

Les API REST

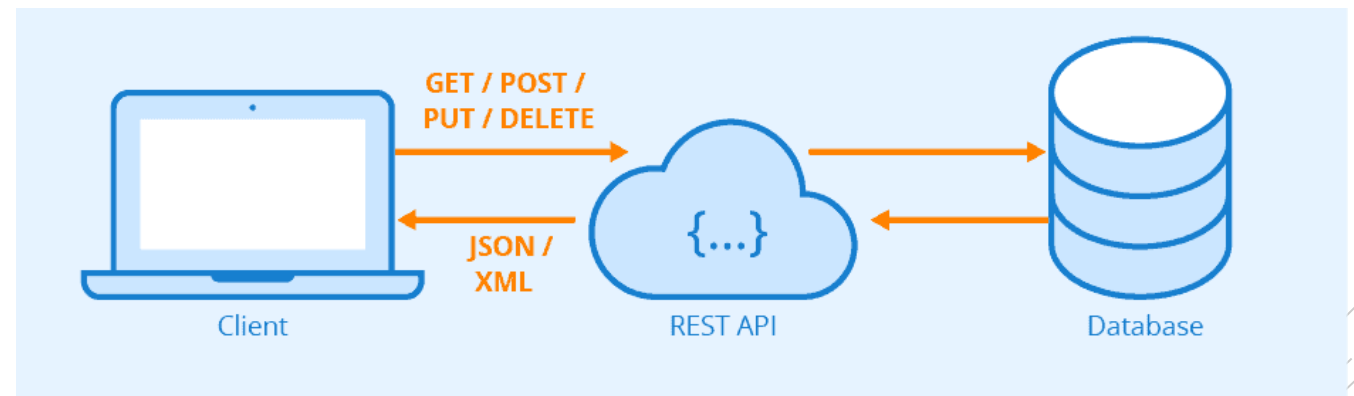
P.PISZYNA - Janvier 2021

Sommaire

- **Présentation**
 - Qu'est-ce qu'une API REST ?
 - Rappel sur les méthodes HTTP
 - Rappel sur la structure JSON
- **Exemple de services en Open Data**
 - Openweathermap
 - Wikipedia
- **Cas pratique avec une table**
 - Structure de la table « etudiant »
 - Privilèges
 - Spécifications de l'API à développer
- **Codage du Back-end**
 - Rôle des fichiers
 - Tests avec Postman
- **Codage du Front-End**
 - Rappel sur l'objet XMLHttpRequest

Présentation

- Qu'est-ce qu'une API REST ?
 - Interface entre un client (web ou application) et un serveur de bases de données
 - Elle spécifie le format des requêtes (HTTP) et la structure des réponses (JSON, XML, ...)
 - *REpresentational State Transfer* : Le serveur reste dans le même état quelle que soit la requête
 - Très répandues dans l'open-data



Présentation

- Méthodes HTTP utilisées pour des architectures REST :
 - GET : Accède à une ressource
 - HEAD : Accède aux métadonnées d'une ressource
 - POST : Ajoute une ressource
 - PUT : Met à jour une ressource **complète** en la remplaçant par une nouvelle version (99% des cas).
 - PATCH : Met à jour une partie d'une ressource en envoyant un différentiel (une sorte de 'git diff')
 - DELETE : Supprime une ressource

Présentation

- Structure d'un fichier JSON :
 - {...} : les **accolades** définissent un objet.
 - "prenom":"Alex" : Les **guillemets** (double-quotes) et les **double-points** définissent un couple clé/valeur.
 - [...] : Les **crochets** définissent un tableau (ou array en anglais).
 - Les **virgules** permettent de séparer les membres d'un tableau ou, comme ici, d'un objet :

```
{  
  "nom": "TERIEUR",  
  "prenom": "Alex",  
  "user": "TA",  
  "pwd": "123456",  
  "section": "SN2"  
}
```

Exemples de services en open-data

- **Openweathermap :**

- Requête :

- api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}

- Réponse : {

- ```
"coord": {
 "lon": -122.08,
 "lat": 37.39
},
"weather": [
 {
 "id": 800,
 "main": "Clear",
 "description": "clear sky",
 "icon": "01d"
 }
],
"base": "stations",
"main": {
 "temp": 282.55,
 "feels_like": 281.86,
 "temp_min": 280.37,
 "temp_max": 284.26,
 "pressure": 1023,
 "humidity": 100
},
```

<https://openweathermap.org/current>

# Exemples de services en open-data

- **Wikipedia :**

- **Requête :**

- <https://fr.wikipedia.org/w/api.php?action=query&titles=Martinique&prop=extracts&exchars=500&explaintext&utf8&format=json>

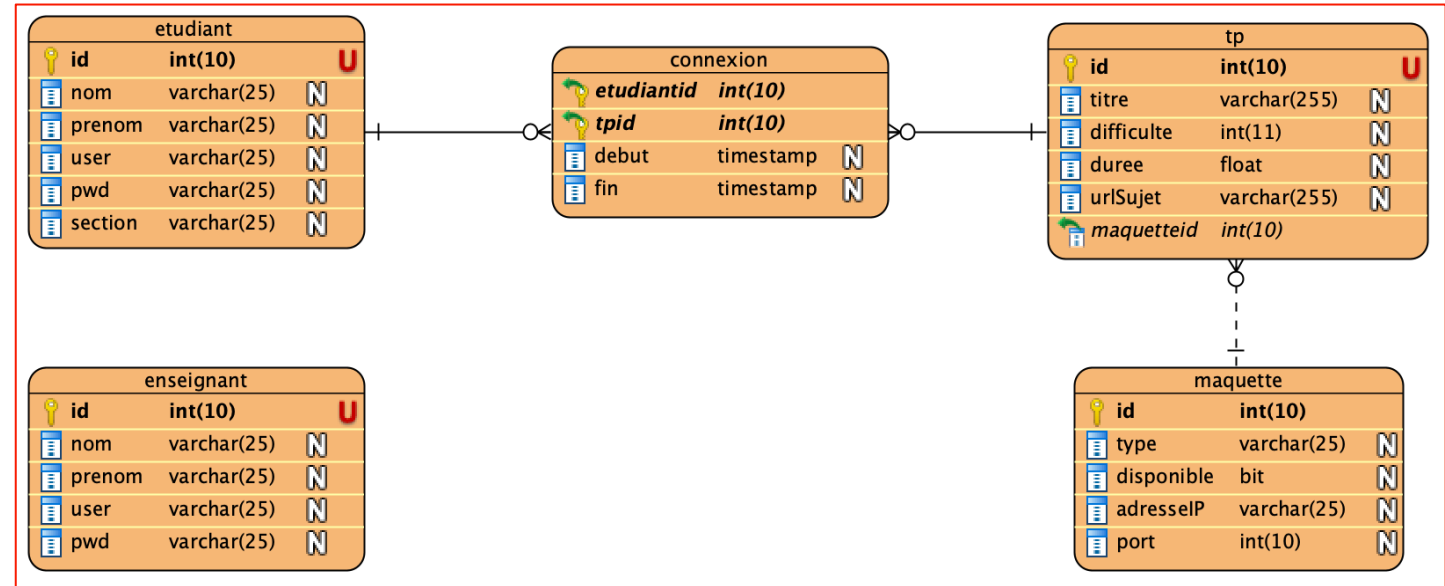
- **Réponse :**

```
{
 batchcomplete: "",
 - query: {
 - pages: {
 - 3075409: {
 pageid: 3075409,
 ns: 0,
 title: "Martinique",
 extract: "La Martinique, en créole martiniquais Matinik ou Ma
située dans les Caraïbes et plus précisément dans l'archipel
972) ; elle est administrée dans le cadre d'une collectivité
l'Union..."
 }
 }
 }
}
```

# Cas pratique

- La table « etudiant » :

- Structure :



- Contenu :

| id | nom       | prenom  | user | pwd         | section |
|----|-----------|---------|------|-------------|---------|
| 1  | MELCHIOR  | Elodie  | ME   | mangobidule | SN2     |
| 2  | BELHUMEUR | Daryl   | BD   | azerty      | SN2     |
| 3  | BERTILLE  | Luckael | BL   | R1isTheOne  | SN2     |
| 7  | EUGENE    | Jilane  | EJ   | 123456      | SN2     |
| 15 | TERIEUR   | Alex    | TA   | 123456      | SN2     |



# Cas pratique

- La table « etudiant » :
  - Privilèges : On va créer un utilisateur ayant pour nom et mot de passe le nom de la base.
  - Exemple :
    - BD : **STPD**
    - Utilisateur : **stpd**
    - Mot de passe : **stpd**

The screenshot shows the phpMyAdmin interface for the 'STPD' database. The left sidebar contains navigation options like 'connexion', 'enseignant', 'etudiant', 'maquette', and 'tp'. The main area displays a table of users with access to the 'STPD' database. The table has columns for 'Utilisateur', 'Client', 'Type', and 'Privileges'. The user 'stpd' is highlighted with a red box, showing it is a local user with 'spécifique à cette base de données' type and 'ALL PRIVILEGES'.

| Utilisateur | Client    | Type                               | Privileges                                  |
|-------------|-----------|------------------------------------|---------------------------------------------|
| access      | %         | global                             | SELECT, INSERT, UPDATE, DELETE, CREATE, DRO |
| etudiant    | %         | global                             | SELECT, INSERT                              |
| etudiant    | DS107     | global                             | SELECT, INSERT                              |
| etudiant    | localhost | global                             | SELECT, INSERT                              |
| mwb         | %         | global                             | ALL PRIVILEGES                              |
| root        | 127.0.0.1 | global                             | ALL PRIVILEGES                              |
| root        | DS107     | global                             | ALL PRIVILEGES                              |
| root        | localhost | global                             | ALL PRIVILEGES                              |
| stpd        | localhost | spécifique à cette base de données | ALL PRIVILEGES                              |

# Cas pratique

- La table « etudiant » :

- Spécifications de l'API :

| Route                           | Méthode | Type | Description                              |
|---------------------------------|---------|------|------------------------------------------|
| http://{url}/api/etudiants      | GET     | JSON | Récupérer tous les étudiants             |
| http://{url}/api/etudiants/{id} | GET     | JSON | Récupérer les données d'un seul étudiant |
| http://{url}/api/etudiants      | POST    | JSON | Insérer un nouvel étudiant               |
| http://{url}/api/etudiants/{id} | PUT     | JSON | Mettre à jour un étudiant                |
| http://{url}/api/etudiants/{id} | DELETE  | JSON | Supprimer un étudiant                    |

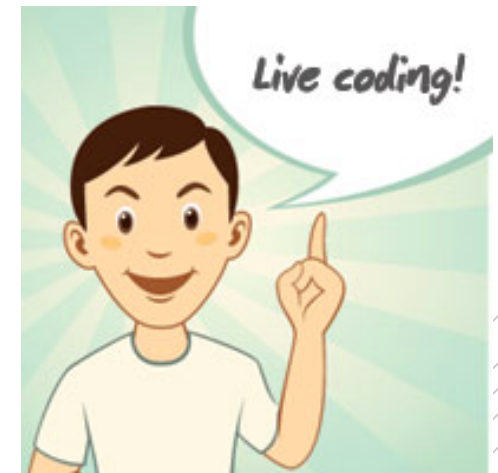
- {url} représente l'adresse IP et le nom de la base
    - Par exemple 10.10.13.214/stpd

← → ↻ 🏠 🌐 10.10.13.214/stpd/api/etudiant/3

```
id: "3",
nom: "BERTILLE",
prenom: "Luckael",
section: "SN2"
```

## Codage du back-end

- La connexion à la base de données et les requêtes SQL seront faites avec les fonctions PHP mysqli.
- Rôle des fichiers :
  - `index.php` : empêche de lister le répertoire
  - `db_connect.php` : connexion à la base de données
  - `etudiants.php` : réalise les fonctions du CRUD
  - `.htaccess` : permet le routage à partir de l'URL de requête
  - ...



## Codage du front-end

- La page web affichant le contenu de la table étudiants sera codée en HTML5 (html/css/javascript)
- L'objet XMLHttpRequest permet :
  - L'envoi d'une requête HTTP
  - La récupération de la réponse en XML ou en JSON
- La fonction JSON.parse() :
  - Transforme un objet JSON en un tableau Javascript
- Le modèle Document (DOM) permet :
  - De pointer un élément dans la page web (id de balise)
  - D'injecter du code HTML à l'intérieur de cette balise