

Les API



Florian DEGRAND

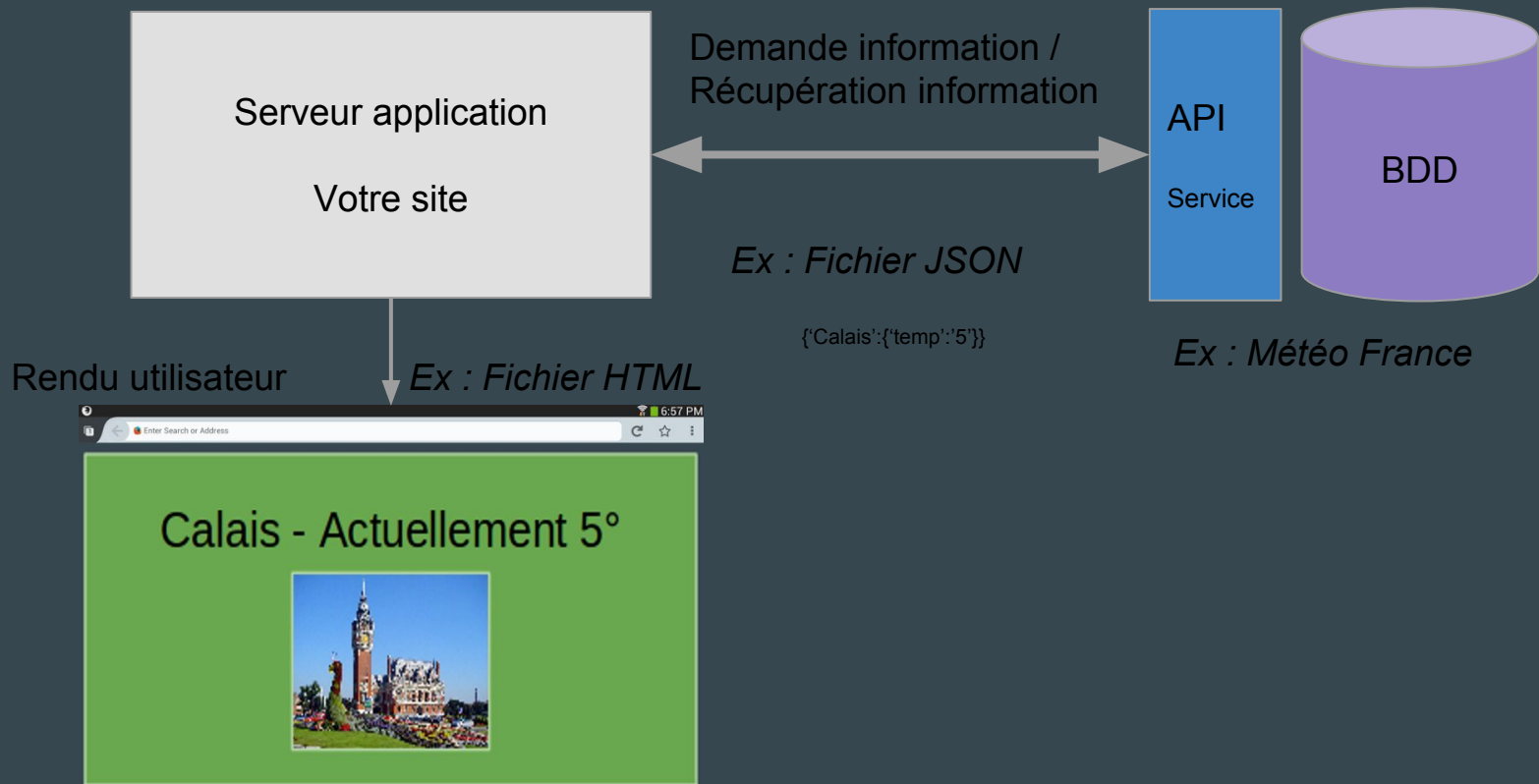
Cours 2

1) Introduction

Qu'est ce qu'une API?

Si les APIs n'existaient pas?

2) Schéma



3) Définition

Une API est un programme informatique.

C'est une interface de façade normalisé accessible sur le web (ou dans certains cas limité sur un réseau. Ex : entreprise)

Elle permet de “se brancher” à une application pour échanger des données.

Lorsque l'on utilise une API:

- On ne connaît pas les détails internes de fonctionnement.
- Permet d'être de plus en plus connecté, favoriser les interactions
 - Innover plus rapidement
 - De plus en plus utilisé pour le mobile

4) Caractéristiques

Pour assurer la bonne utilisation d'une API,
plusieurs caractéristiques importantes :

- Accessible via protocole HTTP/HTTPS
- Documentation utilisation (Ex : Swagger - <http://petstore.swagger.io>)
- Gestion de la sécurité (*Droit utilisation avec clef d'accès*)
- Gestion de la consommation (*limite & quota*)
 - Gestion de pricing (*coût*)
 - ...

6) Normalisation - Architecture RESTful

L'architecture « Representational State Transfer » consiste principalement en la mise en place de services basés sur le protocole HTTP pour les opérations standard de « CRUD » .

L'acronyme informatique anglais **CRUD** (pour *create, read, update, delete*) (parfois appelé SCRUD avec un "S" pour *search*) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

Soit :

- **Create** : créer
- **Read** : lire
- **Update** : mettre à jour
- **Delete** : supprimer

6) Normalisation

Plusieurs verbes dans le protocoles HTTP que l'on va implémenter dans CRUD :

MÉTHODE HTTP	ÉTAPE CRUD
POST	Create
GET	Read
PUT	Update
DELETE	Delete

6) Normalisation

La notion de service RESTful est très normée, et quelques règles doivent être respectées pour se réclamer RESTful:

- Les méthodes POST/GET/PUT/DELETE sont implémentées pour manipuler la collection et/ou les entités, sur le modèle du CRUD.
- Les méthodes PUT et DELETE sont « idempotentes », ce qui signifie que d'effectuer plusieurs fois la même requête aura le même effet que de l'exécuter une seule fois.
- La méthode GET est « sûre », ce qui signifie qu'elle ne modifie pas l'état du serveur ni les données (à l'inverse de POST/PUT/DELETE).
- STANDARDS échanges de données : fichiers JSON/XML

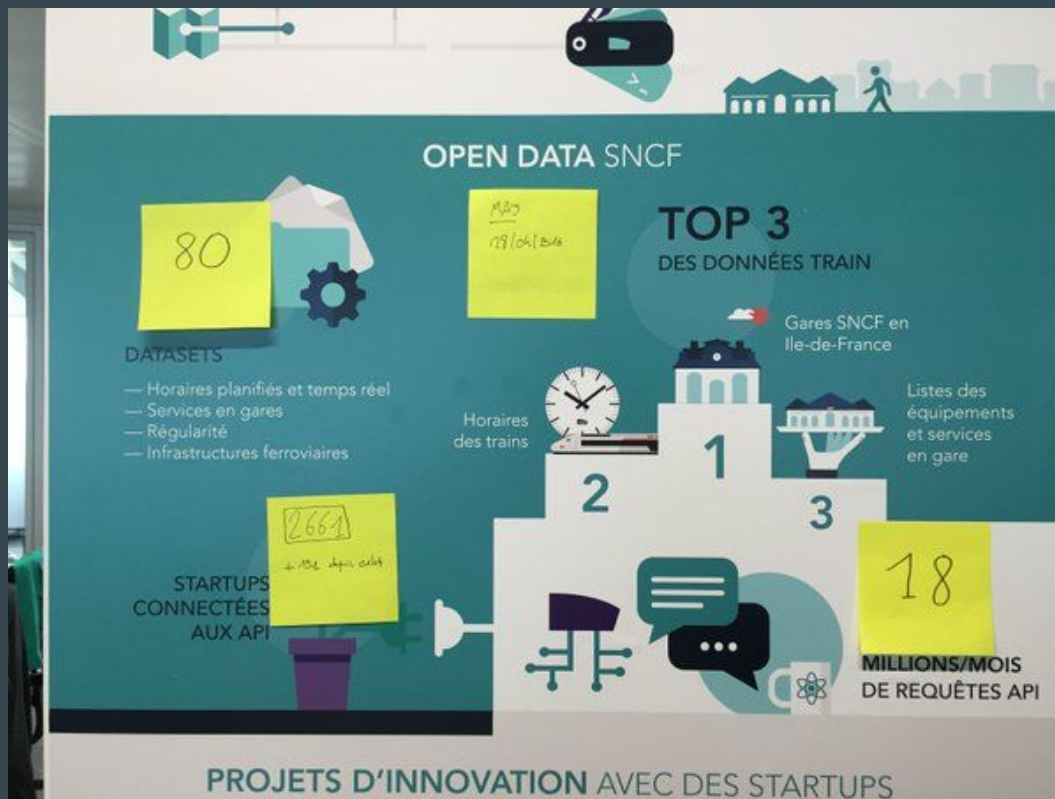
5) Exemple d'API

Les APIs existent pour beaucoup de domaine :

- cartographie (MAPS) Ex : Google / Open Street Map
- envoie de SMS / mails
- vidéos/photos
- boutons connectés (type “like”) beaucoup d'API public par Google, Twitter, ...
- Authentification (OAuth via comptes réseaux sociaux)
- Paiements en ligne (Ex : <https://www.paymill.com>)
- Reconnaissance vocale

... A vous de créer les prochaines API du marché !

Exemple pour des entreprises (SNCF)



18'26

<https://www.youtube.com/watch?v=0FQ6w4CO5Nw>

Pour aller plus loin

<https://openclassrooms.com/courses/utilisez-des-api-rest-dans-vos-projets-web/qu-est-ce-qu-une-api>

Créer une API de A à Z en node :

<https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js>

Les réseau du WAGON (beaucoup de vidéos youtube) :

<https://www.lewagon.com/fr>