



Développement Backend – Node JS

INTERVENANT : THOMAS JAMME

PARTIE 3 : API REST

API REST

- Principes d'une API REST
- Bonnes pratiques

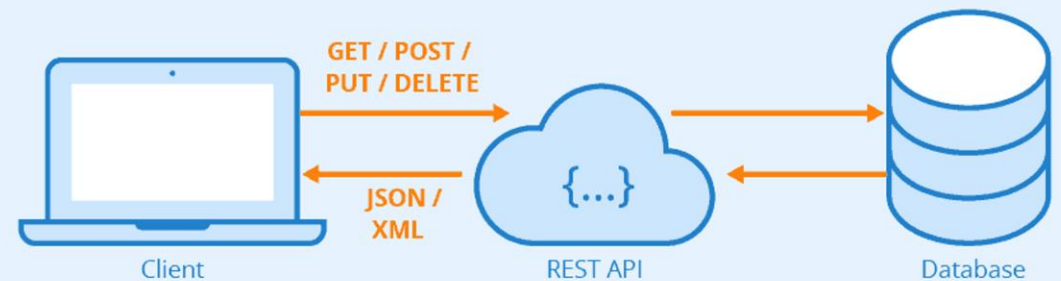


Principes d'une API REST

{ REST }

Principes d'une API REST

- Une API est un acronyme **Application Programming Interface**
- Elle définit quels appels ou requêtes peuvent être réalisés et comment les réaliser : le format des données à utiliser, la structure de la réponse, les conventions à respecter etc.
- Les APIs sont à la base de toutes les interactions entre différentes applications.



Principes d'une API REST

- Une API **REST** respecte les contraintes de l'architecture REST « Representational State Transfer ».
- REST est un ensemble de principes architecturaux. Il ne s'agit ni d'un protocole, ni d'une norme.
- Lorsqu'une requête est émise par le biais d'une API RESTful, celle-ci transfère une représentation de l'état de la ressource au demandeur.
- Le protocole HTTP est utilisé.
- Le format fourni peut être JSON, Text ou XML.
- Il existe énormément d'API ouverte que vous pouvez utiliser !

```
[
  {
    "description": "quarter",
    "mode": "REQUIRED",
    "name": "qtr",
    "type": "STRING"
  },
  {
    "description": "sales representative",
    "mode": "NULLABLE",
    "name": "rep",
    "type": "STRING"
  },
  {
    "description": "total sales",
    "mode": "NULLABLE",
    "name": "sales",
    "type": "INTEGER"
  }
]
```

JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<bilans xmlns="fr:inpi:odrn:cs:bilansSaisisXML" version="1.0">
  <bilan>
    <identite>
      <siren>086080520</siren>
      <date_cloture_exercice>20181231</date_cloture_exercice>
      <code_greffe>4502</code_greffe>
      <num_depot>4258</num_depot>
      <num_gestion>1960B00052</num_gestion>
      <code_activite>6420Z</code_activite>
      <date_cloture_exercice_n-1>20171231</date_cloture_exercice_n-1>
      <duree_exercice_n>12</duree_exercice_n>
      <duree_exercice_n-1>00</duree_exercice_n-1>
      <date_depot>20190617</date_depot>
      <code_motif>01</code_motif>
      <code_type_bilan>C</code_type_bilan>
      <code_devises>EUR</code_devises>
      <code_origine_devises>O</code_origine_devises>
      <code_confidentialite>0</code_confidentialite>
    </identite>
    <denomination>
      <CDATA[
        SAS ORTI
      ]>
    </denomination>
    <adresse>
      <CDATA[
        45140 SAINT JEAN DE LA RUELLE
      ]>
    </adresse>
  </bilan>
</bilans>
```

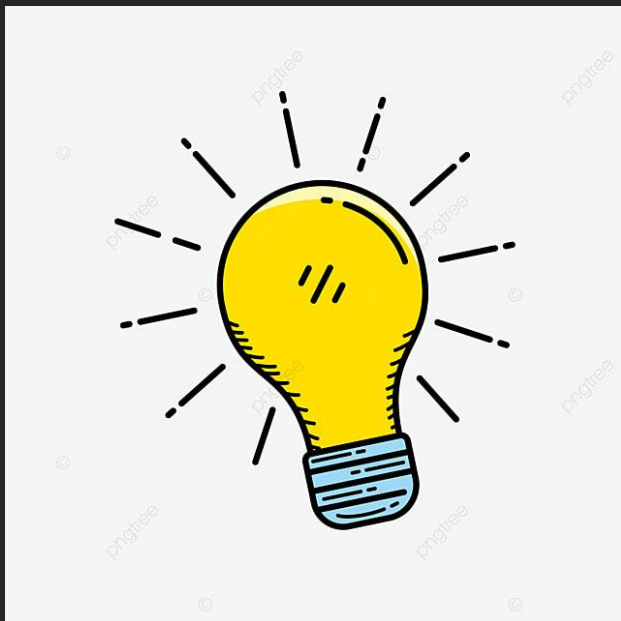
XML

Principes d'une API REST

Une API RESTful doit remplir les critères suivants :

- Une architecture **client-serveur** constituée d'un client, d'un serveur et d'une base de données.
- Des requêtes gérées en **HTTP**.
- Des communications client-serveur sans état. C'est-à-dire qu'aucune donnée n'est stockée dans le serveur.
- Les requêtes doivent être traitées de manière indépendantes.
- Une interface uniforme pour toutes les ressources manipulées.
- Une interface explicite permettant de comprendre quelle action est réalisée.

Bonnes pratiques pour développer une API REST



Bonnes pratiques pour développer une API REST

Bonne pratique N°1 : Utiliser les verbes HTTP

VERBE HTTP	UTILISATION
GET	Lire une donnée
POST	Créer une donnée
PUT	Modifier une donnée
DELETE	Supprimer une donnée

Bonnes pratiques pour développer une API REST

Bonne pratique N°2 : Utiliser des mots dans l'URL pour identifier le type de ressource

■ BIEN:

- /users pour récupérer les utilisateurs
- Utilisez le pluriel

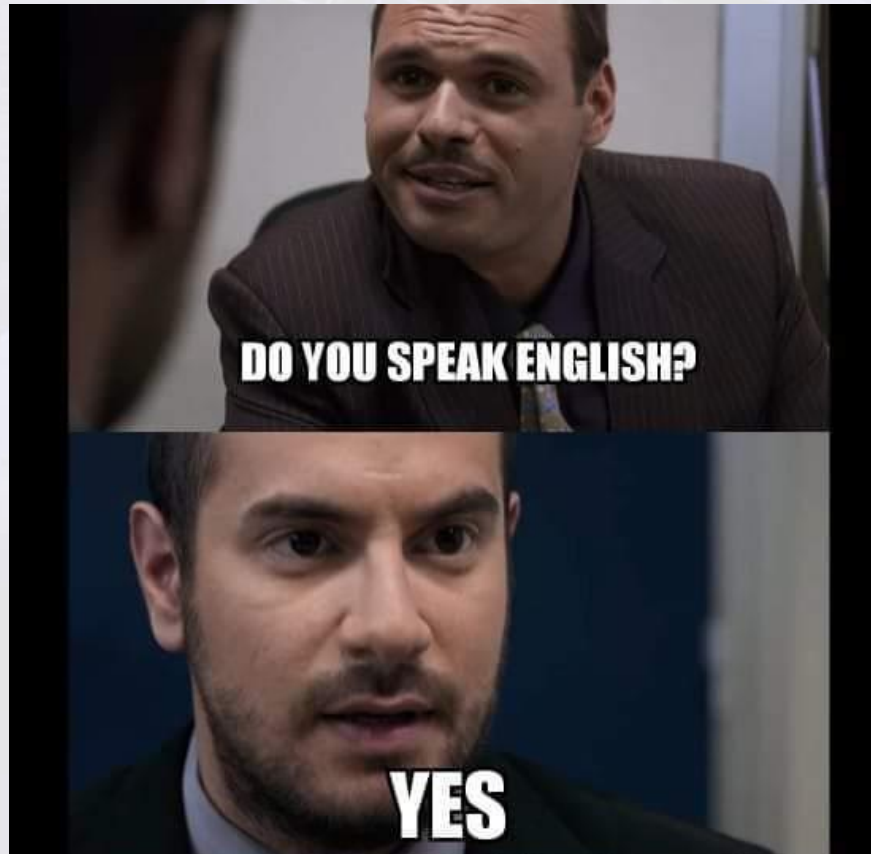
■ PAS BIEN :

- /getUsers pour récupérer les utilisateurs
- Ou /getAllUsers



Bonnes pratiques pour développer une API REST

Bonne pratique N°3 : Utiliser l'anglais



Bonnes pratiques pour développer une API REST

Bonne pratique N°4 : Utiliser des sous-ressources pour identifier un élément unique

Pour cela, il est recommandé d'utiliser un **path param** qui suit le premier chemin qui identifie le type de ressource.

URL	UTILISATION
/users/:userId	Récupérer les informations de l'utilisateur identifié par le userId passé en tant que pathParam

Bonnes pratiques pour développer une API REST

Bonne pratique N°5 : Utiliser des sous-ressources pour identifier une relation

Pour cela, il est recommandé d'ajouter **le nom de l'objet en relation** suite au pathParam qui identifie l'utilisateur. Voici un exemple où le modèle de données est conçu de manière à ce qu'un utilisateur possède des photos.

URL	UTILISATION
/users/:userId/photos	Récupérer les photos de l'utilisateur identifié par le userId passé en tant que pathParam

Bonnes pratiques pour développer une API REST

Bonne pratique N°6 : Utiliser correctement les codes de retour

Code de retour	UTILISATION
200 OK	Le serveur a traité la requête avec succès.
201 CREATED	Une nouvelle ressource a été créée ou modifiée.
204 No Content	Peut être utilisée en réponse à une requête DELETE effectuée avec succès.
206 Partial Content	En réponse à une requête demandant une réponse trop lourde pour être envoyée en une seule fois. De la pagination va être nécessaire pour récupérer l'ensemble des informations
304 Not Modified	Le client peut utiliser les données en cache car elles n'ont pas été modifiées depuis la date spécifiée.

Bonnes pratiques pour développer une API REST

Bonne pratique N°6 : Utiliser correctement les codes de retour

Code de retour	UTILISATION
400 Bad Request	La requête est invalide et ne peut pas être traitée par le serveur.
401 Unauthorized	La requête nécessite que le client soit identifié.
403 Forbidden	Le serveur a compris la requête mais l'utilisateur n'est pas autorisé à accéder à cette API.
404 Not Found	La ressource demandée n'existe pas.
500 Internal Server Error	Votre code ne devrait jamaïs renvoyer cette erreur. Cette erreur devrait être récupérée par votre code et traitée, pour ensuite renvoyer une réponse adéquate au client.

Bonnes pratiques pour développer une API REST

Bonne pratique N°7 : Utiliser des filtres, tris et pagination

Dans le cas où votre API renvoie des données filtrés, triées ou paginées, utilisez les **params** (et non les **path params**) pour communiquer ces contraintes.

Le nom de ces params doivent être explicites. L'ordre n'est pas important.

Exemple : Renvoyer une liste de 50 utilisateurs âgés entre 20 et 30 ans dans l'ordre alphabétique.

```
/users?ageMin=20?ageMax=30?order=asc?limit=50
```

Bonnes pratiques pour développer une API REST

Bonne pratique N°8 : Spécifiez le format de données

Il est important que le développeur qui consomme l'API connaisse le format de la réponse. Pour cela, le header **Content-type** permet d'indiquer le format répondu.

Ainsi, pour une réponse en JSON (majorité des cas), la réponse sera « **application/json** ».

Pour une réponse de type document PDF, la réponse sera « **application/pdf** ».

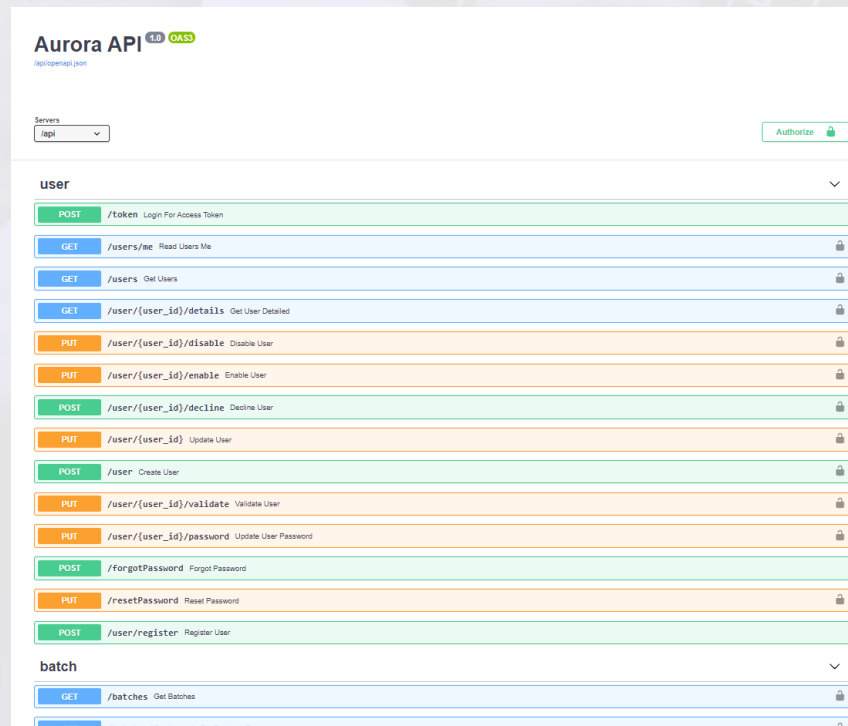
Il est important d'indiquer le charset dans le header. Sauf contre indication, utilisez « utf-8 ».

Bonnes pratiques pour développer une API REST

Bonne pratique N°9 : Documentez l'API

D'excellents outils tels que Swagger permettent de mettre en place une documentation de l'API.

La documentation est généralement accessible depuis une page web et permet de tester l'API.



Bonnes pratiques pour développer une API REST

Bonne pratique N°10 : Pensez à l'application qui consommera l'API

- Gardez une certaine logique dans votre conception.
- Evitez par exemple de renvoyer trop peu d'éléments si l'application est capable d'en afficher beaucoup.
- N'envoyer pas trop d'informations ou pas assez.
- Organisez l'API de manière à ce qu'elle ne soit pas appelée trop souvent par l'application
- Pensez à la bande passante que vous allez utiliser.
- Pensez également à la planète : le coût énergétique de deux recherches sur Google équivaldrait à faire chauffer une tasse d'eau jusqu'à l'ébullition !



TP : Mise en place des bonnes pratiques sur notre API

Mettez désormais en place les bonnes pratiques sur l'API !

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits		
Produit	Récupérer tous les produits d'une catégorie		
Produit	Créer un produit		
Produit	Modifier un produit		
Produit	Supprimer un produit		
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories	GET	/categories
Catégorie	Créer une catégorie	POST	/categories
Catégorie	Modifier une catégorie	PUT	/categories/:idCat
Catégorie	Supprimer une catégorie	DELETE	/categories/:idCat

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie		
Produit	Créer un produit		
Produit	Modifier un produit		
Produit	Supprimer un produit		
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit		
Produit	Modifier un produit		
Produit	Supprimer un produit		
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit		
Produit	Supprimer un produit		
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit		
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories		
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories	GET	/categories
Catégorie	Créer une catégorie		
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories	GET	/categories
Catégorie	Créer une catégorie	POST	/categories
Catégorie	Modifier une catégorie		
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories	GET	/categories
Catégorie	Créer une catégorie	POST	/categories
Catégorie	Modifier une catégorie	PUT	/categories/:idCat
Catégorie	Supprimer une catégorie		

TP : Mise en place des bonnes pratiques sur notre API

Objet final	Action	Verbe	Route
Produit	Récupérer tous les produits	GET	/produits
Produit	Récupérer tous les produits d'une catégorie	GET	/categories/:idCat/produits
Produit	Créer un produit	POST	/produits
Produit	Modifier un produit	PUT	/produits/:idProd
Produit	Supprimer un produit	DELETE	/produits/:idProd
Catégorie	Récupérer toutes les catégories	GET	/categories
Catégorie	Créer une catégorie	POST	/categories
Catégorie	Modifier une catégorie	PUT	/categories/:idCat
Catégorie	Supprimer une catégorie	DELETE	/categories/:idCat



Des questions ?