

Les API Web REST

Le REST c'est quoi ? REST est l'acronyme de « Representational State Transfer »

L'architecture REST a pris de plus en plus d'importance ces dernières années et on retrouve ce type d'architecture dans de nombreux services et réseaux sociaux. REST décrit les caractéristiques du Web qui en ont fait son succès.

Le REST est un style d'architecture logicielle, c'est l'architecture originelle du web et non un standard.

Il n'existe pas de spécifications de REST. Il faut d'abord déterminer comment fonctionne le style REST avant de vouloir concevoir des applications ou des services web.

REST concerne l'architecture globale d'un système. Il ne définit pas la manière de réaliser dans les détails.

En particulier, des services REST peuvent être réalisés dans n'importe quel langage compatible Web.

Règle n°1 : l'URI comme identifiant des ressources

Règle n°2 : les verbes HTTP comme identifiant des opérations

Règle n°3 : les réponses HTTP comme représentation des ressources

Règle n°4 : un paramètre comme jeton d'authentification

I° L'URI comme identifiant des ressources :

L'URI identifie une ressource informatique, l'URL est une des adresses permettant d'atteindre cette ressource.

Il est nécessaire de prendre en compte la hiérarchie des ressources et la sémantique des URL pour les éditer.

Quelques exemples de construction d'URL avec RESTful :

- Liste des livres : <http://mywebsite.com/books>
- Filtre et tri sur les livres : <http://mywebsite.com/books/filtre/policier/tri/asc>
- Affichage d'un livre : <http://mywebsite.com/books/87>
- Tous les commentaires sur un livre : <http://mywebsite.com/books/87/comments>
- Affichage d'un commentaire sur un livre : <http://mywebsite.com/books/87/comments/1568>

II° Les verbes HTTP comme identifiant des opérations :

La seconde règle d'une architecture REST est d'utiliser les verbes HTTP existants plutôt que d'inclure l'opération dans l'URI de la ressource. Ainsi, généralement pour une ressource, il y a 4 opérations possibles :

Créer (create) => POST

Afficher (read) => GET

Mettre à jour (update) => PUT

Supprimer (delete) => DELETE

HTTP intègre les verbes POST, GET, PUT et DELETE mais la plupart du temps, on se contente d'utiliser GET et POST.

- Créer un livre : POST <http://mywebsite.com/books>
- Afficher : GET <http://mywebsite.com/books/87>
- Mettre à jour : PUT <http://mywebsite.com/books/87>
- Supprimer : DELETE <http://mywebsite.com/books/87>

III° Les réponses HTTP comme représentation des ressources :

Il est important d'avoir à l'esprit que la réponse envoyée n'est pas une ressource, c'est la représentation d'une ressource. Ainsi, une ressource peut avoir plusieurs représentations dans des formats divers : HTML, XML, CSV, JSON, etc. On utilise de plus en plus la représentation Json pour sa compacité et sa facilité d'utilisation

IV° Un paramètre comme jeton d'authentification :

C'est un des sujets les plus souvent abordé quand on parle de REST : Comment authentifier une requête ?

La réponse est très simple et est massivement utilisée par des APIs renommées (Google, Yahoo, etc.) : le jeton d'authentification. Chaque requête est envoyée avec un jeton (token) passé en paramètre de la requête.

Ce jeton temporaire est obtenu en envoyant une première requête d'authentification puis en le combinant avec nos requêtes.