



Ce projet est cofinancé
par l'Union européenne



L'EUROPE S'ENGAGE
en région
Auvergne-Rhône-Alpes
avec le FSE



La Région
Auvergne-Rhône-Alpes

Utilisation d'APIs

API GUIDE

REQUEST URL FORMAT:

`http://www.com/<username>/<item ID>`

SERVER WILL RETURN AN XML
DOCUMENT WHICH CONTAINS:

- THE REQUESTED DATA
- DOCUMENTATION DESCRIBING HOW
THE DATA IS ORGANIZED SPATIALLY

API KEYS

TO OBTAIN API ACCESS, CONTACT THE
X.509-AUTHENTICATED SERVER AND
REQUEST AN ECDH-RSA TLS KEY...



IF YOU DO THINGS RIGHT, IT CAN TAKE
PEOPLE A WHILE TO REALIZE THAT YOUR
"API DOCUMENTATION" IS JUST INSTRUCTIONS
FOR HOW TO LOOK AT YOUR WEBSITE.

source: [xkcd](#)

ITÉRATION 1

Rappels/Bases sur les APIs

OBJECTIF

- Utiliser le protocol HTTP depuis python
- Se familiariser avec les APIs pour accéder à des données

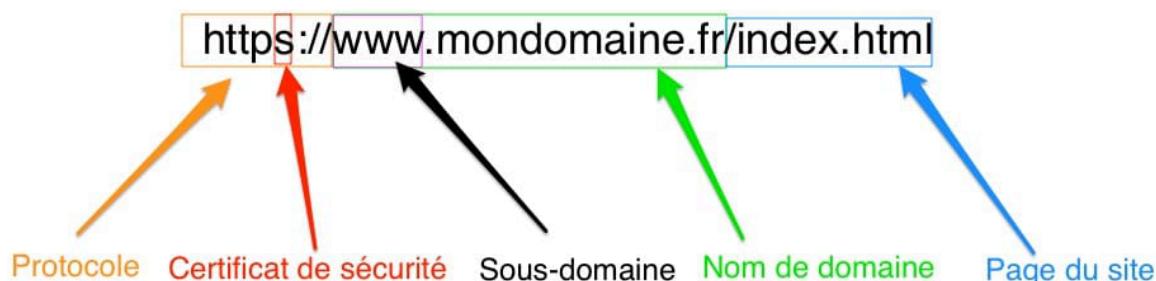
1.1 – Petit point sur le web et HTTP

25 minutes – Présentiel

Beaucoup de services web mettent à disposition des données au grand public. Le fonctionnement d'une API est le suivant: l'utilisateur soumet une requête en utilisant le protocole HTTP et le service web effectue les traitements nécessaires à la mise en forme des données pour renvoyer une réponse à l'utilisateur. Cette réponse peut contenir les données ou le résultat des traitements demandés.

Pour chaque adresse tapée dans la barre d'adresse du navigateur, celui-ci envoie (au moins) une requête au serveur lié à l'URL. Le serveur envoie (au moins) une réponse contenant la page web.

Une URL se décompose comme dans l'image ci-dessous:



Pour effectuer ce genre de requête depuis python on utilise une bibliothèque nommée requests, celle-ci est basée sur le protocole HTTP qui comporte différentes méthodes.

Les différentes méthodes sont:

- POST : ajoute des données (exemple: création d'un utilisateur)
- GET : lit des données en utilisant les paramètres passés (exemple: vue du profil d'un utilisateur)
- PUT/PATCH : met à jour des données (exemple: mise à jour du mot de passe)

- DELETE : supprime des données suivant les paramètres passés (exemple: suppression d'un compte utilisateur)

La méthode qui sera principalement utilisée dans cette partie est la méthode GET.

En python une bibliothèque utilisée pour interagir avec ces API se nomme `requests`.

Suite à l'import de cette bibliothèque son utilisation est assez simple;

`requests.get(url)` effectuera une requête HTTP GET sur l'url donnée en paramètre et

`requests.post(url)` effectuera une requête HTTP POST sur l'url donnée en paramètre.

Le plus souvent lorsque l'on utilise des APIs les données sont stockées dans le format JSON qui ressemble fortement à des dictionnaires python. Pour sauvegarder des dictionnaires python au format JSON, la bibliothèque couramment utilisée est inclue par défaut dans python et se nomme `json`.

1.2 – Lorem ipsum,... pour des images ...

3h – Présentiel

Consignes

- Installez la bibliothèque `requests` dans un environnement adéquate
- En utilisant cette bibliothèque, faites une requête get sur l'url de votre choix
 - Stockez le résultat de cette requête dans une variable
 - Explorez l'objet stocké dans cette variable
 - Que signifie le code "statut"?
 - Dans quel attribut trouve t-on des données?

Pour la suite nous allons utiliser les APIs de lorem picsum, un service permettant d'obtenir des images tirées au hasard. (<https://picsum.photos/>)

- Utilisez une requête get pour obtenir la liste des images au format `json`
 - Comment accède t-on aux données?
- Paramétrez votre requête afin d'obtenir les 50 premières images.
 - Créez une liste avec les urls images où le prénom de l'`author` contient la lettre 'r' (stockez la dans un fichier json)
 - Utiliser la méthode GET pour en télécharger dix de ces images.

RESSOURCES

- Tutoriel sur requests: https://rtavenar.github.io/poly_python/content/api.html
- Documentation de requests: <https://docs.python-requests.org/en/latest/>
- Site Lorem Picsum: <https://picsum.photos/>

LIVRABLES

- le fichier json contenant la liste des images



ITÉRATION 2

Utilisation d'API

Modalités

- Travail individuel en autonomie
- 1 jours en présentiel

Livrables

Le script permettant de récupérer les données demandées.

Les données collectées.

Objectifs

Cette itération a pour objectif d'utiliser les API et d'utiliser les documentations OpenAPI comme guide dans leurs utilisations.

Compétences

- Effectuer des requêtes sur des APIs pour collecter des données
- Lire et utiliser la documentation d'API pour créer des requêtes

1.1 – Introduction

5minutes – Présentiel

Dans la partie scraping, vous avez réussi à récupérer les données concernant les entreprises du département de l'isère, félicitations!

En regardant sur internet, vous trouvez une liste des APIs de l'état français, qui répertorie des données des entreprises en open data.

Dans la documentations vous trouvez la limite du nombre de requête des diff

RESSOURCES

- <https://entreprise.api.gouv.fr/catalogue> (APIs: Extrait RCS, Actes et bilans, Données insee en open data)

1.2 – Les communes des entreprises

2h – Présentiel

En parcourant les données, vous remarquez qu'il manque les codes postaux des différentes communes. Heureusement pour vous, le gouvernement français dispose d'une API permettant de retrouver ces codes et quelques autres informations.

ÉTAPES

- **Étape 1 : Crédation du fichier CSV des communes**

Pour commencer, créez un fichier CSV à partir des données scrapées contenant la liste des communes présentes dans le fichier CSV des entreprises. Ce fichier doit inclure les en-têtes suivants : **Nom de la Commune, Code Postal, Latitude, Longitude**. Utilisez les données déjà récupérées lors de l'étape de scraping pour remplir le champ **Nom de la Commune**.

- **Étape 2 : Récupération des coordonnées géographiques**

Utilisez l'API geo.api.gouv.fr pour récupérer les coordonnées géographiques et les codes postaux des communes. Pour chaque commune, faites une requête à l'API pour obtenir les informations manquantes. Ensuite, ajoutez les informations récupérées (code postal, latitude, longitude) dans le fichier CSV.

- **Étape 3 :Enregistrez un fichier csv**

Enregistrez le fichier au format csv et vérifiez que toutes les communes du fichier initial sont traitées.

RESSOURCES

- <https://geo.api.gouv.fr/>
- Petit rappel sur les apis: <https://blog.uprends.com/technology/the-anatomy-of-an-api-call/>
- Tutoriels sur requests:
<https://realpython.com/python-requests/>
<https://www.dataquest.io/blog/tutorial-an-introduction-to-python-requests-library/>

1.3 – Ajout des entreprises par localité

2h30 – Présentiel

OBJECTIF

L'objectif de cette partie est d'utiliser l'endpoint de recherche géographique (voir ressources) pour collecter les entreprises en fonction des latitudes et longitudes des différentes communes.

ÉTAPES

Étape 1 : Préparation des Données

Utilisez les coordonnées des communes obtenues à partir du fichier CSV des communes créé précédemment.

Étape 2 : Requêtes à l'API de Recherche Géographique

Pour chaque commune, configurez les paramètres de la requête avec les coordonnées de la commune, le rayon, l'activité principale et la section d'activité principale. Utilisez la bibliothèque `requests` pour faire une requête GET à l'endpoint de recherche géographique avec les paramètres configurés. Analysez la réponse de l'API pour extraire les informations sur les entreprises. (il faudra définir un rayon adapté à la majorité des communes)

/!\ Avant de lancer votre script vérifier les limites de requêtage de l'API /!

Étape 3 : Enregistrement des Données

Enregistrez les informations sur les entreprises extraites de l'API dans un fichier CSV.

Étape 4 : Comparer ces données avec celles scrapées précédemment

- Combien d'entreprises ont été scrapées?
- Combien ont été récupérées avec l'API?



- Faite le détail par commune et par secteur d'activité

LIVRABLES

- Un script fonctionnel capable de faire des requêtes à l'endpoint de recherche géographique et de collecter les entreprises en fonction des latitudes et longitudes des différentes communes.
- Un dataset complet contenant les informations sur les entreprises des secteurs ciblés dans les différentes communes de l'Isère.

RESSOURCES

- <https://www.data.gouv.fr/fr/dataservices/api-recherche-dentreprises/>

COMPÉTENCES ASSOCIÉES

- Utiliser une API
- Utiliser une documentation de type OpenAPI

1.4 – Collecte des données de l'INSEE

2h30 – Présentiel

Avant de faire des requêtes à l'API INSEE, vous devez suivre les étapes suivantes pour obtenir un token d'authentification :

- Créer un compte gratuit** : Rendez-vous sur le [portail de l'API INSEE](#) et créez un compte gratuit.
- Créez une application** : Une fois connecté, il est important de créer une application afin de pouvoir authentifier un programme sur les API de l'INSEE.
- Obtenir un token d'authentification** : Suivez les instructions pour obtenir un token d'authentification. Ce token sera nécessaire pour faire des requêtes à l'API INSEE.

Étape 1 : Préparation des Données

Utilisez le fichier CSV des entreprises créé précédemment pour obtenir les numéros SIREN des entreprises.

Étape 2 : Détermination des Champs à Récupérer

Avant de faire des requêtes pour toutes les entreprises, faites une première requête pour déterminer les champs qui seront stockés. Cela vous permettra de savoir quelles informations supplémentaires vous pouvez obtenir et comment structurer votre fichier CSV final.

Étape 3 : Requêtes à l'API INSEE

Pour chaque entreprise, configurez les paramètres de la requête avec le numéro SIREN et les champs spécifiques à récupérer. Utilisez la bibliothèque `requests` pour faire une requête GET à l'endpoint de l'API INSEE avec les paramètres configurés et le token d'authentification. Analysez la réponse de l'API pour extraire les informations supplémentaires sur les entreprises.

Étape 4 : Enregistrement des Données

Ajoutez les informations supplémentaires extraites de l'API INSEE à votre fichier CSV des entreprises. Enregistrez les modifications apportées au fichier CSV.

RESSOURCES

- <https://api.insee.fr/catalogue/site/themes/wso2/subthemes/insee/pages/help.jag>
- <https://portail-api.insee.fr/catalog/all>
- <https://github.com/swagger-api/swagger-ui>
- <https://en.wikipedia.org/wiki/GeoJSON>



Livrables

- les données collectées dans les différents fichiers