

## Construisez vos projets web avec les APIs REST: le lexique

Tout au long du cours vous avez appris de nouveaux termes et découverts de nouveaux concepts. Ce lexique a pour but de centraliser et de vulgariser tous ces termes techniques afin que vous puissiez vous y référer si besoin, si une définition n'est pas claire ou a besoin d'être améliorée, n'hésitez pas à nous contacter !

### Liens utiles

- [Documentation Mozilla sur le protocole HTTP](#)
- [Documentation sur les codes de statut de réponse HTTP](#)
- [Discussion Stackoverflow sur les URI/URL/URN](#)
- [Différence URI/URL](#)
- [Liste de plus de 1400 API](#)
- [Liste de 18 API fun pour vos prochains projets](#)

### **LEXIQUE**

**Client** : un client est celui qui va utiliser l'API. Cela peut être une application, un navigateur ou un logiciel.

**Serveur** : un **serveur** est un ordinateur distant sur lequel tourne un site capable de récupérer des données depuis la base de données

**API** : API est un acronyme et signifie Application Programming Interface (ou interface de programmation d'application en français). Pour faire simple : c'est un moyen de communication entre deux logiciels que ce soit entre différents composants d'une application ou entre deux applications différentes.

# OPENCLASSROOMS

**API publique** : API utilisable par des d'autres personnes qu'elles soient sur votre application ou non afin d'obtenir les données d'une autre application sans autorisation stricte. En général elle permet d'obtenir des informations générales telles que la météo, une liste de films, de la musique (ex: Spotify)

**API privée** : API qui est *uniquement* accessible aux utilisateurs **autorisés** au sein de votre entreprise ou de votre application à utiliser l'API qui peut accéder à la base de données.

**REST** : REST signifie **R**epresentational **S**tate **T**ransfert. signifie **R**epresentational **S**tate **T**ransfer (ou transfert d'état de représentation en français), et constitue un ensemble de **normes**, ou de lignes directrices **architecturales** qui structurent la façon de communiquer les données entre votre application et le reste du monde, ou entre différents composants de votre application. Les APIs RESTful se basent sur le protocole **HTTP** pour transférer les informations — le même protocole sur lequel la communication web est fondée.

**SOAP** : SOAP est l'acronyme de **S**imple **O**bject **A**ccess **P**rotocol, ou protocole simple d'accès aux objets en français. Contrairement à REST, il est considéré comme un protocole, et non comme un style d'architecture. Les API SOAP étaient les API les plus courantes avant l'arrivée de REST.

**Ressource** : une ressource peut être tout type d'objet **nominal** (on lui attribue un nom) que vous pouvez utiliser pour représenter les données dans votre application. C'est cette donnée que l'utilisateur va vouloir manipuler, c'est le coeur d'une API.

**Collection** : ensemble qui regroupe les ressources.

**Authentication token** : Token d'authentification, permet à l'API d'identifier l'utilisateur et de savoir si il.elle a les droits adéquats pour obtenir les données demandées.



## URI, URL, Path

Toutes les URL sont des URI mais toutes les URI ne sont pas des URL. L'URI permet d'identifier une ressource tandis que l'URL permet de la localiser.

**URI** : URI est l'acronyme de Uniform Resource Identifier (identifiant de ressource uniforme en français). L'URI est le moyen d'identifier votre ressource.

**Path** : le chemin vers un fichier, ou une ressource. Autre nom donné à l'URI sur internet en fonction des articles.

# OPENCLASSROOMS

**URL** : permet de localiser la/les ressources, c'est ce qui vous permet d'accéder aux données d'une API.

## JSON et XML

**JSON** : format de donnée sous forme de clé : valeur

```
1 { "series": [  
2   { "titre": "Game Of Thrones",  
3     "realisateur": "Alan Taylor" },  
4   { "titre": "Peaky Blinders",  
5     "realisateur": "Otto Bathurst" }  
6 ] }
```

**XML** : format de donnée semblable au HTML. Se compose de balises ouvrantes et fermantes.

```
1 <series>  
2   <serie>  
3     <titre>Game Of Thrones</titre>  
4     <realisateur>Random</realisateur>  
5   </serie>  
6   <serie>  
7     <titre>Peaky Blinders</titre>  
8     <realisateur>Random</realisateur>  
9   </serie>  
10 </series>
```

## Requêtes et réponses

**Body** : pour une requête, il contient l'information que vous voulez apporter à l'API: création d'une nouvelle ressource ou modification d'une ressource existante.

Dans le cadre d'une réponse, le body contient les données que renvoie l'API (ressource, message d'erreur etc)

**Verbes HTTP** : les **verbes HTTP** correspondent à différents types d'actions que vous pouvez accomplir avec votre requête. Les plus courants que vous rencontrerez sont **GET** (obtenir), **PUT** (mettre), **POST** (publier), et **DELETE** (supprimer)

**Codes de réponses HTTP** : code renvoyé par l'API et qui donne au client une information sur le statut de la requête: réussite ou échec ou autre 🖱️

- 100+ ➡ Information
- 200+ ➡ Succès
- 300+ ➡ Redirection
- 400+ ➡ Erreur client
- 500+ ➡ Erreur serveur