

TP07 - Web Services - API REST

Pré-requis : installer Postman



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

Exercice 1 - Utiliser et tester des APIs

Travail à faire :

- Utilisation de l'**API Geo.api.gouv.fr** (<https://geo.api.gouv.fr/decoupage-administratif>)
 - Lorsqu'on exécute la requête suivante : **GET https://geo.api.gouv.fr/communes**
Quel est le code de la réponse ? Quel résultat obtient-on (*body*) ? Quelles informations obtient-on sur chaque commune ?
 - Quelle requête faut-il exécuter pour obtenir la liste des communes du département de l'Isère ?
 - Quelle requête faut-il exécuter pour obtenir la liste des communes correspondant au code postal 38200 ?
 - Modifier la requête de manière à obtenir de nouvelles informations : *superficie, numéro et nom du département, code et nom de la région*
- Utilisation de l'**API Github** (<https://docs.github.com/en/rest/reference>)
 - Quel résultat obtient-on lorsqu'on exécute la requête suivante : **GET https://api.github.com/users** ?
 - Quelle requête faut-il exécuter pour obtenir les informations relatives à votre compte Github ?
 - Quelle requête faut-il exécuter pour obtenir la liste de tous vos dépôts (*repositories*) publics ? Les classer de la date de création la plus récente à la plus ancienne
 - Lister vos dépôts publics et privés en exécutant la requête suivante : **GET https://api.github.com/user/repos**
Quel est le code de réponse ? Pourquoi la requête échoue ?
 - Obtenir un *token* d'authentification sur le portail Github
 - Cliquer sur le lien : <https://github.com/settings/tokens>
 - Dans le menu, choisir *Personal access tokens*
 - Cliquer sur *Generate new token* : saisir un intitulé (note), sélectionner les autorisations
 - ☒ **repo** Full control of private repositories
 - ☒ **delete_repo** Delete repositories
 - puis cliquer sur *Generate Token*
 - Récupérer (par Copier/coller) le token généré **Token pour API Github — delete_repo, repo**
 - Ajouter le *token* dans l'entête de votre requête GET (*Authorization*)

Exécuter à nouveau la requête GET

- Créer un nouveau *repository* en exécutant la requête suivante : **POST https://api.github.com/user/repos**
Dans le *body*, placer les données du *repository* à créer, au *format JSON* :

```
{
  "name": "NEW-REPO",
  "description": "Nouveau repo créé avec l'API Github !"
}
```

... renseigner le token d'authentification

Exécuter la requête. Quel est le code de réponse ?

Vérifier que le repository a bien été crée (en exécutant une requête GET)

- h. Quelle requête faut-il exécuter pour modifier les informations du repository crée précédemment ?
On souhaite modifier la description, que le repository soit privé et qu'il permette l'activation des issues.
 Quel est le code de la réponse ? Vérifier que les informations du dépôt ont bien été modifiées
- i. Quelle requête faut-il exécuter pour supprimer le repository crée précédemment ?
 Quel est le code de la réponse ? Vérifier que le dépôt a bien été supprimé

Exercice 2 : Consommer une API depuis une application cliente

Ressources : Vous disposez de 3 applications clientes exploitant l'**API Geo.api.gouv.fr**

- une application Windows Forms codée en C# : **WinTestAPI.sln**
- une page Web codée en PHP : **CallWSenPHP.php**
- une application mobile Android codée en Java : **MyApplicationWS2**

Travail à faire :

1. Tester et analyser chaque application cliente afin de comprendre son fonctionnement et l'exploitation de l'API
2. Modifier chacune des applications afin d'afficher pour chaque commune les informations suivantes : *superficie, numéro et nom du département, code et nom de la région*

Exercice 3 : Créer une application cliente utilisant une API

Travail à faire : Créer une application cliente (de bureau, web ou mobile) utilisant l'**API Github**

Fonctionnalités attendues :

- lister vos repositories
- afficher les détails d'un repository sélectionné
- créer un repository
- supprimer un repository

Exercice 4 : Créer une API REST en PHP

L'API **APIBiblio** permet de gérer une **ressource Livre**. Plusieurs opérations sont possibles sur cette ressource.

Liste des EndPoints :

Route (URI)	Méthode	Type	Description
localhost/APIBiblio/livres	GET	JSON (réponse)	Obtenir tous les livres
localhost/APIBiblio/livres/{isbn}	GET	JSON (réponse)	Obtenir les données d'un seul livre (dont l'ISBN est spécifié)
localhost/APIBiblio/livres	POST	JSON (données envoyées et réponse)	Ajouter un nouveau livre
localhost/APIBiblio/livres/{isbn}	PUT	JSON (données envoyées et réponse)	Modifier les données d'un seul livre (dont l'ISBN est spécifié)
localhost/APIBiblio/livres/{isbn}	DELETE	JSON (réponse)	Supprimer un livre (dont l'ISBN est spécifié)

Les données caractérisant les livres sont stockées dans une base de données MySQL. L'API est en cours de développement, elle est codée en PHP.

Ressources : Vous disposez des éléments suivants :

- **Script biblio.sql** : script SQL de création de la base de données **Biblio** (ce script crée la table **Livre** et insère un jeu de données)

➡ Structure de la table Livre

```
LIVRE(isbnLivre, titreLivre, anneeLivre, auteurLivre, pagesLivre)
clé primaire : isbnLivre
```

➡ Contenu de la table Livre après exécution du script

isbnLivre	titreLivre	anneeLivre	auteurLivre	pagesLivre
2266	La fille de papier	2013	Guillaume Musso	608
4208	Harry Potter et l'enfant maudit	2016	J.K Rowling	352
7333	Petit pays	2016	Gael Faye	224

- **Dossier APIBiblio.sql** : ce dossier contient les différents scripts PHP de l'API

➡ Script fonctionsDAL.inc : script PHP contenant la fonction de connexion à la base de données **Biblio** et toutes les fonctions réalisant les opérations CRUD sur la table **Livre**

```
// connexion à la base de données Biblio
function connexionBase() {

// fonction qui retourne tous les livres
function DAL_getAllLivres() {

// fonction qui retourne le livre dont l'isbn est spécifié
function DAL_getLivreByISBN($isbn) {

// fonction qui permet de modifier un livre
function DAL_updateLivre($isbn, $titre, $annee, $auteur, $pages) {

// fonction qui permet d'insérer un nouveau livre
function DAL_insertLivre($isbn, $titre, $annee, $auteur, $pages) {

// fonction qui permet de supprimer un livre
function DAL_deleteLivre($isbn) {
```

➡ Script livres.php : script PHP qui contrôle (orchestre) les différentes requêtes HTTP

Travail à faire :

Pré-requis :

- Créer la base de données **Biblio** (exécution du script biblio.sql)
- Copier le dossier **APIBiblio** dans le répertoire de publication de votre serveur Web

1. Requête GET localhost/APIBiblio/livres

- A l'aide de *Postman*, tester le bon fonctionnement de la requête : **GET localhost/APIBiblio/livres**
Vérifier le code (*status*) et le contenu (*body*) de la réponse
- Analyser le code des scripts `livres.php` et `fonctionsDAL.inc` afin de comprendre comment on parvient à ce résultat
- Analyser le code afin de comprendre la gestion des exceptions

2. Requête POST localhost/APIBiblio/livres

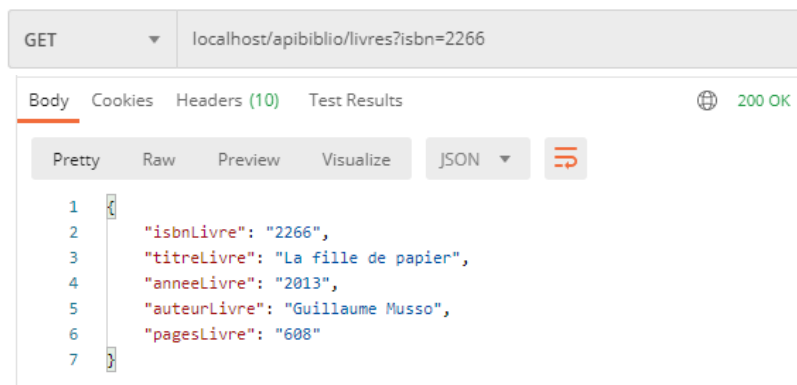
- Analyser le code des scripts `livres.php` et `fonctionsDAL.inc` afin de comprendre comment la requête POST est traitée et relever quels sont les différents cas d'erreurs pris en compte
- A l'aide de *Postman*, tester de manière exhaustive le bon fonctionnement de la requête : **POST localhost/APIBiblio/livres**

3. Requête POST localhost/APIBiblio/livres/{isbn}

- Compléter le code de la fonction `getLivre($isbn)` du script `livres.php` afin de gérer la requête GET qui permet d'obtenir les données du livre dont l'isbn est spécifié.

Deux cas sont à envisager :

➡ Cas 1 : l'isbn existe



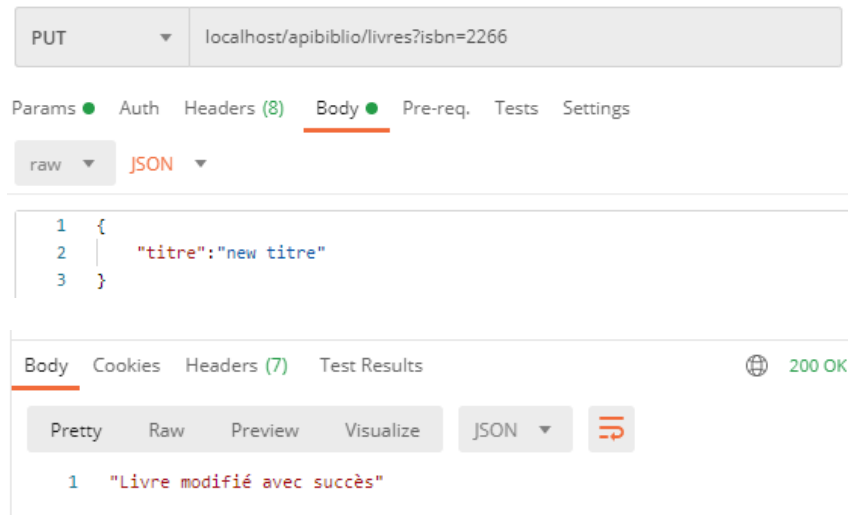
➡ Cas 2 : l'isbn n'existe pas



4. Requête PUT localhost/APIBiblio/livres/{isbn}

- Compléter le code de la fonction `DAL_updateLivre(...)` du script `fonctionsDAL.inc` permettant de mettre à jour dans la table *Livre* les données du livre dont l'isbn est spécifié
- Compléter le code de la fonction `putLivre($isbn)` du script `livres.php` afin de gérer la requête PUT qui permet de mettre à jour les données du livre dont l'isbn est spécifié. Les données modifiées sont envoyées dans le corps de la requête au format JSON

Exemple :



5. Requête DELETE localhost/APIBiblio/livres/{isbn}

- Compléter le code de la fonction `DAL_deleteLivre(...)` du script `fonctionsDAL.inc` permettant de supprimer dans la table *Livre* le livre dont l'isbn est spécifié
- Compléter le code de la fonction `deleteLivre($isbn)` du script `livres.php` afin de gérer la requête DELETE qui permet de supprimer le livre dont l'isbn est spécifié.

Exemple :

