

# SmartCAPEX OCM – Documentation APPLI WEB



*<BACKEND & FRONTEND >*

Rédigé par	Gabin TIEN
Validé par	
Dernière modification	03/10/2021
Version	1.0.0

## I. Introduction

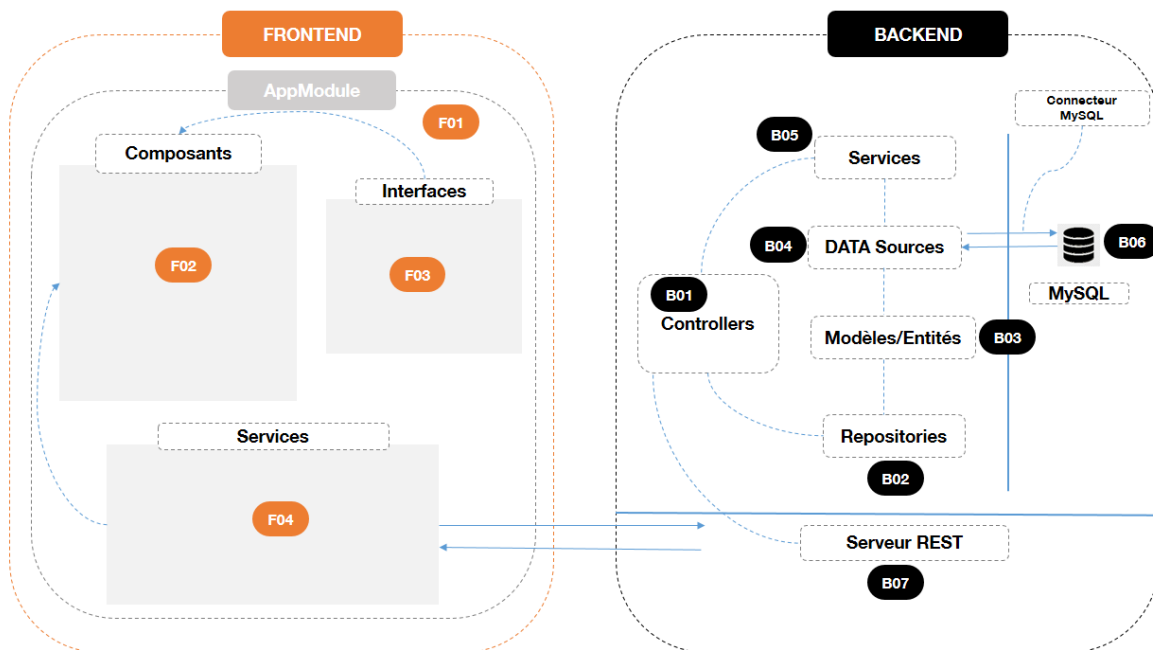
Smart CAPEX optimise les activités de déploiement du réseau en termes de valeur commerciale. Il permet l'évolution des décisions basées sur des métriques techniques vers des métriques basées sur la valeur, basées sur le nombre de clients concernés et leur contribution aux activités OCM.

Indépendamment des valeurs d'entrée et des critères, SmartCAPEX fournit un classement cohérent des actions de réseau évaluées et les hiérarchise. En outre, il estime la valeur totale créée à partir de la marge supplémentaire associée à chaque investissement spécifique.

Ce document décrit l'application web (Backend & Frontend) qui permet de :

- *Restituer les résultats d'analyse SmartCAPEX,*
- *Faire des simulations de planifications réseaux.*

## II. Architecture fonctionnelle de l'application WEB



**Schema 1.** Architecture fonctionnelle de l'application web

L'application web est composée de deux grandes parties à savoir le Frontend et le Backend. Des Endpoints sont développés au niveau Backend et ceux-ci sont consommés par la partie Frontend.

### 1. Frontend

Le frontend est la partie visuelle de l'application web ; l'interface utilisateur. Il est développé avec le framework Angular (**version 10.2.4** plus précisément) et est composé de plusieurs éléments :

- Un module principal AppModule,
- Des composants,
- Des interfaces,
- Des services.

#### a. AppModule (FO1)

Le module **AppModule** regroupe un ensemble d'éléments clés à savoir les composants, les interfaces et les services. Ces éléments communiquent entre eux ; les services et les interfaces sont utilisés dans les différents composants pour des fonctionnalités bien définies. Ce module principal requiert d'autres modules tiers pour pouvoir fonctionner. Entre autres nous pouvons citer :

- BrowserModule,

- AppRoutingModuleModule
- HttpClientModule,
- FormsModule,
- ReactiveFormsModule,
- NgxUiLoaderModule,
- NgxUiLoaderHttpModule,
- BrowserAnimationsModule,
- ToastrModule,
- ngxPaginationModule,
- NgSelectModule,
- CalendarModule

### b. Les interfaces (FO3)

Le Frontend comprend également les interfaces qui sont utilisées dans les composants pour garantir une fiabilité au niveau des structures de données. Nous avons :

- Interface **OpexStructure**: définit la structure des données Opex Structures,

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
DATA	number	
VOIX	number	

- Interface **Constante** : définit la structure des constantes,

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
identifiant	string	
parameter	number	
value	number	

- Interface **Site** : définit la structure des Sites Techniques,

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>siteId</i>	string	
<i>sitePhysique</i>	string	
<i>siteStatus</i>	string	
<i>siteVille</i>	string	
<i>siteGestionnaire</i>	string	
<i>siteQuartier</i>	string	
<i>siteCommune</i>	string	
<i>siteDepartment</i>	string	
<i>siteRegion</i>	string	
<i>siteDistrict</i>	string	
<i>siteZoneCommerciale</i>	string	
<i>siteLongitude</i>	string	
<i>siteLatitude</i>	string	
<i>siteTowerHeight</i>	string	
<i>siteTypeBaie</i>	string	
<i>siteGeotype</i>	string	
<i>siteEnergyType</i>	string	
<i>opexKey</i>	string	

- Interface **OpexTelco** : définit la structure des données Opex Telco,

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>YEAR</i>	string	
<i>TYPE</i>	string	
<i>1</i>	number	
<i>2</i>	number	
<i>3</i>	number	
<i>4</i>	number	

- Interface **Planification** : définit la structure des planifications

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>name</i>	string	
<i>budget</i>	number	
<i>zonesConsidered</i>	object[ ]	
<i>sitesConsidered</i>	sites[ ]	
<i>createdBy</i>	string	
<i>updatedBy</i>	string	
<i>validatedBy</i>	string	
<i>sitesExcluded</i>	sites[ ]	
<i>impactDuration</i>	string	
<i>etatValidation</i>	string	
<i>wacc</i>	string	
<i>steeringNetwork</i>	steeringNetwork[ ]	
<i>dateDebut</i>	string	
<i>anneeDeploiement</i>	string	
<i>statutAnalyse</i>	string	

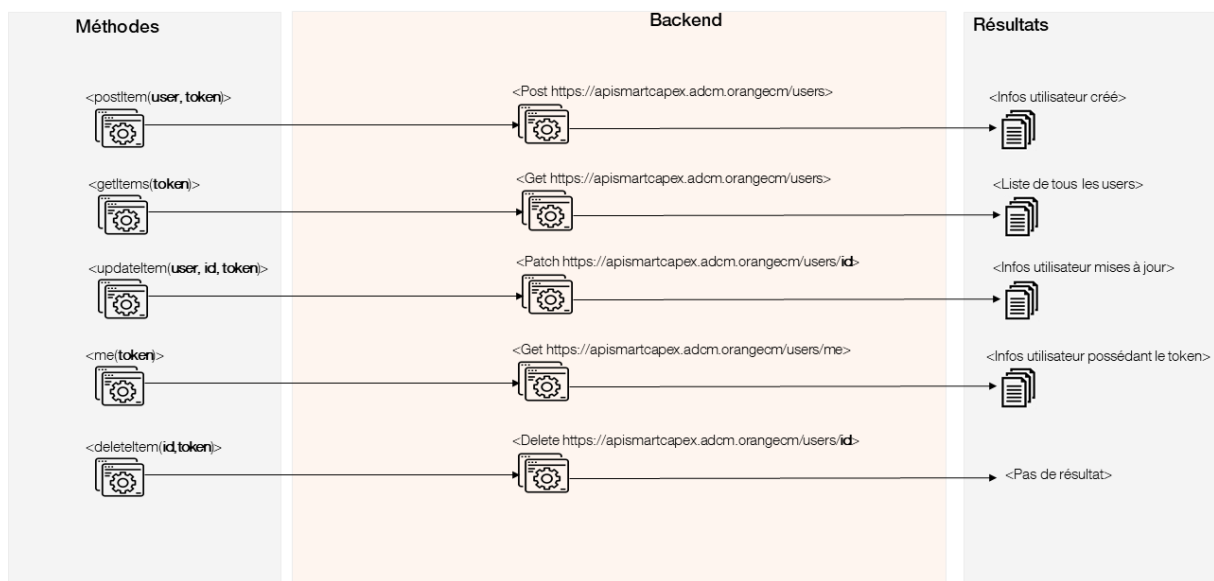
### c. Les services (FO4)

Les services permettent de partager des informations entre les composants de l'application web SmartCAPEX. Ils permettent également de faire toutes les requêtes vers le Backend ; ce sont des méthodes CRUD (Create, Read, Update & Delete) qui consomment les différentes APIs fournies par le Backend. Ci-dessous le récapitulatif des différents services ainsi que les méthodes associées.

- **UserService**

Le service userService permet de faire toutes les requêtes CRUD concernant les utilisateurs de l'application SmartCAPEX. La structure des données implémente l'**Interface Site** vue précédemment.

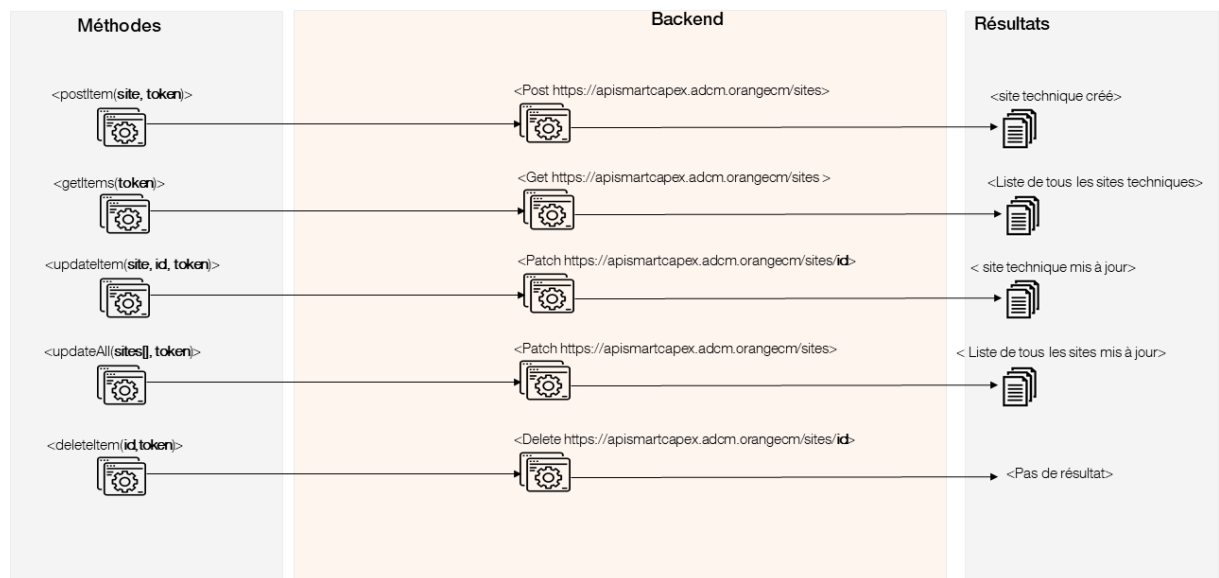
Ci-dessous, le schéma récapitulatif de l'ensemble des méthodes.



**Schema 2.** Ensemble des méthodes du service userService

- **SitesService**

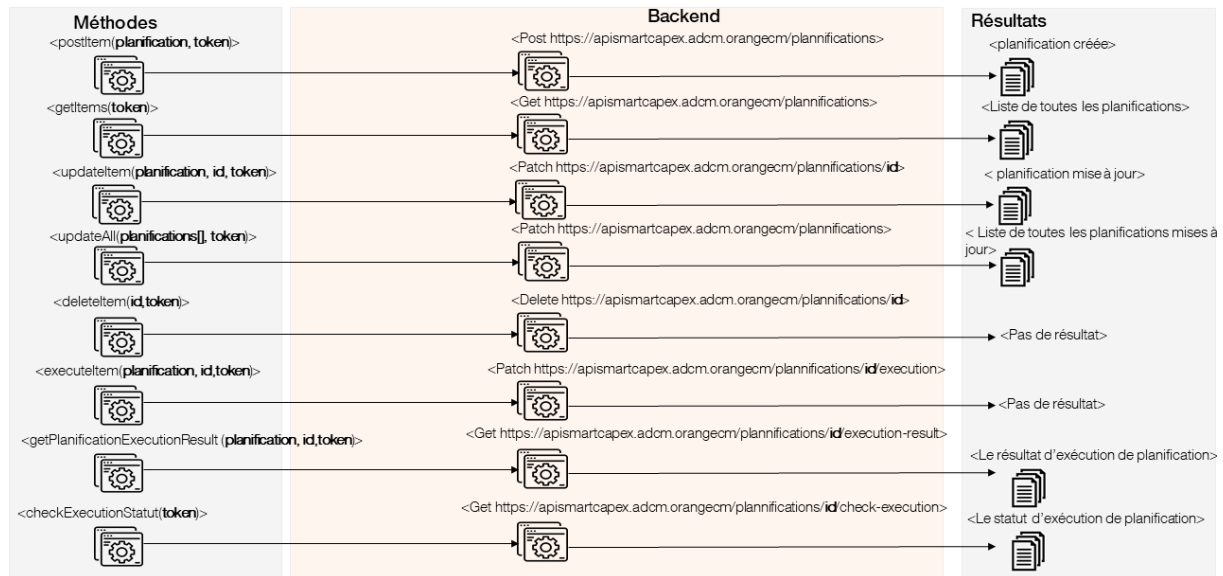
Le service SitesService permet de faire toutes les requêtes CRUD concernant les sites techniques. La structure des données implémente l'**Interface Site** vue précédemment.



**Schema 3.** Ensemble des méthodes du service SitesService

- **PlanificationsService**

Le service PlanificationsService est celui qui est en charge des planifications. Il permet entre autres de faire les requêtes CRUD mais aussi l'exécution, la récupération des résultats d'exécution et la vérification des statuts d'exécution des planifications. La structure des données implémente l'**Interface Planification** vue précédemment.

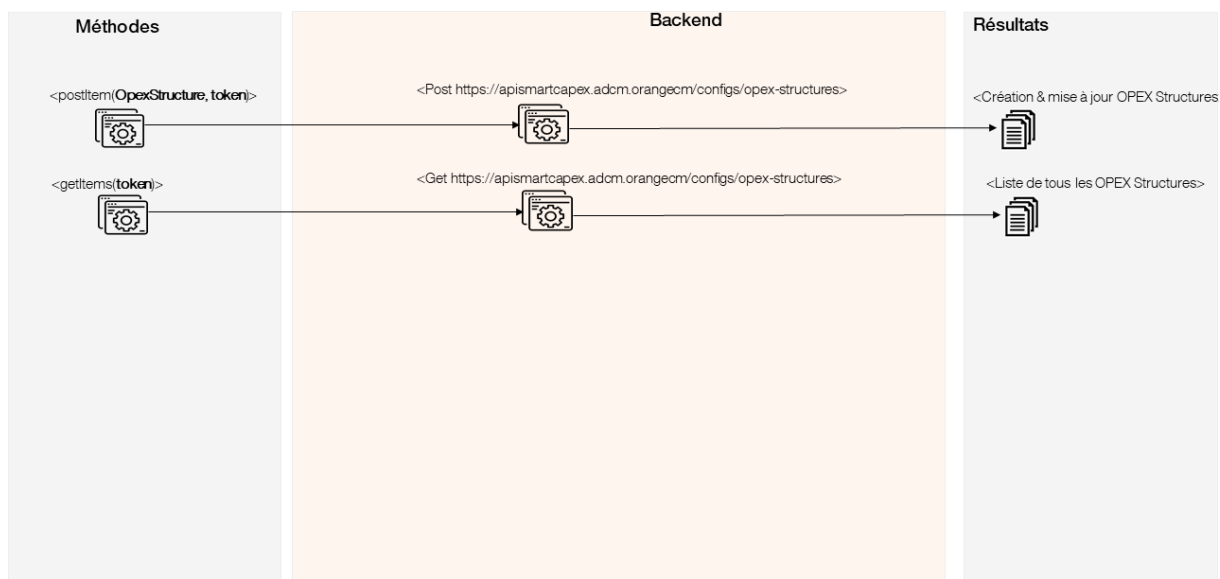


**Schema 4.** Ensemble des méthodes du service PlanificationService

- **OpexStructureService**

Le service OpexStructureService permet de faire toutes les requêtes CRUD concernant les Opex Structures. La structure des OPEX Structures implémente l'**Interface OpexStructure** vue plus haut.

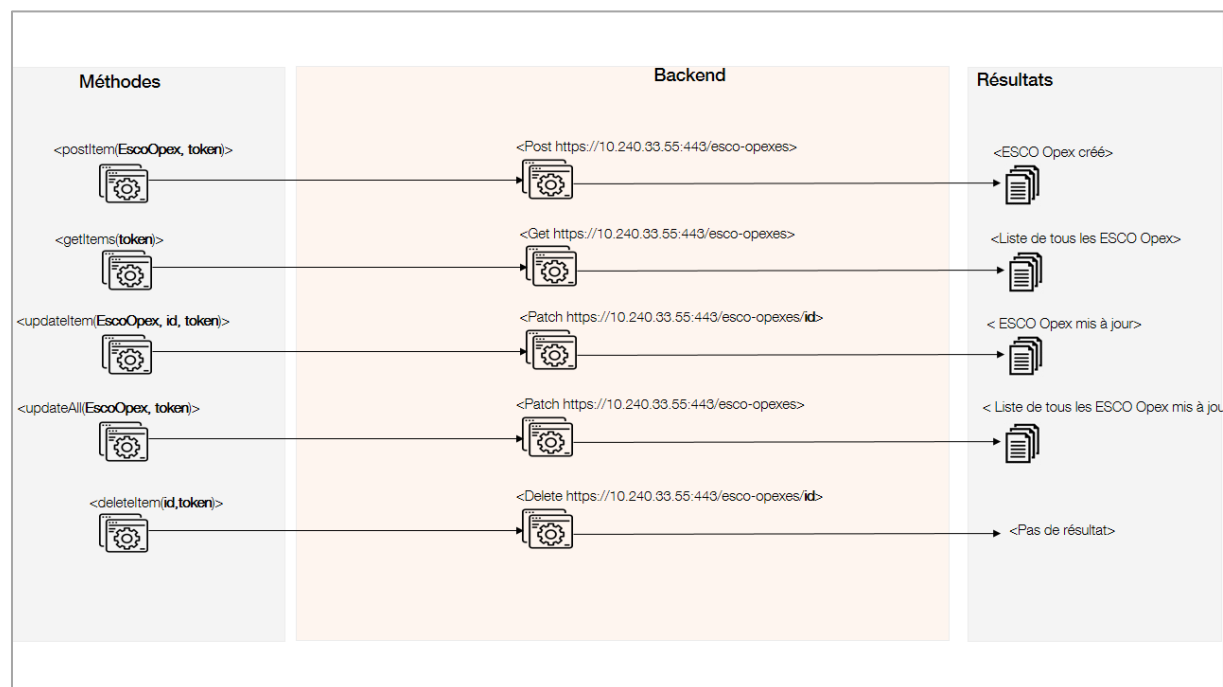




**Schema 5.** Ensemble des méthodes du service OpexStructureService

- **OpexTelcoService**

Le service OpexTelcoService permet de faire toutes les requêtes CRUD concernant les Opex Telcos. La structure des OPEX Telcos implémente l'**Interface OpexTelco** vue précédemment.

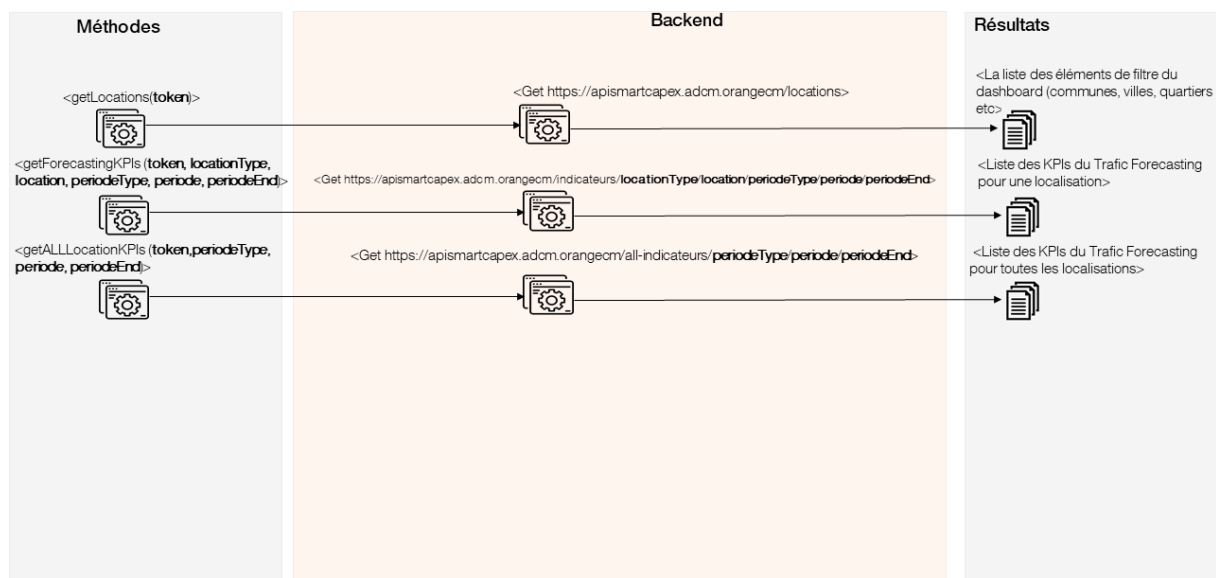


**Schema 6.** Ensemble des méthodes du service OpexTelcoService

- **DashboardService**

Le service DashboardService permet de récupérer les KPIs du Traffic Forecasting Dashboard. Il comprend trois méthodes. Ci-dessous le récapitulatif des paramètres requis pour ces méthodes.

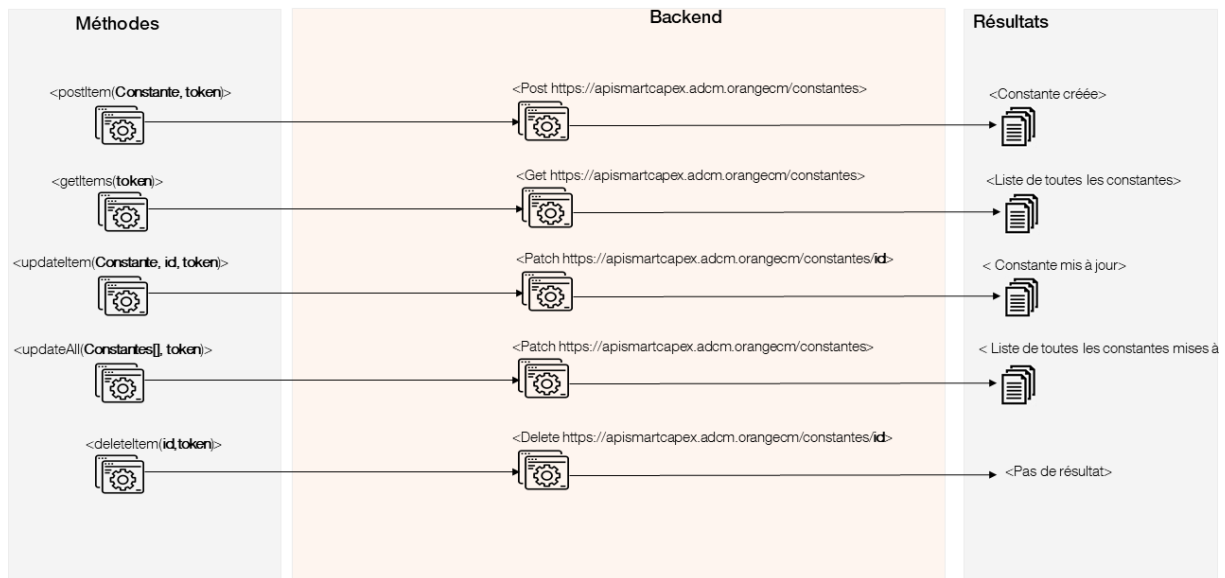
Paramètres	Valeurs possibles	Commentaire
token	Le token de l'utilisateur connecté	Le token donne les autorisations à l'utilisateur de faire les requêtes backend.
locationType	site_physique, site_quartier,site_ville, site_commune, site_department, site_region, site_district, site_zone_commerciale	
location	La valeur de locationType	
periodeType	week_period, year, date	
periode	la valeur de periodeType	



**Schema 7.** Ensemble des méthodes du service DashboardService

- **ConstanteService**

Le service ConstanteService permet de faire toutes les requêtes CRUD concernant les Constantes. La structure des **Constantes** implémente l'Interface Constante vue précédemment.

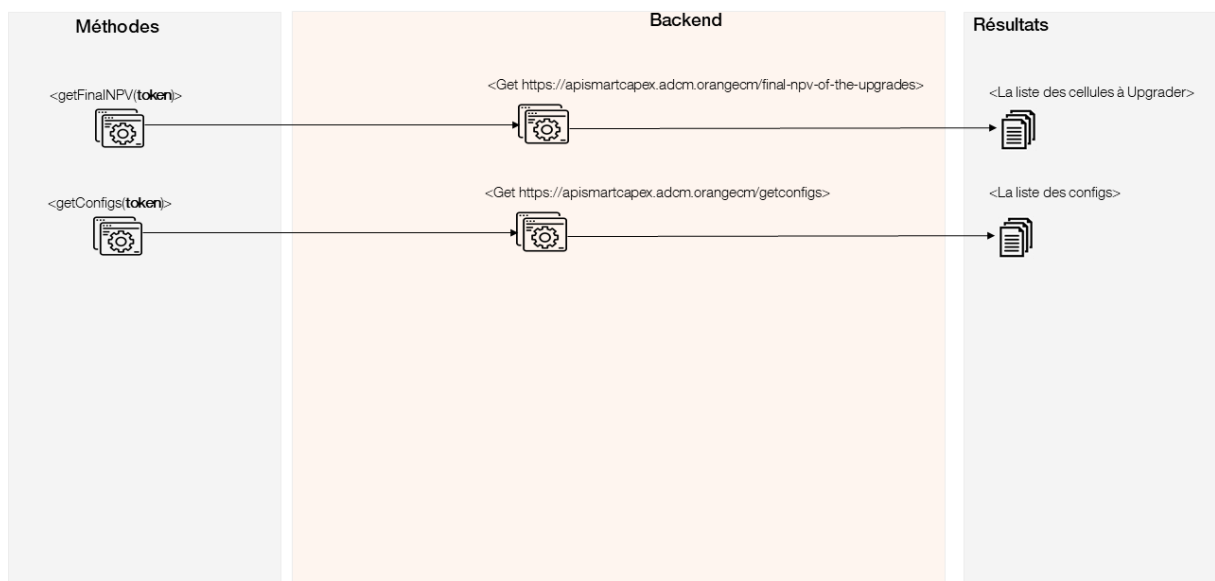


**Schema 8.** Ensemble des méthodes du service ConstanteService

- **BackendService**

Il y a deux méthodes dans le service BackendService :

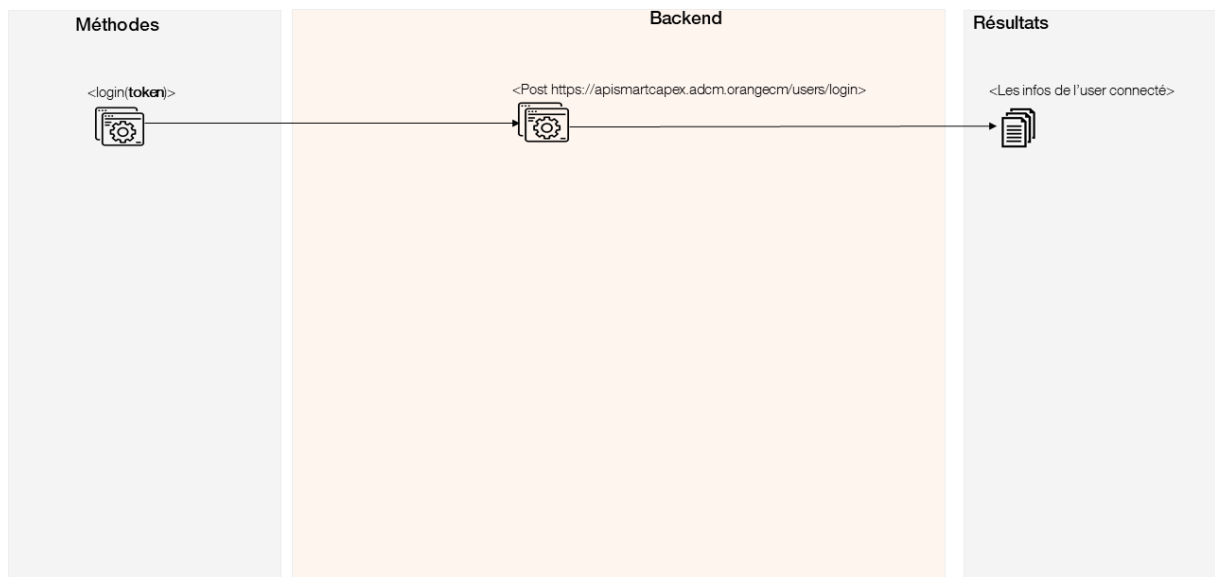
- La première (**getFinalNPV(token)**) permet de récupérer les résultats du module capacité,
- La deuxième, quant à elle permet de récupérer la liste des configurations SmartCAPEX.



**Schema 9.** Ensemble des méthodes du service BackendService

- **AuthService**

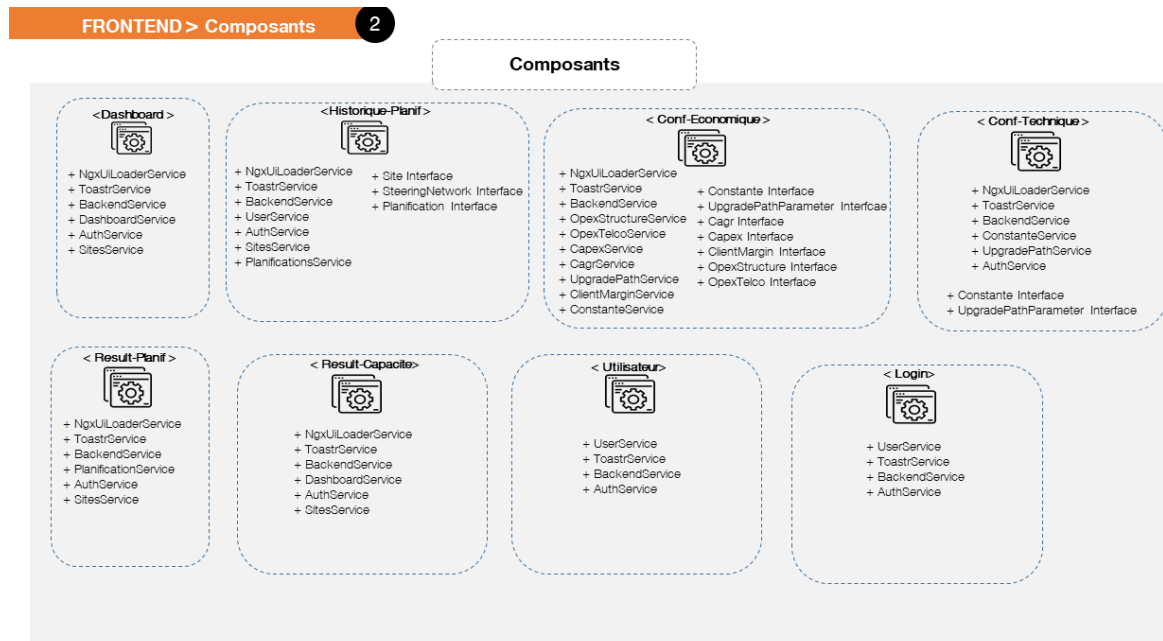
Le service AuthService permet de gérer l'aspect sécurité de l'application web SmartCAPEX. Il contient la méthode de login.



**Schema 10.** Ensemble des méthodes du service AuthService

#### d. Composants (FO2)

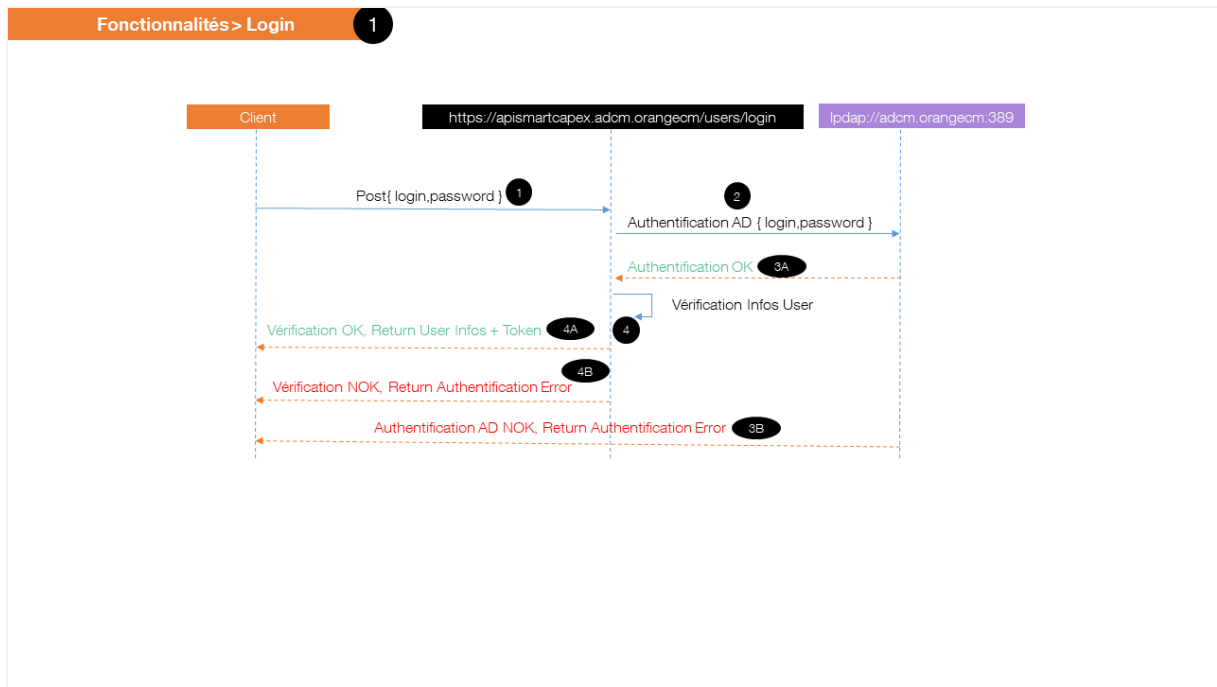
Les composants sont des éléments très importants au niveau de l'application web SmartCAPEX. Ils permettent de gérer les différentes vues et utilisent de façon générale les méthodes définies dans les services. Ci-dessous la liste des composants clés ainsi que le résumé de ces différentes méthodes. Un accent particulier sera mis l'authentification.



**Schema 11.** Ensemble des composants clés du Frontend avec les services associés

- **Login**

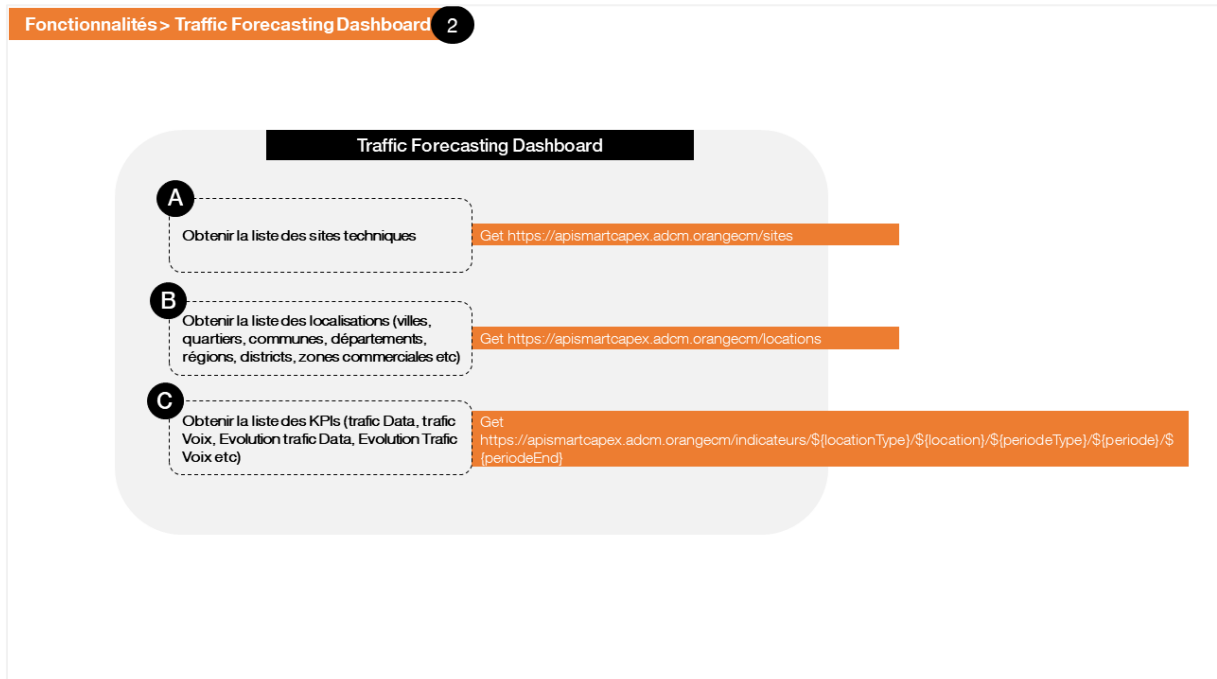
Le login est la première interface que l'utilisateur a en utilisant l'application SmartCAPEX ; il contient le formulaire de connexion. Ci-dessous le résumé du processus de connexion.



**Schema 12.** Processus d'authentification

- **Le Traffic Forecasting Dashboard**

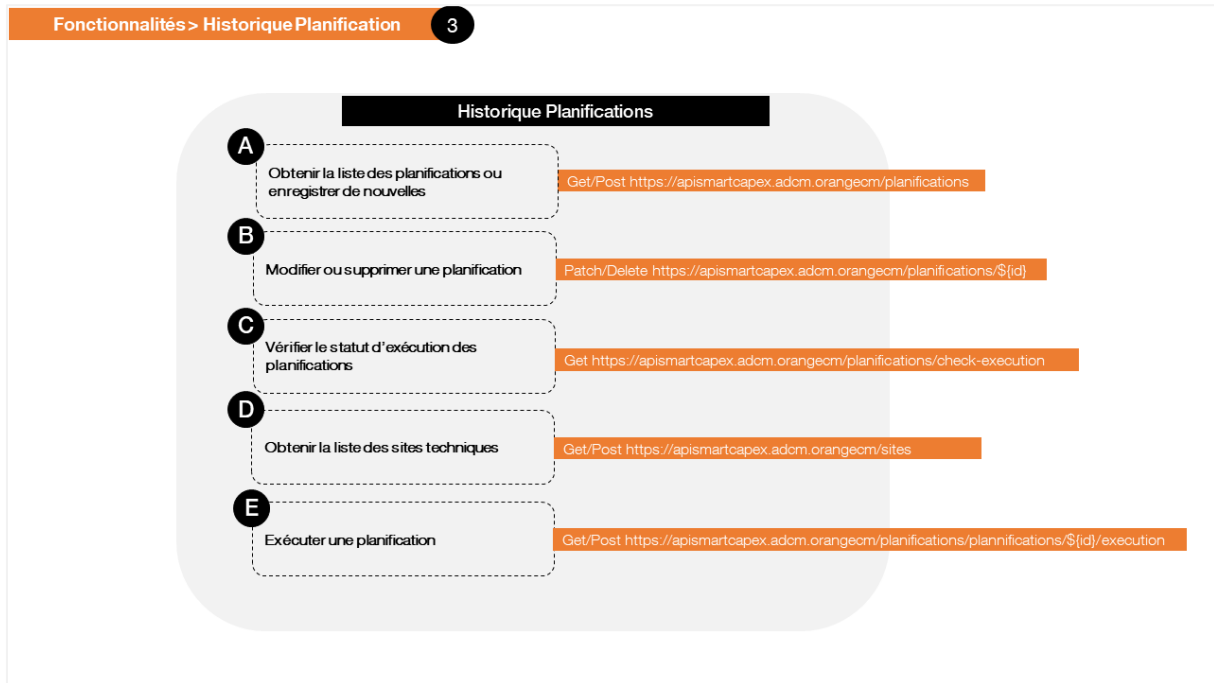
Le Traffic Forecasting Dashboard est l'interface de visualisation des résultats des prédictions du trafic data et voix. Ci-dessous l'ensemble des méthodes utilisées pour récupérer les données.



**Schema 13.** Liste des URIs utilisées dans la vue Traffic Forecasting Dashboard

- **Historique de planification**

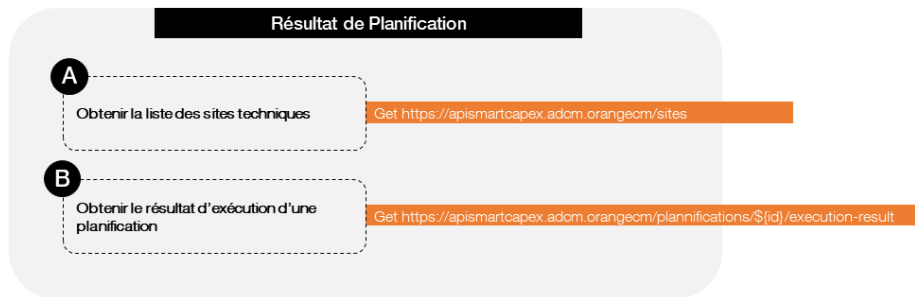
Le composant Historique de Planification est l'interface de gestion des planifications. Ci-dessous, la liste des URIs consommées dans ce composant.



**Schema 14.** Liste des URIs utilisées dans la vue Historique de Planification

- **Résultat de Planification**

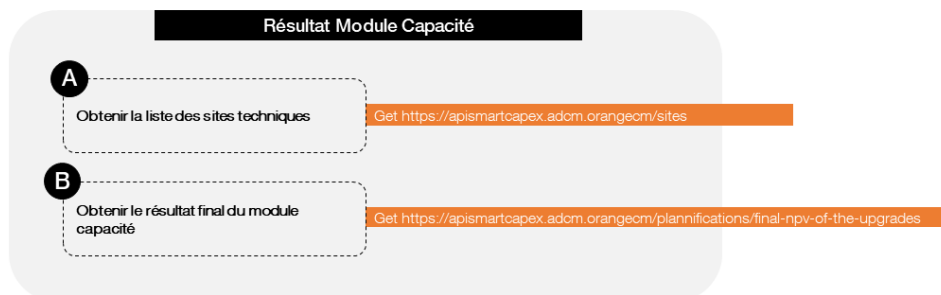
Le Résultat de Planification est le composant qui permet de restituer les résultats d'exécution des planifications. Ci-dessous, la liste des URIs consommées dans ce composant.



**Schema 15.** Liste des URIs utilisées dans la vue Résultat de Planification

- **Résultat du Module Capacité**

Le Résultat du Module Capacité est le composant qui permet de restituer les résultats du module Capacité de SmartCAPEX. Ci-dessous le résumé des URIs consommées.

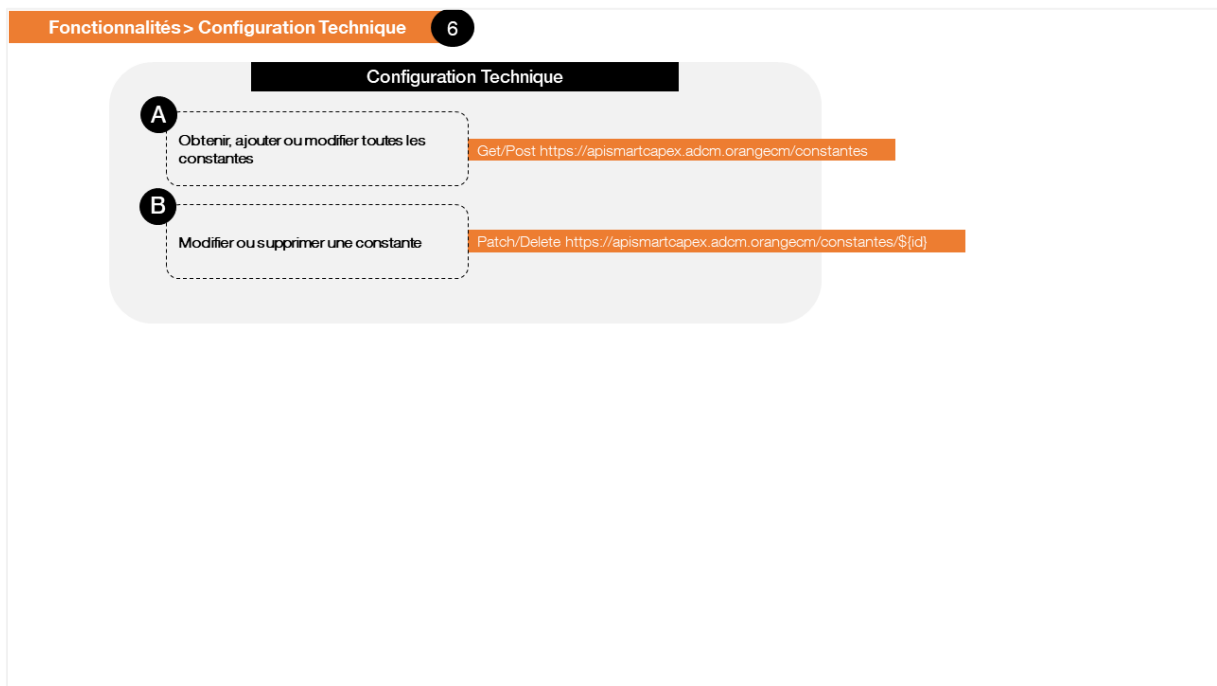


**Schema 16.** Liste des URIs utilisées dans la vue Résultat du module Capacité



- **Configuration Technique**

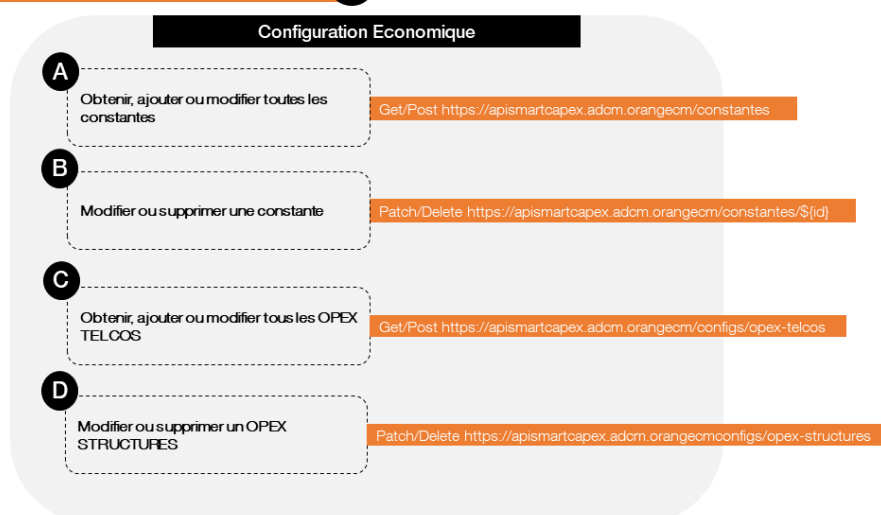
La configuration Technique est le composant utilisé pour la gestion des configurations techniques de SmartCAPEX. Ci-dessous l'ensemble des URIs consommées.



**Schema 17.** Liste des URIs utilisées dans la vue Configuration Technique

- **Configuration Economique**

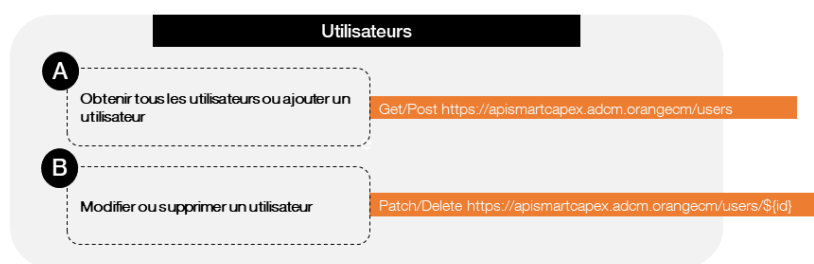
La Configuration Economique est le composant utilisé pour la gestion des configurations économiques de SmartCAPEX. Ci-dessous l'ensemble des URIs consommées.



**Schema 18.** Liste des URIs utilisées dans la vue Configuration Economique

- **Utilisateurs**

Le composant Utilisateurs est celui qui permet de gérer l'ensemble des utilisateurs de l'application web SmartCAPEX. Ci-dessous, la liste des URIs consommées.



**Schema 20.** Liste des URIs utilisées dans le composant Utilisateurs

## 2. Le Backend

Le backend est la partie de l'application web qui permet d'exposer à la partie frontend des URIs pour les différentes fonctionnalités. Il est développé avec le framework LoopBack dans sa version 4. Il comprend :

- Un serveur REST exposé au Frontend,
- Des entités/modèles,
- Des repositories associés aux entités/modèles,
- Des Controllers dans lesquelles les URIs sont développées,
- Des services qui apportent des fonctionnalités supplémentaires aux autres composants,
- Des Data Sources qui donnent des accès aux bases de données.

Dans la suite de cette partie nous allons voir les URIs développées dans les Controllers et les entités ; les autres parties étant plus techniquement des composants du Framework Loopback.

### a. Controllers

Les controllers sont les composants dans lesquels toutes les URIs sont développées.

La liste de ces URIs avec les différents paramètres est accessible à l'adresse :

<https://apismartcapex.adcm.orangebcm/planifications/explorer/>

### b. Entités/Modèles

- Le modèle **User**

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>id</i>	<i>number</i>	
<i>uName</i>	<i>string</i>	
<i>firstname</i>	<i>string</i>	
<i>uLogin</i>	<i>string</i>	
<i>email</i>	<i>string</i>	
<i>isActive</i>	<i>string</i>	
<i>createdAt</i>	<i>datetime</i>	
<i>updatedAt</i>	<i>datetime</i>	
<i>createdBy</i>	<i>string</i>	
<i>updatedBy</i>	<i>string</i>	
<i>role</i>	<i>string</i>	
<i>phoneNumber</i>	<i>string</i>	

- Le modèle **Login** : la structure des données utilisées pour l'authentification

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>login</i>	<i>number</i>	
<i>password</i>	<i>string</i>	

- Le modèle **Location** : la structure des données utilisées pour les différents filtres au niveau du trafic forecasting dashboard

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>villes</i>	<i>string[ ]</i>	
<i>sitePhysiques</i>	<i>string[ ]</i>	
<i>regions</i>	<i>string[ ]</i>	

- Le modèle **PlanificationExecution** : la structure des données attendues pour l'exécution d'une planification

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>id</i>	<i>number</i>	<i>L'id de la planification à exécuter</i>
<i>wacc</i>	<i>number</i>	
<i>sitesConsidered</i>	<i>object[ ]</i>	<i>Les sites considérés dans la planification</i>

- Le modèle **FinalNpvOfTheUpgrade** : la structure des données du résultat final du module capacité

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>id</i>	<i>number</i>	
<i>siteId</i>	<i>string</i>	
<i>cellBand</i>	<i>string</i>	
<i>cashFlowYear0</i>	<i>number</i>	
<i>cashFlowYear1</i>	<i>number</i>	
<i>cashFlowYear2</i>	<i>number</i>	
<i>cashFlowYear3</i>	<i>number</i>	
<i>cashFlowYear4</i>	<i>number</i>	
<i>cashFlowYear5</i>	<i>number</i>	

- Le modèle **DfPredictedTrafficKpis** : la structure des données du résultat du trafic forecasting

<i><b>Champ</b></i>	<i><b>Type</b></i>	<i><b>Commentaire</b></i>
<i>siteId</i>	<i>string</i>	
<i>sitePhysique</i>	<i>string</i>	
<i>siteStatus</i>	<i>string</i>	
<i>siteVille</i>	<i>string</i>	
<i>siteGestionnaire</i>	<i>string</i>	
<i>siteQuartier</i>	<i>string</i>	
<i>siteCommune</i>	<i>string</i>	
<i>siteDepartment</i>	<i>string</i>	
<i>siteRegion</i>	<i>string</i>	
<i>siteDistrict</i>	<i>string</i>	
<i>siteZoneCommerciale</i>	<i>string</i>	
<i>siteLatitude</i>	<i>string</i>	
<i>siteLongitude</i>	<i>string</i>	
<i>siteTowerHeight</i>	<i>string</i>	
<i>siteTypeBaie</i>	<i>string</i>	
<i>siteGeotype</i>	<i>string</i>	
<i>siteEnergyType</i>	<i>string</i>	
<i>opexKey</i>	<i>string</i>	
<i>cellNmame</i>	<i>string</i>	

<i>date</i>	<i>date</i>
<i>cellTech</i>	<i>string</i>
<i>cellBand</i>	<i>string</i>
<i>cellSector</i>	<i>number</i>
<i>year</i>	<i>number</i>
<i>week</i>	<i>number</i>
<i>weekPeriod</i>	<i>number</i>
<i>totalDataTrafficDIGb</i>	<i>number</i>
<i>totalVoiceTrafficKerlands</i>	<i>number</i>
<i>averageThroughputUserDIKbps</i>	<i>number</i>
<i>cellOccupationDIPercentage</i>	<i>number</i>
<i>lostTrafficKerland</i>	<i>number</i>
<i>averageNumberOfUsersInQueue</i>	<i>number</i>

- Le modèle **Cagr** (voir ***interface Cagr*** dans la partie Frontend)
- Le modèle **Constante** (voir ***interface Constante*** dans la partie Frontend)
- Le modèle **Planification** (voir ***interface Planification*** dans la partie Frontend)
- Le modèle **SitesConsidered** (voir ***interface Site*** dans la partie Frontend)
- Le modèle **SitesExluded** (voir ***interface Site*** dans la partie Frontend)
- Le modèle **Site** (voir ***interface Site*** dans la partie Frontend)
- Le modèle **SteeringNetwork** (voir ***interface SteeringNetwork*** dans la partie Frontend)