



BTS SNIR

Lycée Louis Armand
94 - Nogent sur Marne



Module Web 8

Initiation API REST



Sommaire

Les compétences visées.....	2
Mise en route du serveur web.....	5
Importation de la base de données.....	5
Importation du site web.....	6
Création des répertoire.....	6
Analyse du fichier « model.php ».....	6
Modification des fichiers.....	7
Test de la base de données.....	7
Méthodes de requête.....	8
Demande du client.....	9
Récupération de la méthode.....	10
Récupération de l'URL.....	10
Test de l'API.....	10
Extraction des données JSON.....	13
Insertion dans la base de données.....	13
Insertion de l'utilisateur.....	13
Insertion d'un nouveau vol.....	13
Insertion des états de vol.....	14
Création d'un fichier configuration.....	15
Création d'un fichier « .htaccess ».....	15

Les compétences visées

</>	Catégorie	Je suis capable de :	
	BdD	Créer une base de données	
		Créer des clés primaires	
		Créer des clés étrangères	
		Importer des données	
		Visualiser les données dans phpmyadmin	
	JSON	Comprendre le format JSON	
		Structurer des données au format JSON	
		Convertir des données JSON en tableau associatif et vice-versa	
	API	Récupérer les méthodes d'une requête HTTP	
		Récupérer les données dans le corps d'une requête HTTP	
		Récupérer la cible de la requête	
		Faire un traitement différent en fonction de la méthode d'envoi	
	PHP	Inclure des fichiers	
		Créer et utiliser des fonctions	
		Se connecter à une base de données	
		Exécuter des requêtes SQL	
		Traiter la réponse d'une requête SQL	

TD 1 - Base de données ou BdD

La base de données suivante a été construite en première année et vous sera fournie si besoin.

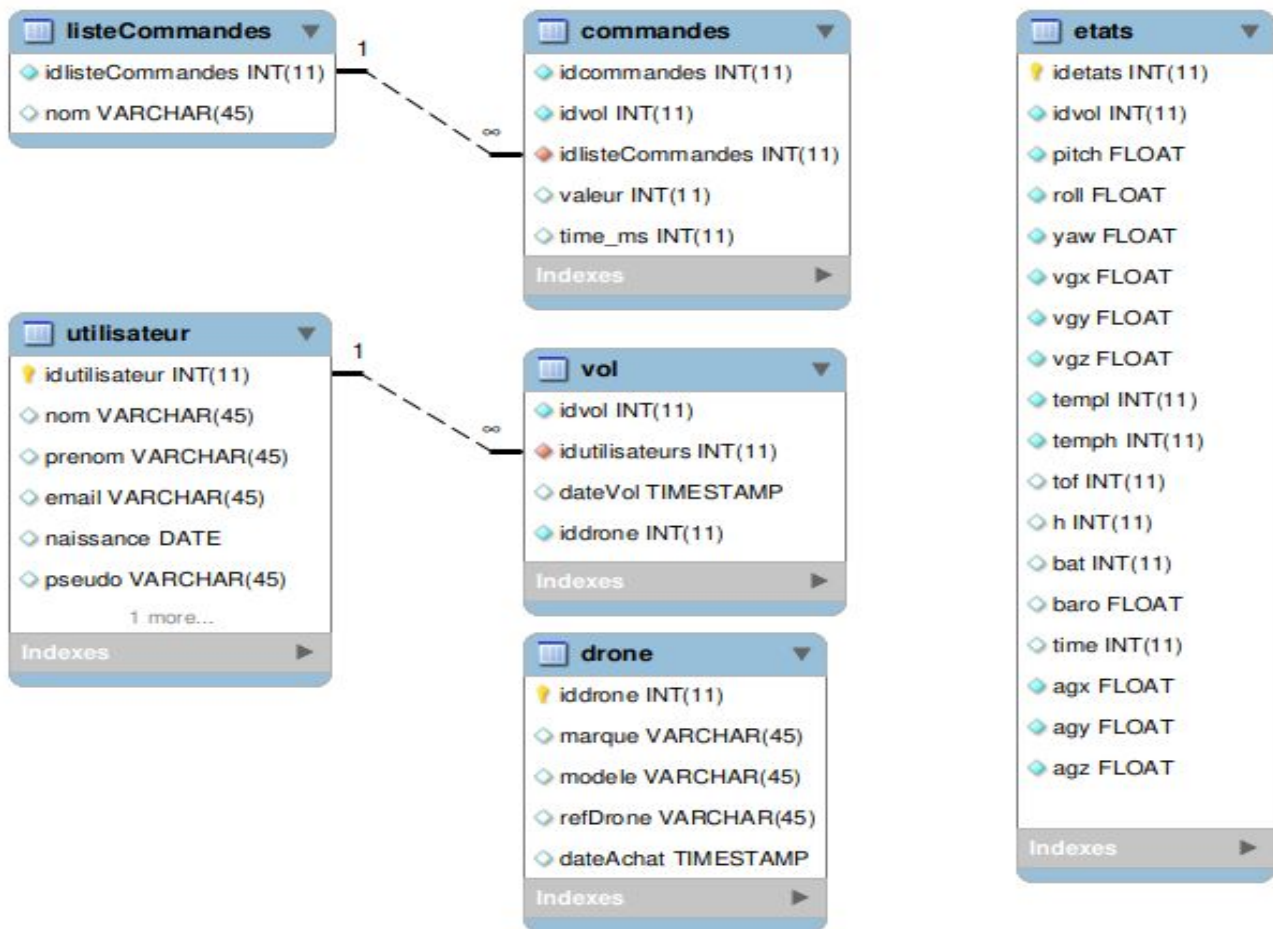


Table 1: Base de données drone Tello

? Qu'est ce qu'une base de données ?

? Comment sont stockées les informations d'une BdD

? Lister les différentes tables utilisées dans ce module.

? Lister les différentes clés primaires

? Lister les différentes clés étrangères

? Les liaisons entre les différentes tables sont appelées « cardinalités ». Que signifie ce terme ?

</> Compléter la BdD précédente en y intégrant les relations clé primaire / étrangère ainsi que leurs cardinalités.

Défi 1 - Test du site web

Mise en route du serveur web

</> Allumer et connecter vous, via les applications FilleZila ou smarTTY, à votre mini-PC sous linux en utilisant les identifiants créés durant la première année.

</> Noter dans la case suivante les informations suivantes liées à votre machine linux :

Adresse MAC :

Adresse IP :

Utilisateur / Mot de passe :

Utilisateur root / Mot de passe root :



Quels sont les deux services web nécessaire pour héberger un site web ?

</> Ouvrez un terminal distant en utilisant smartTTY

</> Vérifier que ces deux services sont démarré sur votre machine linux en utilisant la commande suivante dans un terminal distant :

```
service --status-all
```

</> Il faut également vérifier la version PHP installée

```
php --version
```

Avant de commencer à utiliser la machine linux, il faut faire des mises à jour.

</> Faire une mise à jour de votre machine linux en utilisant les lignes de commandes suivantes :

```
sudo apt update && apt upgrade && sudo apt autoremove
```

Importation de la base de données

</> Ouvrir phpmyadmin et créer une base de données MW8_votre_nom



Les liaisons entre les tables imposent que les données soient importées dans un ordre précis. Par exemple il n'est pas possible d'insérer des données de vol si la table utilisateurs ne contient aucun enregistrements. Quelle est donc l'ordre d'insertion des enregistrements des tables dans la BdD ?

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

</> Importer les fichiers avec l'extension « .sql » fournit avec votre dossier de travailler correspondant aux différentes tables de la BdD

Importation du site web

Création des répertoire

</> Créer le dossier MW8 sur votre compte linux.

</> Ajouter les fichiers fourni avec le sujet

Analyse du fichier « model.php »

? Quelle fonction permet d'initier une connexion à la base de données ?

? Quels sont les paramètres à modifier dans la fonction précédente ?

? A quoi sert la fonction « preparerRequete() ?

? A quoi sert la fonction « exectuterRequetePrepare() ?

? A quoi sert la fonction « extraireDonneesRequetePrepare() ?

? A quoi sert la fonction « recupererLeDernierIdInserer() ?

? A quoi sert la fonction « fermerCursor() ?



A quoi sert la fonction «`executerRequeteCurl()`» ?

Modification des fichiers

- </> Compléter le fichier « `model.php` » avec des lignes de commentaires expliquant l'utilité de chaque fonction présente.
- </> Modifier la fonction « `ConnectBD()` » dans le fichier « `model.php` » pour l'adapter à votre serveur.
- </> Modifier la variable `$url` dans la fonction « `ExecuterRequeteCURL(.....)` » afin de renvoyer vers votre API REST c'est-à-dire vers le fichier « `rest.php` ».

Test de la base de données

- </> Créer le fichier « `rest.php` »
- </> Ajouter le code permettant d'exécuter une requête et d'en afficher le résultat en suivant les étapes :

1. Inclure le fichier « `model.php` »
2. Faire appel à la fonction `ConnectDB()`
3. Ecrire une requête de test permettant de sélectionner tous les utilisateurs de la Bdd
4. Exécuter la requête précédente en utilisant les fonctions du fichier « `model.php` »
5. Afficher la réponse en utilisant la fonction `print_r()`

- </> Tester votre code en affichant dans un navigateur la page web `rest.php`

TD 2 - Requête HTTP

Méthodes de requête

Les méthodes de requête HTTP permettent de définir l'action que l'on souhaite réaliser sur la ressource indiquée. Il en existe plusieurs mais nous allons nous focaliser sur les suivantes GET, POST, PUT, DELETE.

? Définir dans quel cas chaque méthode est utilisée :

GET :

POST :

PUT

DELETE

Il est donc important de connaître la méthode utilisée avant de faire la requête SQL.

? Quelle variable super global utilisée en PHP permet d'extraire des informations sur la connexion ?

? Quelle clé est associée à la méthode d'envoi d'une requête HTTP du tableau associatif \$_SERVER ?

? Quelle clé est associée au chemin URL envoyé par le client et contenu dans le tableau associatif \$_SERVER ?

```
Soit la requête HTTP suivante :  
GET http://localhost/MW8/rest.php/utilisateurs/hagot
```

```
Soit le fichier rest.php :  
$req_type=$_SERVER['REQUEST_METHOD'];  
$req_path=$_SERVER['PATH_INFO'];
```

? Quel est le contenu de la variable \$req_type ?

? Quel est le contenu de la variable \$req_path ?

Demande du client

La demande du client web est contenu dans la variable `$req_path`. Il faut alors adapter la requête SQL à exécuter.

Exemple 1 : demander à la BdD tous les vols effectué par l'utilisateur iervese

```
GET http://localhost/MW8/rest.php/vols/iervese
```

Exemple 2 : demander à la BdD toutes les informations de l'utilisateur georges

```
GET http://localhost/MW8/rest.php/utilisateurs/georges
```

? Quelle est la valeur de `$req_path` dans chaque exemple ?

? Pour extraire les différents termes de la variable `$req_path`, nous allons convertir la chaîne de caractères en un tableau en utilisant la fonction « `explode(string $separator, string $string)` ». Donner la ligne de code et stocker le résultat dans la variable `$req_data`.

? Quelle case du tableau `$req_data` contient le terme « vols » ou « utilisateurs » ?

Exemple 1 :

Exemple 2 :

? Quelle case du tableau `$req_data` contient le nom des utilisateurs concerné ?

Exemple 1 :

Exemple 2 :

Défi 2 - Mise en place de l'API REST

Récupération de la méthode

</> Compléter le fichier rest.php en y intégrant le code permettant de récupérer la méthode HTTP envoyée par le client.

</> Ajouter le code suivant :

```
Si la méthode vaut GET alors j'affiche le mot « get » avec la fonction « echo »  
Si la méthode vaut POST alors j'affiche le mot «post» avec la fonction « echo »
```

Récupération de l'URL

</> Compléter le fichier rest.php en y intégrant le code permettant de récupérer la demande du client contenue dans l'URL : création de la variable \$req_path et \$req_data.

</> Ajouter le code suivant dans chaque méthode HTTP :

```
Si dans l'URL il y a le terme « vols » alors j'affiche le mot «vols» avec la fonction « echo »  
    Puis s'il contient également un nom d'utilisateur avec j'affiche le nom de l'utilisateur  
  
Si dans l'URL il y a le terme «utilisateurs» alors j'affiche le mot «utilisateurs» avec la fonction  
« echo »  
    Puis s'il contient également un nom d'utilisateur avec j'affiche le nom de l'utilisateur
```

Test de l'API

L'extension RESTClient d'un navigateur permet d'envoyer une requête HTTP et d'afficher la réponse.

</> Ouvrir l'extension RESTClient de votre navigateur, à installer si besoin, et modifier les paramètres de « Requête » et « URL ». Vérifier que la réponse correspond bien à votre code.

TD 3 - Format JSON ou XML

RESTful ne revendique ni n'impose un format d'échange entre client et serveur. Cependant deux grandes familles sont largement utilisées : le JSON et le XML.

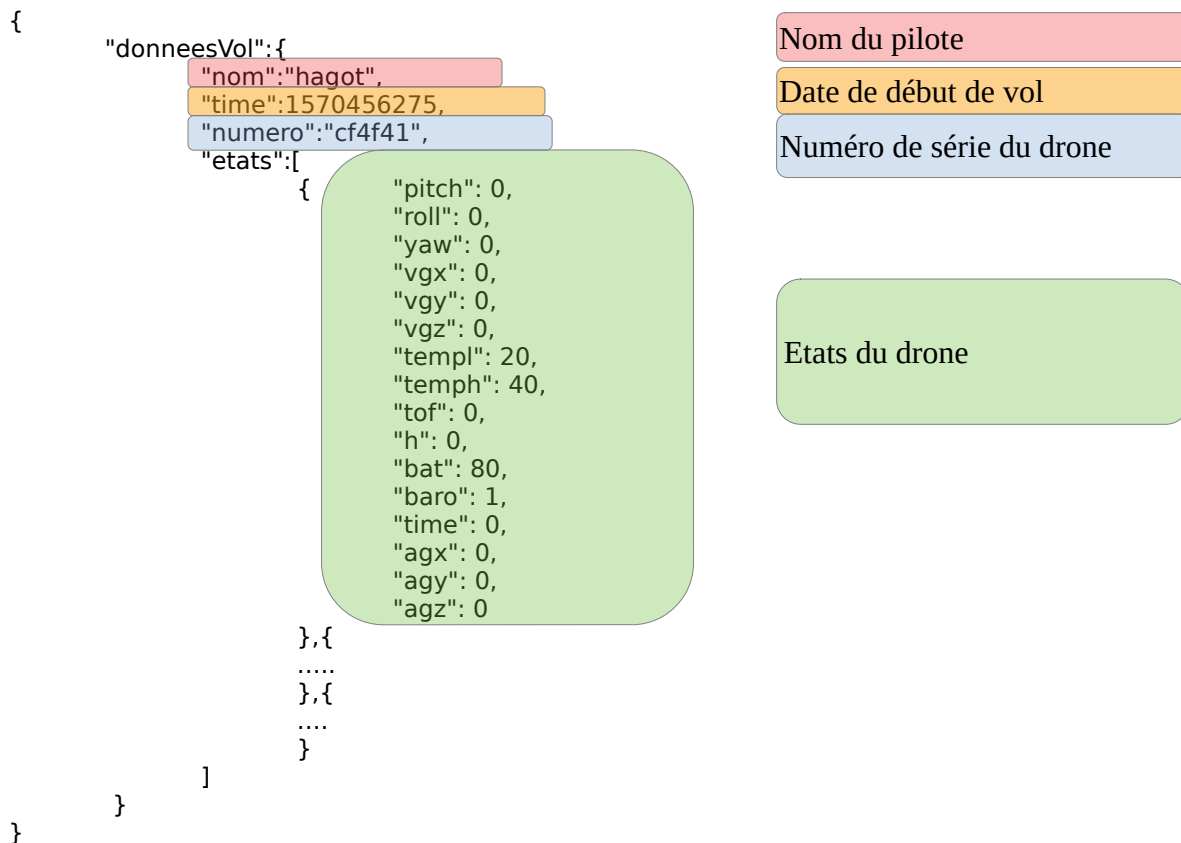
? Que signifie les acronymes JSON et XML ?

JSON :

XML :

? Donner 2 avantages du JSON par rapport au XML :

Dans le module suivant, nous utiliserons le format JSON suivant qui est envoyé par le serveur Tello et que nous devons exploiter dans notre API REST :



Pour exploiter les données de vol reçues au format JSON dans le langage PHP, nous devons les convertir la chaîne de caractère en un tableau associatif avec la fonction « `json_decode(string $json, true)` ».

? Donner le code permettant de convertir les chaîne de caractère json stockées dans la variable « `$donneesVolJSON` » en un tableau associatif « `$donneesVolAssoc` »

? Quel est la valeur du code suivant : `$donneesVolAssoc ['donneesVol'] ['nom']`

- ? Quel ligne de code permet de récupérer le numéro de série du drone ?
- ? Quel ligne de code permet de récupérer les états de vol ? Stocker la valeur dans une variable \$etatsVol
- ? En utilisant la variable précédente, donner la syntaxe permettant de récupérer la première série de données de vol
- ? Ecrire une boucle for permettant de récupérer tous les états de vol et de les stocker dans une variables \$donneesEtatsVol

Défi 3 - Exploitation des données JSON

Après un vol de drone, les données sont envoyées au serveur au format JSON.

Extraction des données JSON

Pour récupérer les données envoyées dans le corps d'une requête HTTP, il faut utiliser le code

```
file_get_contents("php://input")
```

- </> Compléter le fichier rest.php afin de récupérer les données JSON et les insérer dans la variable « \$donneesVolJSON ».
- </> Extraire les données JSON dans une variables « \$donneesVolAssoc ».

Insertion dans la base de données

Insertion de l'utilisateur

L'insertion de l'utilisateur se fera uniquement si son nom n'est pas déjà présent dans la BdD

- ? Quelle requête permet de sélectionner un utilisateur connaissant son nom ?

- ? Quelle requête permet d'insérer un utilisateur connaissant son nom ?

Ici nous allons détailler la procédure à suivre afin d'insérer un nouvelle utilisateur :

- 1) Sélectionner l'idutilisateurs connaissant son nom.
- 2) Si le résultat n'est pas vide alors on stockera son id dans un \$_COOKIE['idutilisateur']
- 3) Sinon on insérera un nouvelle utilisateur et on récupéra son id que l'on stockera dans un cookie

- </> Compléter le fichier PHP afin de créer un nouvel utilisateur s'il n'est pas présent dans la BdD
- </> Tester votre fichier en utilisant « RESTClient » en y ajoutant dans le body les données JSON. Vous devez voir l'utilisateur s'ajouter dans la BdD ou alors un cookie idutilisateur apparaître.

Insertion d'un nouveau vol

Pour chaque données reçues, nous allons créer un nouveau vol si le time envoyé par le JSON est différent de la date dans le BdD pour un utilisateur donné sinon nous ferons une mise à jour des données.

Attention pour la date, il faut transformer le timestamp en une valeur sous la forme « AAAA-MM-DD HH:mm:ss ». Pour cela vous pouvez utiliser la fonction date de PHP. On l'utilisera de la façon suivante :

\$date=date('Y-m-d H:i:s',\$time) ou \$time contient la valeur au format timestamp.

- ? Quelle requête permet de sélectionner un vol connaissant le nom de l'utilisateur et la date ?

- ? Quelle requête permet d'insérer un vol connaissant l'idutilisateur et la date ?

- </> Compléter le fichier PHP afin de créer un nouveau vol s'il n'est pas déjà présent dans la BdD
- </> Tester le bon fonctionnement avec « RESTClient »

Insertion des états de vol

? Quelle requête permet d'insérer état de vol connaissant idvol ?

- </> Compléter le fichier PHP afin d'insérer un nouvel « état » de vol dans la base de données.
- </> Compléter par une boucle for dans le cas de plusieurs états de vol envoyés par la requête HTTP
- </> Tester le bon fonctionnement avec « RESTClient »

Défi 4 - Pour aller plus loin

Création d'un fichier configuration

</> Créer un fichier config « app.ini »

</> Créer les variables permettant de configurer la connexion au serveur BdD

```
host = localhost
name = dbname
user = db_username
pass = db_password
type = mysql
```

</> Utiliser `$ini=parse_ini_file(string $filename)` pour lire le fichier « app.ini »

</> Utiliser le tableau `$ini` dans le fichier « model.php » pour la configuration de la base de données

Création d'un fichier « .htaccess »

Les requêtes HTTP pour l'instant sont sous la forme `POST /rest.php/utilisateurs/iervese`. Pour avoir une structure plus élégante, nous aimerions que cela deviennent `POST /api/utilisateurs/iervese` sans modifier le nom des fichiers.

Pour cela nous allons utiliser les redirections dans le fichier htaccess

</> Créer un fichier vide et l'enregistrer sous « .htaccess »

</> En vous aidant du net, modifier le fichier afin de faire des redirection `/api` en `/rest.php`

</> Tester vos requêtes HTTP en remplaçant dans l'URL `/rest.php` par `/api` et constater le résultat.