

API REST

Création, sécurisation et documentation d'une API en respectant les standards REST

Alexandre Gaillot

Plan du cours

1. Qu'est-ce qu'API et REST
2. Nos premières fonctions et le format JSON
3. Organiser son code pour plus de clarté
4. Server Request Method
5. Sécurisation de l'API
6. Documentation de l'API

Qu'est-ce qu'API REST?

API

Application programming Interface

(ensemble normalisé de [classes](#), de [méthodes](#), de [fonctions](#) et de [constantes](#) qui sert de [façade](#) par laquelle un logiciel offre des services à d'autres logiciels)

REST

Representational State Transfer

(style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web)

Les 5 contraintes de l'architecture REST

(+1 facultative)

1. Client/Server
2. Stateless
3. Cacheable
4. Layered system (système multi-couches)
5. Uniform interface
6. Code on demand

Nos premières fonctions et le format json

<https://www.postman.com/downloads/>

CRUD

- Récupérer les catégories de la BDD
- Ajouter une nouvelle catégorie
- Supprimer une catégorie
- Modifier une catégorie

```
/**
 * Récupère les données de la BDD à envoyer au front
 *
 * @param string $statement
 * @param boolean $one
 */
public function query(string $statement, bool $one=false){
    try {
        $query = $this->pdo->query($statement, PDO::FETCH_OBJ);
        // if ($one) {
        //     $data = $query->fetch();
        // } else{
        //     $data = $query->fetchAll();
        // }
        $message = "data ok";
        $this->result($message, $data);
    } catch (\Throwable $th) {

        $message = "Une erreur s'est produite lors
        de la récupération des informations";
        $this->result($message, array(), false);
    }
}
```


Organiser son code pour plus de
clarté

Server Request Method

Server Request Method

- `$_SERVER['Request_method ']` => Récupère l'information de la méthode d'appel de la page: GET, POST, PUT, DELETE, ...
- Création d'un switch pour définir le traitement des informations en fonction de la méthode

Sécurisation de l'api

Sécurisation de l'api

- Plusieurs méthodes possibles: Faire sa propre sécurisation ou passer par un composant de sécurisation tel que JWT
- Générer un token pour chaque utilisateur: `uniqid(md5())`
- Envoyer le token sur chaque appel de page pour vérifier ses droits

Documentation de l'api

Documentation de l'api

- Plusieurs méthodes possible:
 - Faire sa propre documentation / Risque de non mise à jour
 - Passer par un composant de documentation ex: Swagger
- Découverte de Swagger très demandé sur le marché

<https://github.com/zircote/swagger-php>

composer require zircote/swagger-php