

# Application Web : City In Live

## Choix technologiques :

Notre single page application se résume à un panneau de visualisation d'informations diverses sur une ville.

Le visiteur sera amené à choisir une ville (via une barre de recherche ou une carte Google Maps), puis obtiendra des informations sur cette ville.

### Informations affichées et API REST utilisées :

- affichage et recherche sur une carte : **Google Maps**
- informations générales sur la ville : **Wikipedia**
- météo dans la ville : **OpenWeatherMap**
- tweets publiés dans la ville : **Twitter**
- photos prises dans la ville : **Instagram**

### Autres informations, si présentes dans la ville :

- trains en partance des gares de la ville : **SNCF**
- séances de cinéma : **International Showtimes API**

### Fonctionnalités :

- choix de la ville depuis une carte Google Maps ou une barre de recherche
- affichages d'informations :
  - + description sur la ville
  - + nombre d'habitants
  - + météo
  - + tweets et photos récentes
  - + horaires des trains en partance des gares ferroviaires
  - + prochaines séances de cinéma
- affichage réglable des modules de la page (taille, choix des modules, position, ...)

La plupart des API que nous utilisons ont besoin d'une clé d'authentification privée. De ce fait, nous ne pouvons pas envoyer cette clé aux clients pour qu'ils fassent les requêtes eux-mêmes. Nous allons donc avoir un serveur proposant API REST permettant au client d'y effectuer des requêtes pour avoir les informations pour chaque module sur une ville.

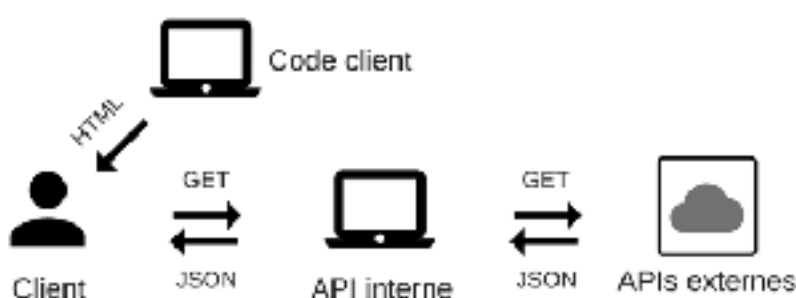
Le serveur ira effectuer les requêtes correspondantes aux API qui nous fournissent les données.

Coté serveur, nous allons utiliser *Sinatra*, un framework (*gem*) Ruby permettant de facilement ouvrir un serveur web répondant des données (JSON) à des URLs requises.

Nous souhaitons mettre en place une architecture MVC au niveau du client.

JavaScript étant le seul langage pouvant être exécuté coté client, nous allons donc utiliser *Angular* qui est un framework MVC en JavaScript.

Voici un schéma global du fonctionnement de l'application :



# **Application Web : City In Live**

Les serveurs fournissant l'API interne ainsi que le code client se synchronisent automatiquement sur les sources disponibles sur Github.

On utilisera un serveur personnel qui hébergera l'API interne et le code client.

## **Environnement de développement**

Chaque personne du groupe utilisera l'éditeur (intégré ou non) de son choix, tel qu'Atom ou Vim.

Angular sera installé avec npm, le gestionnaire de paquet de Node.js. Yeoman sera utilisé pour la mise en place du projet, et pour gérer les dépendances de Angular.