



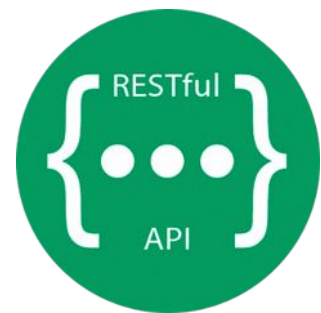
# BTS SNIR

Lycée Louis Armand  
94 - Nogent sur Marne



## Module Web 9

### Création de graphes JS



## Les compétences visées

| </>        | Catégorie               | Je suis capable de :                               |  |
|------------|-------------------------|--|--|
| <b>PHP</b> | MVC                     | Organisation des fichiers                          |  |
|            |                         | Gestion des redirections                           |  |
|            |                         | Traitement des informations                        |  |
|            | REST                    | Gérer les requêtes HTTP GET                        |  |
|            |                         | Communiquer avec le modèle MVC                     |  |
|            | Code                    | Générer des tableaux HTML en PHP                   |  |
|            |                         | Exploitation de la barre URL                       |  |
|            |                         | Créer des formulaires HTML à partir de données PHP |  |
| <b>JS</b>  | Amchart ou graphe en JS | Inclure les librairies                             |  |
|            |                         | Créer un graphe                                    |  |
|            |                         | Générer des axes X et Y                            |  |
|            |                         | Ajouter une légende aux axes                       |  |
|            |                         | Insérer des données JSON                           |  |

# PARTIE 1

## MISE EN PLACE DU SITE

### TD 1 - Comprendre l'architecture MVC

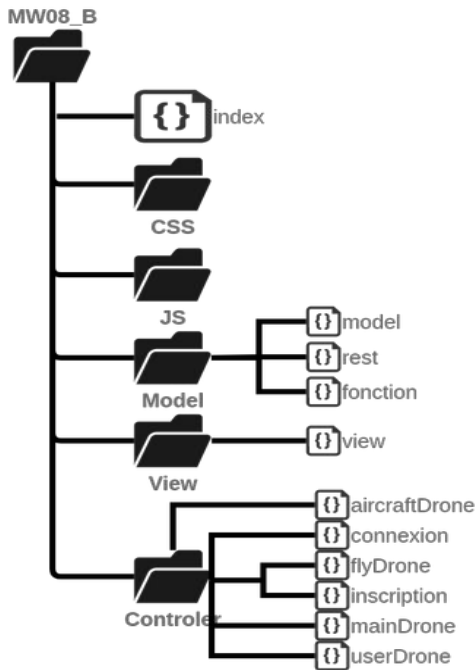


Figure1: Arborescence MVC

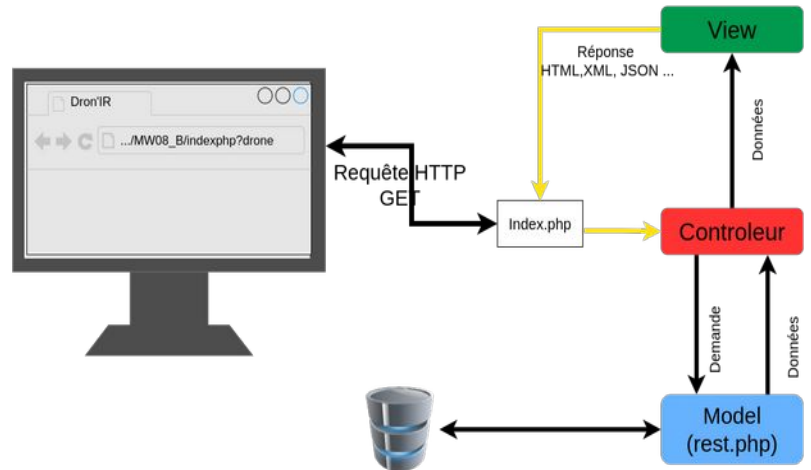


Figure 2: Architecture MVC

### Objectif :

Restructurer votre dossier selon le modèle MVC. La figure « Architecture MVC » permet de comprendre l'architecture MVC.

### Architecture MVC

L'organisation de votre dossier MW09 selon l'architecture MVC permet de séparer la logique de chaque fichier et de bien définir son rôle.

En vous aidant d'internet, donner le rôle de chaque dossier

? Le modèle sert à :

? La vue sert à :

### ? Le contrôleur sert à :

Lorsqu'un client souhaite afficher un contenu ou une page web, voici les étapes de téléchargement dans une architecture MVC :

1. Le lien renvoie vers l'index.php et ajoutant un mot clé dans l'URL correspondant au contenu demandé
2. En fonction du mot clé, l'index.php inclut le fichier du contrôleur
3. Si besoin, le contrôleur fait appel au modèle via une requête CURL
4. Le modèle envoie au contrôleur les données demandées
5. Le contrôleur traite les données et renvoie le résultat à la vue
6. La vue génère le code HTML à intégrer au site et l'affiche via la fonction "echo"

### ? En vous aidant de l'annexe, lister les mots clés à intégrer dans l'url en fonction de l'onglet de la barre de navigation

## Défi 1 - Mise en place du modèle MVC

### Trier les fichiers

L'arborescence de la figure 1 permet de réorganiser ses dossiers selon l'architecture MVC.

- </> Importer le site web MW09 dans votre serveur web linux et le tester en affichant la page index.php
- </> Appliquer l'organisation de la figure 1 à votre dossier de travail MW09
- </> Modifier si besoin le fichier index.php afin de retrouver votre site (css, js ...)

### Activer la barre de navigation

- </> Modifier le lien « href » des balises « <a> » de la barre de navigation en y intégrant des mots clés.  
Compléter la partie del'index.php afin d'inclure les fichiers nécessaire en fonction de la demande du client. Il faudra lire l'URL avec la méthode GET et sélectionner le bon fichier. Vous pouvez vous aider de l'annexe pour choisir le fichier.
- </> Vérifier le bon fonctionnement en affichant les formulaires de connexion et d'inscription

# PARTIE 2

## AFFICHER DES DONNÉES

### Objectif :

Dans cette partie, vous allez apprendre répondre à une demande du client selon l'architecture MVC

### TD 2 - Partie générale



Figure 3: Résultat onglet "Drone"

- ? Quelle fichier doit on inclure dans l'index.php afin d'afficher le résultat de la figure 3
- ? Quelle requête HTTP faut il envoyer à l'API pour récupérer le nombre de drones, de vols et d'utilisateurs ?
- ? Quelle requête SQL faut-il pour récupérer le nombre de drones dans la bdd ?
- ? Quelle requête SQL faut-il pour récupérer le nombre de vols dans la bdd ?
- ? Quelle requête SQL faut-il pour récupérer le nombre d'utilisateurs dans la bdd ?
- ? Selon le modèle MVC, donner l'ordre des fichiers qui seront appelés :

Pour répondre à cette problématique, nous utiliserons les fonctions include ou curl en php

## Défi 2 - Partie générale

### App.ini

C'est l'équivalent d'un fichier de configuration en programmation web.

**</>** Modifier le fichier app.ini contenant les paramètres de connexion à la base de données

### Index.php

**</>** Gérer le click sur l'onglet Drone en ajoutant dans le fichier index.php l'inclusion du bon fichier php (voir l'annexe).

**?** Pourquoi a t on des erreurs ?

### mainDrone.php

Le fichier mainDrone.php doit envoyer une requête HTTP au fichier rest.php en utilisant la fonction « CURL » La fonction permettant d'envoyer une requête CURL est « executerRequeteCurl() ».

**?** Quel fichier php contient la fonction précédente ?

**?** En étudiant la fonction précédente, lister les différentes fonctions « curl » nécessaire et dans l'ordre.

**</>** Modifier la variable \$url dans la fonction précédente « executerRequeteCurl() » afin de diriger la requête HTTP vers votre fichier rest.php

### rest.php

**</>** Compléter la partie « GET » selon les étapes suivantes :

```
si $req_data[1] vaut drone
    alors exécuter les trois requêtes précédentes (nombre de drones, vols et utilisateurs)
sinon rien pour l'instant

retourner la réponse sous format json
```

Voici le format json de la réponse :

```
{"nbdrone": "4", "nbvol": "56", "nbutilisateur": "10"}
```

**</>** Utiliser un client rest de votre navigateur pour tester votre api en envoyant une requête GET vers /rest.php/drone.

**</>** Vérifier maintenant le résultat en affichant votre page web « Drone »

### view.php

Afin de valider notre architecture MVC, nous devons maintenant utiliser la partie « view ». Dans notre TP, nous allons utiliser l'index.php comme fichier de vue. Pour cela vous aller devoir :

**</>** Remplacer l'utilisation de echo par une variable « \$reponseClient » dans le mainDrone.php

**</>** Faire un echo de la variable \$reponseClient dans le fichier index.php

**</>** Vérifier que le site affiche toujours le même résultat que la figure 3

## TD 3 - Données drones

| Dron'IR      |        |        |           |                     |               |
|--------------|--------|--------|-----------|---------------------|---------------|
| PRESENTATION |        | DRONE  |           | CONNEXION           | INSCRIPTION   |
| Numéro drone | Marque | Modèle | Référence | Date achat          | Action        |
| 1            | DJI    | Tello  | C4F41     | 2018-01-10 00:00:00 | Mettre à jour |
| 2            | DJI    | Tello  | C4F52     | 2018-01-10 00:00:00 | Mettre à jour |
| 3            | DJI    | Tello  | C4CBD     | 2018-01-10 00:00:00 | Mettre à jour |
| 4            | DJI    | Tello  | C4C4BD    | 2018-01-10 00:00:00 | Mettre à jour |

Figure 4: Représentation tableau de la liste des drones



Parmi les fichiers PHP, lequel permet de charger la liste des drones ?



Quelle requête HTTP faut il envoyer à l'API REST pour récupérer toutes les informations des drones ?



Donner le code à ajouter pour envoyer une requête HTTP



Quelle requête SQL faut-il pour récupérer toutes les informations concernant les drones dans la bdd ?

## Défi 3 - Données drones

- </> Compléter les fichiers PHP nécessaires aux résultats de la figure 4. L'API REST devra envoyer au contrôleur le résultat de la requête SQL complète sous forme JSON
- </> Modifier le fichier « aircraftDrone.php » afin de remplacer le « echo » par la variable \$reponseClient et d'appeler le fichier view.php tout en bas

## TD 4 - Données utilisateurs

| DRON'IR            |          |          |                             |                   |        |               |
|--------------------|----------|----------|-----------------------------|-------------------|--------|---------------|
| PRÉSENTATION       |          | DRÔNE    |                             | CONNEXION         |        | INSCRIPTION   |
| Numéro utilisateur | Nom      | Prénom   | Email                       | Date de naissance | Pseudo |               |
| 1                  | ABATAN   | Evan     | evan.abatan@gmail.com       | 1970-01-02        | login1 | Mettre à jour |
| 3                  | BECKER   | Maxime   | Maxime.BECKER@gmail.com     | 1970-01-04        | login3 | Mettre à jour |
| 4                  | BERANGER | Aurélien | Aurélien.BERANGER@gmail.com | 1970-01-05        | login4 | Mettre à jour |
| 5                  | BLIN     | Florent  | Florent.BLIN@gmail.com      | 1970-01-06        | login5 | Mettre à jour |
| 6                  | CESARI   | Jeffrey  | Jeffrey.CESARI@gmail.com    | 1970-01-07        | login6 | Mettre à jour |

Figure 5: Représentation tableau de la liste des utilisateurs



Parmi les fichiers PHP, lequel permet de charger la liste des utilisateurs ?



Quelle requête HTTP faut-il envoyer à l'API REST pour récupérer toutes les informations des utilisateurs ?



Donner le code à ajouter pour envoyer une requête HTTP



Quelle requête SQL faut-il pour récupérer toutes les informations concernant les utilisateurs dans la bdd ?

## Défi 4 - Données utilisateurs

</> Compléter les fichiers PHP nécessaires aux résultats de la figure 5. L'API REST devra envoyer au contrôleur le résultat de la requête SQL complète sous forme JSON

</> Modifier le fichier « userDrone.php » afin de remplacer le « echo » par la variable \$reponseClient et d'appeler le fichier view.php tout en bas

## TD 5 - Données vols

| DRON'IR      |                     |              |                 |               |              |
|--------------|---------------------|--------------|-----------------|---------------|--------------|
| PRÉSENTATION |                     | DRÔNE        |                 | CONNEXION     |              |
| Numéro vol   | Date de vol         | Numéro drone | Nom utilisateur |               |              |
| 70           | 2021-09-22 11:11:51 | 4            | Jean            | Mettre à jour | Afficher Vol |
| 67           | 2021-09-13 14:53:25 | 3            | hagot           | Mettre à jour | Afficher Vol |
| 66           | 2021-09-06 10:43:45 | 1            | hagot           | Mettre à jour | Afficher Vol |
| 65           | 2021-09-02 13:09:36 | 1            | hagot           | Mettre à jour | Afficher Vol |
| 64           | 2021-09-02 12:23:24 | 1            | hagot           | Mettre à jour | Afficher Vol |

Figure 6: Représentation tableau de la liste des vols



Parmi les fichiers PHP, lequel permet de charger la liste des vols ?



? Quelle requête HTTP faut il envoyer à l'API REST pour récupérer toutes les informations des vols?

? Donner le code à ajouter pour envoyer une requête HTTP

? Quelle requête SQL faut-il pour récupérer toutes les informations concernant les vols dans la bdd ?

## Défi 5 - Données vols

- </> Compléter les fichiers PHP nécessaires aux résultats de la figure 6. L'API REST devra envoyer au contrôleur le résultat de la requête SQL complète sous forme JSON
- </> Modifier le fichier « flyDrone.php » afin de remplacer le « echo » par la variable \$reponseClient et d'appeler le fichier view.php tout en bas

# PARTIE 3

## AFFICHER DES GRAPHE

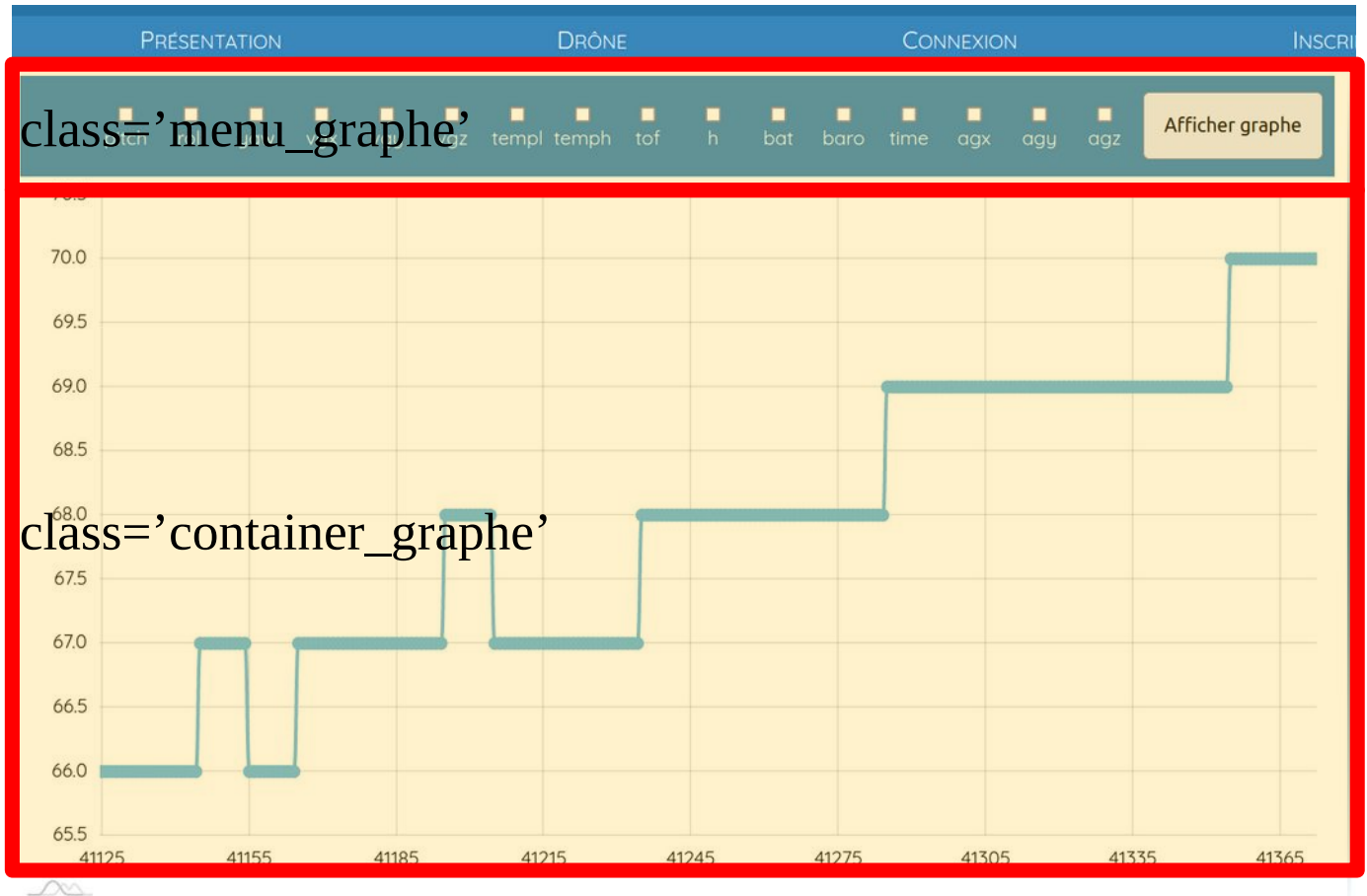


Figure 7: Menu de choix des états à afficher

### Défi 6 - Page graphe.php / Requête du client

</> Créer un fichier graphe.php

</> Diviser le contenu en deux divisions. La première sera réservée au menu et la deuxième au graphe JS. Elles auront pour id : « menu\_graphe » et « container\_graphe »

</> Insérer, en HTML, un menu avec les différents champs présents dans la base de données. Tous les champs seront dans un formulaire HTML, balise form avec comme action « index.php » et méthode « get » avec des checkbox permettant de faire le choix de l'utilisateur. Veuillez à ajouter les attributs for, « value » et « id ». Exemple : <label class='etat' for='tof'><input type='checkbox' name='etat[]' value='tof' id='tof'>tof</label>

? Pourquoi a t on écrit name='etat[]' et non pas name='etat' ?

</> Ajouter un bouton « submit » permettant de valider le formulaire.

Lorsque l'on valide le formulaire, nous n'avons plus aucun moyen de connaître l'idvol choisit. Nous allons donc l'intégrer au formulaire mais de manière caché

? Donner la ligne de code à ajouter au formulaire

</> Modifier le fichier graphe.php et index.php si besoin pour réafficher la page graphe.php

</> Choisissez les cases pitch et roll puis valider sur le bouton Afficher graphe

? Que voyez vous dans la barre URL ?

## Défi 7 - Créer/Envoyer les données JSON

Maintenant que nous avons dans l'URL le choix du client, nous devons compléter le fichier graphe.php afin d'envoyer une requête HTTP vers notre API.

La requête HTTP à envoyer est la suivante : **GET /rest.php/etat/numero\_idvol/pitch/roll** selon le choix du client à la fin du Défi 6 - par exemple.

? Quel code faut il ajouter pour exécuter une requête http précédente dans le fichier graphe.php. Vous pouvez utiliser \$\_GET['etat'] afin de récupérer tous les états choisis

? Quelle requête SQL faut il alors écrire dans l'API REST pour récupérer dans la table etat uniquement des champs et les enregistrements correspondants à la requête HTTP du Défi 6 - ? Nous ajouterons les valeurs de idetat pour avoir l'axe horizontal dans le graphe

? Donner le code complet à intégrer dans l'API REST afin de répondre à la problématique précédente tout en généralisant à tous les champs possible de la table etat

</> Compléter et tester votre fichier graphe.php

## Défi 8 - Créer un graphe JS

Pour la création de graphe en javascript, nous allons utiliser la bibliothèque amchart (<https://www.amcharts.com/docs/v4/getting-started/>). Nous pouvons soit utiliser la librairie en local soit en ligne. Nous allons choisir la deuxième solution.

? Quelles sont les scripts alors à ajouter afin d'utiliser amchart ?

? Lister deux types différents de graphe possible :

Dans la suite, nous allons utiliser les « XY chart »

? Quel code JS faut-il ajouter pour créer un graphe de type « XY chart » ?

? Que représente le code js « chart.data »

? Par quoi faut-il remplacer « chart.data » pour récupérer des données d'une source extérieure ?

? Pour créer les axes, il faut utiliser «CategoryAxis() et ValuesAxis() ». A quoi correspondent ces valeurs ?

? En vous aidant de l'exemple de l'annexe compléter le code permettant d'afficher un graphe de type XY, avec en X la valeur de l'idetat et en Y la valeur de la batterie en utilisant les données JSON récupérer depuis l'API REST

```
var chart = am4core.create("chartdiv", am4charts.  
// Set up data source  
chart.data =  
  
// Create axes  
var categoryAxis = chart.xAxes.push(new am4charts.  
categoryAxis.dataFields.category = "idetat";  
  
// Create value axis  
var valueAxis = chart.yAxes.push(new am4charts.  
  
// Create series  
var series1 = chart.series.push(new am4charts.  
series1.dataFields.valueY =  
series1.dataFields.categoryX = "idetat";  
series1.name =  
series1.strokeWidth = 3;  
series1.tensionX = 0.7;  
series1.bullets.push(new am4charts.CircleBullet());
```

## Défi 9 - Afficher un graphe JS

Pour cette partie, nous allons intégrer du code JS dans une page PHP. Vous allons donc devoir, en utilisant un echo, créer une variable js et l'initialiser avec des données issues d'une variable PHP

- ? Donner le code php permettant de créer une variable js « etatsVol » et d'y insérer les données json récupérer depuis l'API REST

- </> Intégrer le code à la suite de votre fichier graphe .php  
</> Ajouter une div avec un id='chartdiv' dans le « container\_graphe »  
</> Vérifier le bon fonctionnement en affichant les données de la batterie

Il faut maintenant être capable d'afficher autre chose que la batterie. Nous devons pour cela récupérer la valeur de l'objet JSON envoyé (pitch,roll, yaw...) afin de modifier l'axe des X et de pouvoir récupérer les valeurs dans le format JSON. Voici un exemple avec la valeur de « templ » :

```
let etatsVol='[{"idetat":"41125","templ":"66"}, {"idetat":"41126","templ":"66"}, {"idetat":"41127","templ":"66"}, {"idetat":"41128","templ":"66"}, {"idetat":"41129","templ":"66"}, {"idetat":"41130","templ":"66"}, {"idetat":"41131","templ":"66"}]';
```

Il existe une méthode JS permettant de récupérer les clés idetat et templ.

```
let listeEtat = Object.keys(etatsVol[0]);
```

- ? Donner le code permettant de récupérer le terme « idetat » et « templ » à partir de la variable listeEtat.

- </> Intégrer la partie précédente dans votre code JS et vérifier le bon fonctionnement en affichant différent état.  
Pour afficher plusieurs graphe, il faut ajouter autant de série que nécessaire.  
</> Modifier votre code JS afin de tenir compte de tous les paramètres d'état de vol possible

## Défi 10 - Afficher plusieurs courbes

Pour afficher plusieurs courbes, il faut ajouter autant de série que nécessaire.

- </> Modifier votre code js afin d'afficher plusieurs courbes si le client le souhaite

# PARTIE 4

## AJAX

### Objectif :

Faire les mêmes requêtes HTTP mais en javascript en utilisant l'AJAX.

**</>** Faire la même chose mais en utilisant uniquement le langage JS avec l'AJAX et le XMLHttpRequest() et ne plus passer par les fichiers PHP

# ANNEXE

## Liste des requêtes HTTP sur index.php

### GET

| URL                           | Description                                   | Fichier           |
|-------------------------------|---|-------------------|
| Index.php?connexion           | Afficher le formulaire de connexion           | connexion.php     |
| Index.php?inscription         | Afficher le formulaire d'inscription          | inscription.php   |
| Index.php?drone               | Afficher la page de drone                     | mainDrone.php     |
| Index.php?drone&nbdrones      | Afficher le table de données des drones       | aircraftDrone.php |
| Index.php?drone&nbvol         | Afficher le table de données des vols         | flyDrone.php      |
| Index.php?drone&nbutilisateur | Afficher le table de données des utilisateurs | userDrone.php     |

## Liste des requêtes HTTP sur rest.php

### POST

| URL  | Description            | Données |
|------|------------------------|---------|
| /vol | Insérer un nouveau vol | json    |

### GET

| URL                  | Description  | Données |
|----------------------|--|---------|
| /drone               | Récupérer le nombre de drones, de vols et d'utilisateurs | json    |
| /drone/nbdrones      | Récupérer toutes les données de la table drone           | json    |
| /drone/nbvol         | Récupérer toutes les données de la table vol             | json    |
| /drone/nbutilisateur | Récupérer toutes les données de la table utilisateur     | json    |

### PUT

| URL                  | Description   | Données |
|----------------------|---|---------|
| /drone/nbdrones      | Mettre à jour un enregistrement de la table drone       | json    |
| /drone/nbvol         | Mettre à jour un enregistrement de la table vol         | json    |
| /drone/nbutilisateur | Mettre à jour un enregistrement de la table utilisateur | json    |

## Exemple amchart de type XYChart

```
<div id="chartdiv" style="width: 100%; height: 400px;"></div>

<script>
// Create chart instance
var chart = am4core.create("chartdiv", am4charts.XYChart);

// Add data
chart.data = [{
  "country": "Lithuania",
  "litres": 501.9,
  "units": 250
}, {
  "country": "Czech Republic",
  "litres": 301.9,
  "units": 222
}, {
  "country": "Ireland",
  "litres": 201.1,
  "units": 170
}, {
  "country": "Germany",
  "litres": 165.8,
  "units": 122
}, {
  "country": "Australia",
  "litres": 139.9,
  "units": 99
}, {
  "country": "Austria",
  "litres": 128.3,
  "units": 85
}, {
  "country": "UK",
  "litres": 99,
  "units": 93
}, {
  "country": "Belgium",
  "litres": 60,
  "units": 50
}, {
  "country": "The Netherlands",
  "litres": 50,
  "units": 42
}
];

// Create axes
let categoryAxis = chart.xAxes.push(new am4charts.CategoryAxis());
categoryAxis.dataFields.category = "country";
categoryAxis.title.text = "Countries";

let valueAxis = chart.yAxes.push(new am4charts.ValueAxis());
valueAxis.title.text = "Litres sold (M)";

// Create series
var series = chart.series.push(new am4charts.ColumnSeries());
series.dataFields.valueY = "litres";
series.dataFields.categoryX = "country";
series.name = "Sales";
series.columns.template.tooltipText = "Series: {name}\nCategory: {categoryX}\nValue: {valueY}";
series.columns.template.fill = am4core.color("#00ff00"); // fill
```