

Les API Rest

Les api REST c'est une méthode utilisé afin de partager des données avec tout un tas de client (des téléphones, consoles de jeux, GPS, TV etc ...). C'est aussi la possibilité de partager des données avec d'autres API Rest (des api d'amazon, google etc ...).

Ces api rest sont basé sur le protocole HTTP. Elle s'organise en suivant des règles stricts.

Resource

Les api rest possède des **Resources**, ces "resources" représente la donnée de votre application. Elles sont généralement présenté en formater en utilisant le format JSON.

En REST chaque resource représente de la données.

Exemple :

imaginons une api de vente de chaussure "shoes.me", nous pouvons imaginer les ressources suivante :

```
https://shoes.me/shoes
https://shoes.me/users
https://shoes.me/orders
etc ...
```

Les Collections et les Documents

Il existe 2 formes de ressources :

- La collection, représente une suite de documents, généralement nous utilisons une tableaux JSON pour la représenté : (`[]`)
- Le document, représente une données au singulier, généralement nous utilison un objet JSON pour la représenté : (`{}`)

Exemple :

```
https://shoes.me/shoes (Collection)
https://shoes.me/users (Collection)
https://shoes.me/orders (Collection)
https://shoes.me/me (Document)
```

Les API Rest sont dites « idempotante », c'est à dire à dire qu'une resource qui est un collection restera toujours une collection. Il en va de même pour les document, si une resource est un document alors il restera document toute sa vie.

Une collection ne peut pas se transformer en document, les documents ne peuvent pas se transformer en collection !

L'imbrication des ressources

Les API Rest on était conçu pour représenter n'importe quelle données. Pour cela il existe une notion "d'imbrication", des collections peuvent contenir des documents qui eux même peuvent contenir des collection etc ...

```
http://shoes.me/shoes/10 (Document)
http://shoes.me/shoes/10/comments (Collection)
http://shoes.me/shoes/10/comments/4 (Document)
http://shoes.me/shoes/10/comments/4/user (Document)
```

Attention, si les ressources s'imbrique dans un sens elle s'imbrique aussi dns l'autre
exemple :

```
http://shoes.me/connected/user (Document, Invalide ....)
http://shoes.me/connected (INvalide, ni à un document, ni à une collection)
```

Les actions

Les API Rest possède des actions symbolisé grâce aux méthode HTTP :

- **GET** : Obtenir, récupérer
- **POST** : Créer
- **DELETE** : Effacer, supprimer
- **PUT** : Modifier **l'intégralité de la resource** (d'une collection ou d'un document)
- **PATCH** : Modifier **une partie de la resource**

Exemple :

```
Je veux créer une nouvelle chaussure :
POST https://shoes.me/shoes
```

```
Je veux récupérer des chaussures
GET https://shoes.me/shoes
```

Pour qu'une API Rest soit valide, les méthodes HTTP sont interchangeable. Si je peux faire un GET alors, je peux faire un POST / PUT / PATCH et DELETE !

Les filtres

Il est possible de filtrer des ressources. Pour cela nous utilisons les **QueryString** :

Exemple :

```
Je veux récupérer les 10 chaussure trier par pix croissant
https://shoes.me/shoes?limit=10&orberBy=price&direction=ascending
```

```
Je veux récupérer les chaussure de marque "nike"
https://shoes.me/shoes?brand=nike
```

Les filtres ne change pas le forme d'une resource, si c'est collection ça restera une collection, si c'est un document ça restera un document :

```
https://shoes.me/shoes?id=10 (Collection)
```

Les query strings se place à la fin de le resource, nous ne pouvons pas imbriquer query string et resource :

```
https://shoes.me/shoes?limit=10&brand=nike/52/comments (Invalide, on imbrique des query dans des ressources)
```

Entraînez-vous !

Voici une série d'énoncé, pour chaque énoncé imaginé la forme de la resource, des query string et de la méthode HTTP (plusieurs solution sont possible).

Je veux récupérer toutes les chaussures

GET https://shoes.me/shoes (Valide, Collection)

Je veux créer une chaussure

POST https://shoes.me/shoes (Valide, Collection / Document)

Je veux modifier la chaussure avec l'identifiant n°10

PUT https://shoes.me/shoes/10 (Valide, Document)

PATCH https://shoes.me/shoes/10 (Valide, Document)

PATCH https://shoes.me/shoes?id=10 (Valide, Collection)

PUT https://shoes.me/shoes?id=10 (Valide, Dangereux, Collection)

Je veux récupérer les chaussures de la marque nike et ayant un prix maxium de 100 euros

GET https://shoes.me/shoes?brand=nike&orderBy=price<=100 (Valide)

GET https://shoes.me/shoes?brand=nike&price=le100 (Valide)

GET https://shoes.me/shoes?brand=nike&price=moins-100 (Valide)

GET https://shoes.me/shoes?brand=nike&price=<=100 (~ Valide)

GET https://shoes.me/shoes?brand=nike&maxPrice=100 (Valide)

Je veux récupérer l'utilisateur qui vend la chassure n°9

GET https://shoes.me/shoes/id=9?user (Pas valide)

GET https://shoes.me/users/9 (Pas valide)

GET https://shoes.me/shoes/9/user (Valide, Document)

GET https://shoes.me/USERS/shoes?id=9 (Non valide)

GET https://shoes.me/users?sellShoes=9 (Valide, Collection)

Je veux récupérer toutes les commandes de l'utilisateur avec l'id 53

GET http://shoes.me/commandes/user?id=53 (Pas valide)

GET https://shoes.me/user/53/orders (Pas valide)

GET https://shoes.me/users/53/orders (Valide)

GET https://shoes.me/orders?user=53 (Valide)

Je veux supprimer le commentaire avec l'id n°56 de la chaussure avec l'id n° 65

DELETE https://shoes.me/shoes/65/comments/56 (Valide)

DELETE http://shoes.me/shoes/65/comment?id=56 (Pas valide)
DELETE https://shoes.me/shoes/65/comment/56 (Ps valide)
DELETE http://shoes.me/shoes/65/comments?id=56 (Valide, Très dangereux ...)

Je veux modifier mon adresse email dans mes informations personnelles

PATCH http://shoes.me/user (Valide)
PATCH https://shoes.me/users/user?=me&email@ (Pas valide)
PATCH https://shoes.me/user/email (Valide, String)
PATCH https://shoes.me/users (Valide, Très dangereux ..)
PATCH https://shoes.me/users/username=irina (Pas valide)
PATCH http://shoes.me/personal_data/ (Valide)
PATCH http://shoes.me/me (Valide)
PATCH http://shoes.me/profil (Valide)
PATCH http://shoes.me/account (Valide)
PATCH https://shoes.me/users/53/

Je veux récupérer les utilisateur qui vendent des chaussure avec la marque "nike"

GET https://shoes.me/users/shoes?brand=nike (Pas valide)
GET https://shoes.me/users?shoes?brand=nike (Valide)
GET https://shoes.me/users?sellShoes&brand=nike (Valide)
GET https://shoes.me/users/sells?brand=nike (Pas valide)
GET https://shoes.me/shoes?brand=nike&user (Pas valide)
GET https://shoes.me/users?shoes.brand=nike (Valide)
GET https://shoes.me/brands/nike/users (Valide)