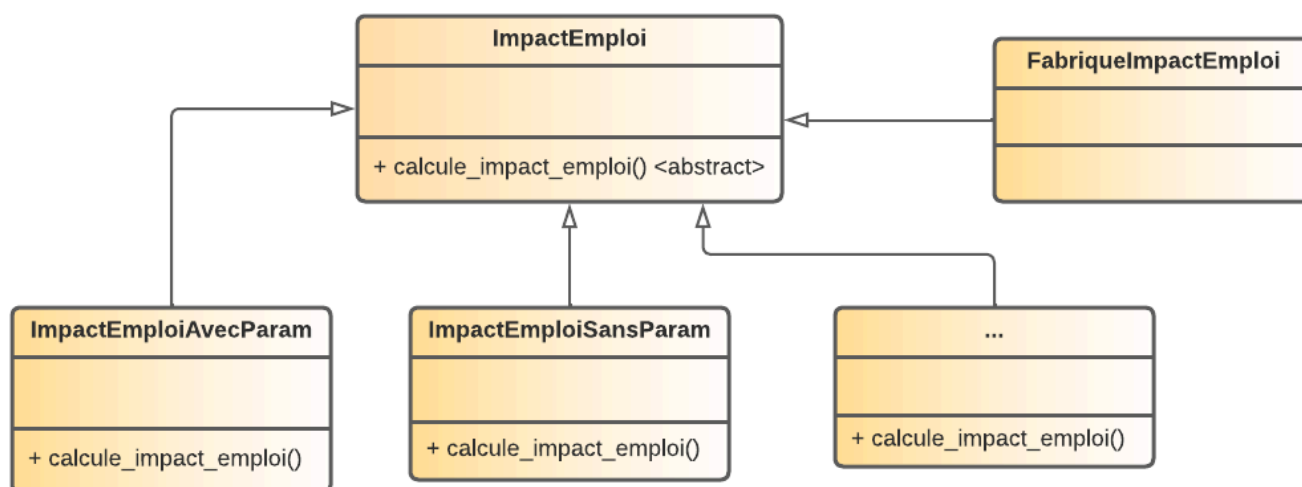


Diagramme de classe du module stratégie territoriale

Le module de tableau de bord

Il s'agit ici de la fonctionnalité de tableau de bord principalement développée en recourant à la conception orientée objet.

La classe **DonneesPourRepresentation** du fichier Python permet de mettre en forme les données pour chacune des représentations. Cette classe transmet ses attributs et méthodes au moyen d'un héritage, à plusieurs classes filles. On compte une classe fille qui hérite de cette classe mère pour chaque représentation.

On peut résumer l'architecture de ces classes au moyen d'un diagramme UML approprié.

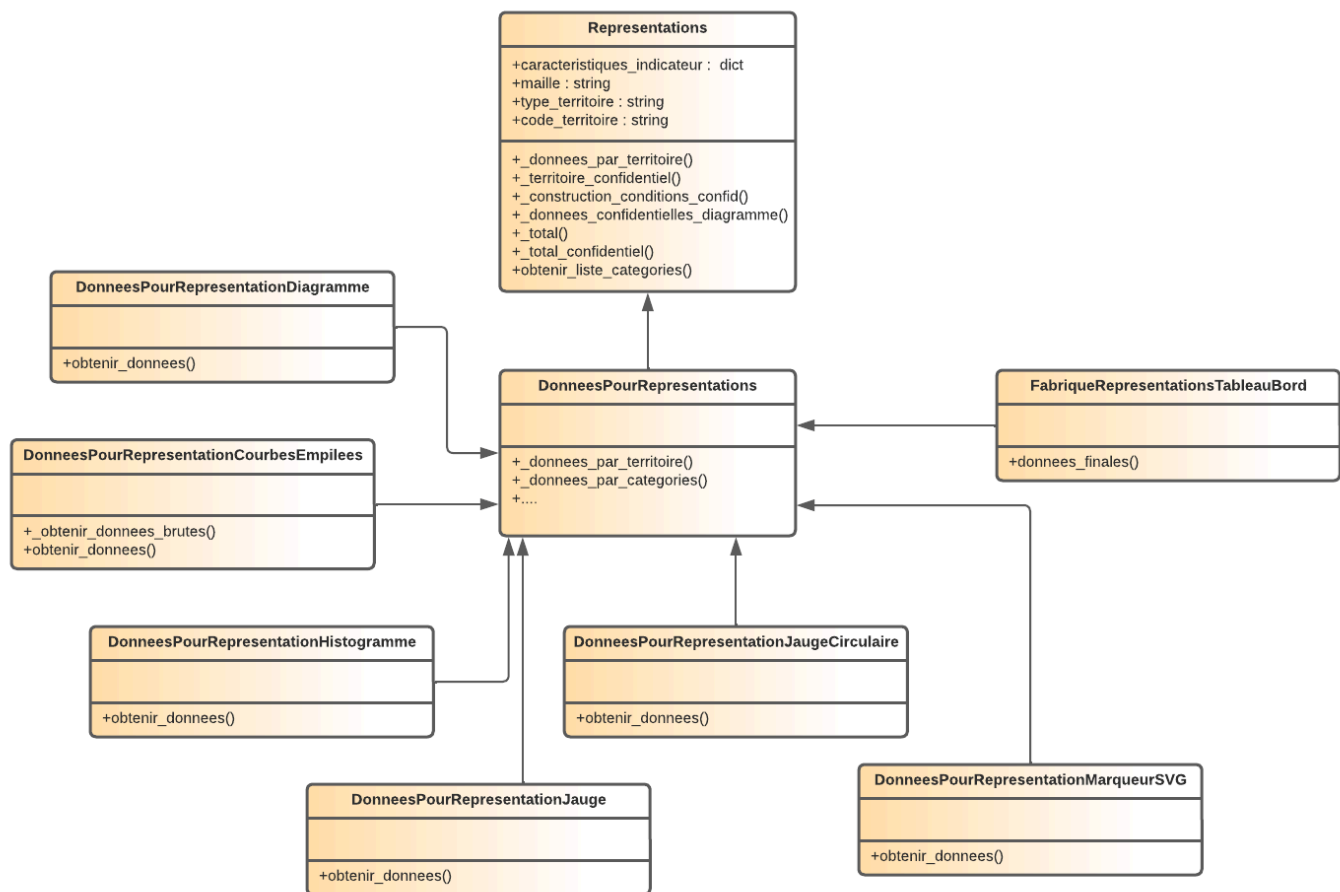


Diagramme de classe des représentations de tableau de bord

Il existe un composant javascript pour chaque représentation. Ces derniers sont instanciés par une fabrique dédiée à laquelle on fait notamment appel dans la fonctionnalité de création de tableau de bord.

Les fichiers liés à ce composant sont :

- Pour la mise en forme de données coté serveur (back) :

`terriapi/terriapi/controller/donnees_pour_representation.py`

- Pour la restitution de données coté client (front) :

`front/src/components/DiagrammeCirculaire.js`

`front/src/components/ DiagrammeRadar.js`

`front/src/components/ CourbesEmpilees.js`

`front/src/components/ CourbesComparees.js`

`front/src/components/ Histogramme.js`

`front/src/components/ Jauge.js`

`front/src/components/ MarqueurSVG.js`

`front/src/components/ FabriqueRepresentation.js`

Les interfaces d'administration

La gestion des utilisateurs

Pour permettre aux utilisateurs d'enregistrer un certain nombre de données et de les charger pour pouvoir y retravailler ou sortir les résultats de leur travail lorsqu'ils le souhaitent, il est nécessaire qu'ils puissent créer un compte et s'authentifier. TerriSTORY® est donc pourvu de fonctionnalités qui permettent d'authentifier un utilisateur en toute sécurité ainsi que d'une interface dédiée au moyen de laquelle on peut gérer les travaux précédemment enregistrés.

Il existe également des fonctionnalités de partage de stratégies territoriales afin que les territoires puissent s'échanger leurs travaux entre eux.

Les fichiers qui permettent de gérer les utilisateurs sont :

- Coté client (front) : **front/src/components/UsersManage.js**

front/src/components/Auth.js

- Coté serveur (back) : **terriapi/terriapi/controller/auth.py**

La gestion des interfaces d'administration

En outre, on distingue un autre genre d'acteur, Il s'agit des administrateurs qui disposent en plus d'une interface spécifique qu'il est prévu d'enrichir et grâce à laquelle l'administrateur peut faire plusieurs actions :

- Activer / désactiver / supprimer les comptes utilisateurs
- Désactiver / activer les indicateurs
- Rendre modifiable / non modifiables les couches d'équipements.
- Personnaliser le pop-up d'accueil
- ...

The screenshot displays the 'Administration' section of the TerriSTORY web application. At the top, there is a navigation bar with 'Administration', 'Mes stratégies territoriales', and 'Mes tableaux de bord'. Below this, a sub-menu includes 'Infos personnelles', 'Utilisateurs', 'Indicateurs, données et territoires', 'Equipements', 'Historique', 'Objectifs', 'Personnalisation pop-up accueil', and 'Gestion des données communes'. The main content area is titled 'Mes informations' and shows the following details:

- Identifiant : **terristory@auvergnehonealpes-ee.fr**
- Prénom :
- Nom :
- Organisation :
- Fonction :
- Territoire :
- Profil : **admin**

At the bottom of this section, there are two buttons: 'Demande de suppression de compte' and 'Changer mon mot de passe'. The footer of the page contains links for 'À PROPOS', 'CONTACT', and 'MENTIONS LÉGALES ?'.

Le fichier qui permet de gérer les interfaces d'administration est : **front/src/components/Profile.js**

Le composant principal côté client

Au sein de ce composant **front/src/Index.js**, on distribue les données dont on a besoin pour rendre fonctionnels tous les autres composants et on met à disposition un certain nombre de fonctions utilisées un peu partout dans le code JavaScript de l'application. Dans la plupart des cas, le rôle de ces fonctions est de mettre à jour l'état du composant principal afin de mettre à jour les données mobilisées par ailleurs dans le reste de l'application.

Il constitue la porte d'entrée par laquelle on accède aux différentes pages et où on fait appel à l'ensemble des composants de l'application.

Le composant principal côté serveur

Le fichier **terriapi/terriapi/api.py** est le fichier qui fait le lien entre les différentes composantes de l'application et les fichiers python approprié (**api**).