

Ressources, URI et représentations

Les ressources

Une **ressource** est tout ce qui mérite d'avoir un nom et une adresse : un utilisateur, un dépôt, un livre, un emprunt, le profil de l'utilisateur connecté...

Une ressource n'est pas une table SQL. `/user` sur GitHub est une ressource (« moi »), alors qu'elle correspond à une ligne parmi d'autres de la table des utilisateurs. À l'inverse, un emprunt de livre peut réunir des données de plusieurs tables.

Les URI

Chaque ressource est identifiée par une **URI** (*Uniform Resource Identifier*). L'API de GitHub en donne de bons exemples :

Type	Exemple	Signification
Collection	<code>/users</code>	Tous les utilisateurs
Élément	<code>/users/thibaudfrichet</code>	L'utilisateur <code>thibaudfrichet</code>
Sous-collection	<code>/users/thibaudfrichet/repos</code>	Les dépôts de <code>thibaudfrichet</code>
Singleton	<code>/user</code>	L'utilisateur connecté (il n'y en a qu'un)

L'URI désigne **quoi**. C'est la méthode HTTP qui dit **quoi faire** avec. Les règles de nommage sont détaillées au chapitre 5.

GET	<code>/v1/livres/3</code>	□ lire le livre 3
DELETE	<code>/v1/livres/3</code>	□ supprimer le livre 3
GET	<code>/v1/deleteLivre?id=3</code>	□ un verbe dans l'URL

Les représentations

Le client ne reçoit jamais la ressource elle-même, mais une **représentation** de son état à un instant donné. Une même ressource peut avoir plusieurs représentations.

Le client indique ce qu'il veut avec le header `Accept`. C'est la **négociation de contenu**.
Exemple avec l'API d'une bibliothèque :

En JSON :

```
curl -H "Accept: application/json" https://api.biblio.example/v1/livres/3
```

```
{"id": 3, "titre": "Le Petit Prince", "auteur": "Antoine de Saint-Exupéry"}
```

En XML :

```
curl -H "Accept: application/xml" https://api.biblio.example/v1/livres/3
```

```
<livre id="3">
  <titre>Le Petit Prince</titre>
  <auteur>Antoine de Saint-Exupéry</auteur>
</livre>
```

Et si le format demandé n'existe pas, le serveur répond `406 Not Acceptable` :

```
curl -i -H "Accept: image/png" https://api.biblio.example/v1/livres/3
```

```
HTTP/1.1 406 Not Acceptable
```

Une représentation, c'est comme une photo : la même personne peut être en couleur ou en noir et blanc, en portrait ou en miniature. La personne, elle, ne change pas.

Dans l'autre sens, c'est le header `Content-Type` qui indique le format du body que le client **envoie**.