# Site web

## Lancer le site

* Se placer dans le répertoire /site
* Entrer la commande php -S localhost:9000, puis accéder depuis un navigateur à l’adresse localhost:9000

A noter le port 9000 peut-être changé par le port que l’on souhaite, tant qu’il ne se superpose pas au ports réservés (soit les ports en dessous de 1024) <https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers>

## PHP

A la fin des requêtes php il faut retourner une donnée au format JSON.  
echo “mon truc format JSON”

### Debug

Pour debug php, faire un echo puis dans le navigateur, lancer l’inspecteur d’éléments > réseaux > MonFichier.php (appelé pour récupérer les données et qui aura réalisé le echo)

### Retours

On crée un tableau pour le renvoyer avec echo json\_encode

Pour faire un tableau et ajouter des éléments en php :  
```

$var = array();

array\_push($var, NouvTruc);

```

### Connection à une base de données MongoDB

```

$manager = new MongoDB\Driver\Manager("mongodb://mongo1:30001,mongo2:30002,mongo3:30003/?replicaSet=rs0");

```

Lien non fonctionnel pour le moment

## Base de donnée MongoDB

(Voir le README.md du dossier /bdd pour le lancement, l’arrêt, l’ajout de données et la structure de la base)

### Structure MongoDB

La structure d’une base mongo db :  
database

collections

document

### Accès à la BDD

Par lien:

| mongodb://mongo1:30001,mongo2:30002,mongo3:30003/?replicaSet=rs0i  # ou pour se connecter directement à la bonne db (data):  mongodb://mongo1:30001,mongo2:30002,mongo3:30003/data/?replicaSet=rs0 |
| --- |

Par docker:

| docker exec -it mongo1 mongosh --port 30001 |
| --- |

Pour le moment les liens ne fonctionnent pas

### Commandes utiles

Documentation :

* <https://shellhacks.com/mongodb-show-collection-data-mongo-shell/>
* <https://www.php.net/manual/en/class.mongodb-driver-query.php>
* <https://www.mongodb.com/docs/php-library/current/reference/method/MongoDBCollection-listIndexes/>
* <https://www.mongodb.com/docs/manual/reference/method/db.collection.getIndexes/>

**> show dbs**

Montre le contenu du répertoire actuel de la bdd

**> use data**

Entre dans le répertoire data

**> show collections**

Montre les différentes données dans data

**> db.agriculteur.find()**

Affiche les différentes données dans la table agriculteur

**> db.collection.getIndexes()**

Permet d’accéder à l’index de l'îlot

## 

## Données

Les données entre backend et frontend sont à envoyer au format JSON. Il faudra retourner les données suivantes :

* les numéros de champs, de l’agriculture ;
* les numéros d’îlots, dans le champ ;
* le nom d'utilisateur ; (à discuter après le jalon)
* les données moyennes les plus récentes de température/humidité/luminosité, pour le champ donné (en une unique réponse) ;
* "OK" ou "ERR", si un capteur du champ n'envoie plus de données depuis par exemple 30 minutes (soit 10 mesures). Toujours dans le retour, le nombre de capteurs fonctionnels et la date, au format local français (UTC+1/+2), à laquelle la requête a été effectuée (pour indiquer sur le site, "mis à jour le XXX à XXX");
* les données de tous les capteurs d'un champ (dans l’ordre, ID, Ilot, Date/heure, Temp, Humi, Lumi) ;
* les données, pour une seule unité de mesure, pour un îlot spécifique, sur 30 jours.

Les champs envoyés au backend sont disponibles dans le fichier site/scripts/afficherDonnees.js.

Ils se trouvent dans les fonctions éponymes de ce qu’elles s’attendent à récupérer dans le backend, dans la variable champPost.

Par exemple, la fonction afficherMoyennes(), va envoyer en POST au fichier recupMoyennes.php, avec dedans numChamp.

Cela va permettre de récupérer dans le backend, l’ID du champ sur lequel l’on souhaite récupérer la moyenne des mesures de température, humidité, luminosité, pour la dernière mesure réalisée.

Le retour sera donc, sous format JSON, dans le format suivant [température, humidité, luminosité] avec pour valeurs, par exemple, [22.1, 80.8, 30000].