



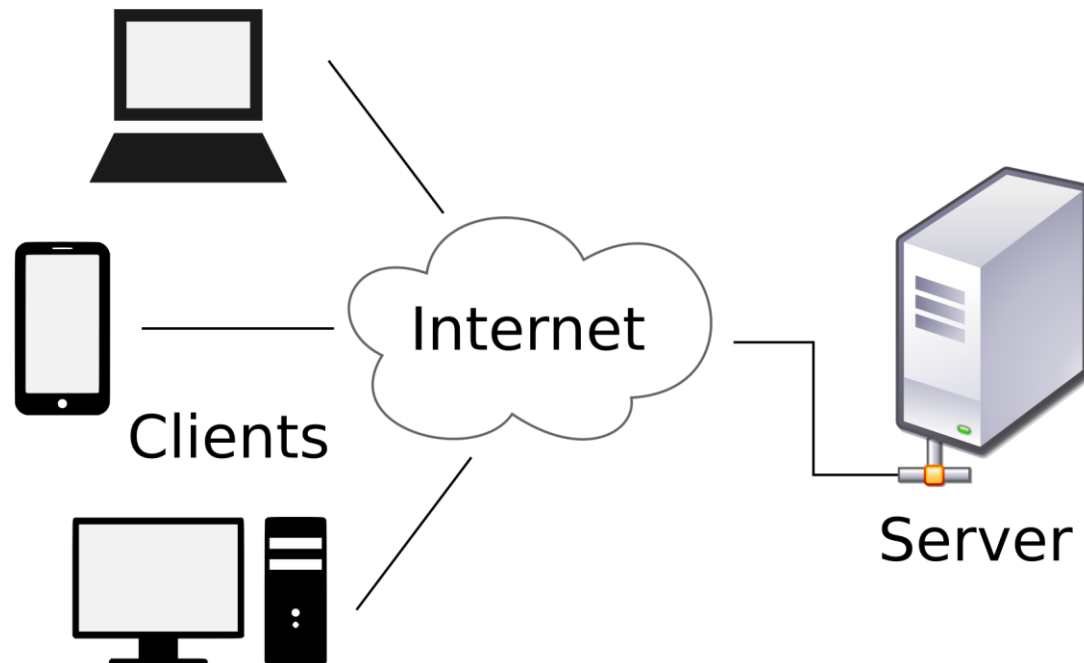
PYTHON ET OPEN DATA

API ET REQUÊTES HTTP



PRINCIPE GÉNÉRAL

➤ Le principe Client-Serveur



PRINCIPE GÉNÉRAL

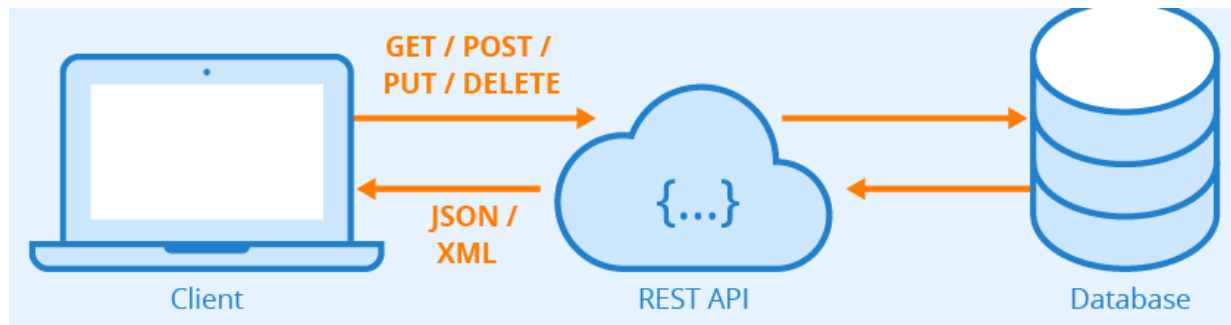
➤ Protocole HTTP

- Protocole d'échange d'informations sur le web
- Méthodes GET et POST (entre autres)
- Paramètres passés dans l'URL, ex :

➤ Résultat au format texte

- Données structurées : JSON, XML (HTML)

<https://data.explore.star.fr/api/records/1.0/search/?dataset=tco-metro-stations-etat-tr&rows=100>



[Source](#)

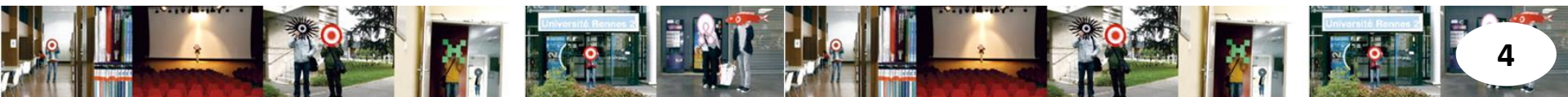
➤ Utilisation du module `requests`

- Permet de lire le résultat d'une requête HTTP GET

➤ Préparation de l'URL (cf. slide suivant)

➤ Analyse du dictionnaire

```
contenu_brut = requests.get(url)  
contenu_json = contenu_brut.json()
```



➤ Préparation de l'URL

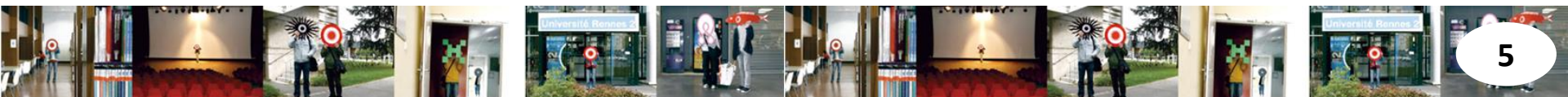
- Écrire l'URL de base du serveur
- Compléter les paramètres pour filtrer les résultats
- Parmi les deux versions ci-dessous, laquelle préférer ?

```
import requests
```

```
url =  
"http://truc.web/api?age=20"  
contenu = requests.get(url)
```

```
import requests
```

```
def filtre_20ans(donnees):  
    # [...]  
  
url = "http://truc.web/api"  
contenu = requests.get(url)  
post_filtre =  
    filtre_20ans(contenu)
```



➤ Préparation de l'URL

- Écrire l'URL de base du serveur
- Compléter les paramètres pour filtrer les résultats
- Parmi les deux versions ci-dessous, laquelle préférer ?

```
import requests

url =
"http://truc.web/api?age=20"
contenu = requests.get(url)
```

```
import requests

def filtre_20ans(donnees):
    # [...]

url = "http://truc.web/api"
contenu = requests.get(url)
post_filtre =
    filtre_20ans(contenu)
```

➤ Requête HTTP

- Préparer la requête (filtrage des résultats)
- Connaître la syntaxe
- Décoder en JSON

➤ Post-traitement

- Pas à pas
- Attention aux types

