

# PHP

## - 1 - Objet et automatisation

Objectif : réaliser un site qui affiche le contenu d'une base de données de manière automatisées, en séparant les responsabilités.

Créez deux tables sur des thèmes libres, dans le même schéma et insérez-y des données.

Créez les fichiers suivants :

- **index.php**
- **db.php**
- **Model.php**
- Deux fichiers php portant le nom de vos tables, avec une majuscule
- **controller.php**
- **view.php**

**index.php** se contente de charger les scripts dans l'ordre :

```
function loadMyClass($className) { include_once $className.".php"; }
spl_autoload_register('loadMyClass');

include_once "db.php";
include_once "controller.php";
include_once "view.php";
```

Les fichiers du modèle ne sont pas à inclure : ils contiennent des classes qui seront chargées automatiquement.

Le contrôleur doit récupérer le paramètre GET nommé « table » pour charger les données.

```
$table = $_GET["table"];
$class = ucfirst($table);
$data = $class::all();
```

La vue (**view.php**) contient l'entête HTML et affiche les données (**var\_dump(\$data)** dans un premier temps).

Créez les classes du modèles et la fonction **all** : elle doit charger toutes les données de la table et les retourner sous la forme d'une liste (**array**) d'instances. Cette fonction doit être la plus générique possible.

Faites en sorte que la vue soit plus élégante qu'un **var\_dump**, avec un affichage sous forme de **<table>** par exemple.

Pour aller plus loin :

- Gérez l'erreur de l'absence de paramètre GET.
- Si un « id » est donné en paramètre GET, ne chargez que cette instance. Faites en sorte que la vue s'adapte entre une liste d'instances et une instance seule.

Pour aller beaucoup plus loin :

Créez deux table liées (exemple : game et category) et gérez la jointure : Faites en sorte qu'au chargement d'un objet, l'objet lié soit chargé aussi, sous forme d'instance directement accessible :

```
$g = Game::load(1);  
echo $g->category->name;
```