

Je devais faire une copie d'un gros fichier pour un camarade, ça a pas mal ralenti ma machine.

8:55 : La base de données est installée, je vais la lier à bd.inc.php

QUESTIONS PARTIE 1 :

1.1. A l'aide de la documentation officielle PHP, rappeler le rôle des mots clés suivants :

`include_once`, `include`

`include` : L'expression de langage `include` inclut et exécute le fichier spécifié en argument.

`include_once` : L'instruction `include_once` inclut et évalue le fichier spécifié durant l'exécution du script. Le comportement est similaire à `include`, mais la différence est que si le code a déjà été inclus, il ne le sera pas une seconde fois, et `include_once` retourne **true**. Comme son nom l'indique, le fichier sera inclut une seule fois.

La structure `include_once` est utilisée de préférence lorsque le fichier va être inclus ou évalué plusieurs fois dans un script, ou bien lorsque vous voulez être sûr qu'il ne sera inclus qu'une seule fois, pour éviter des redéfinitions de fonctions ou de classes.

(<https://www.php.net/manual/fr>)

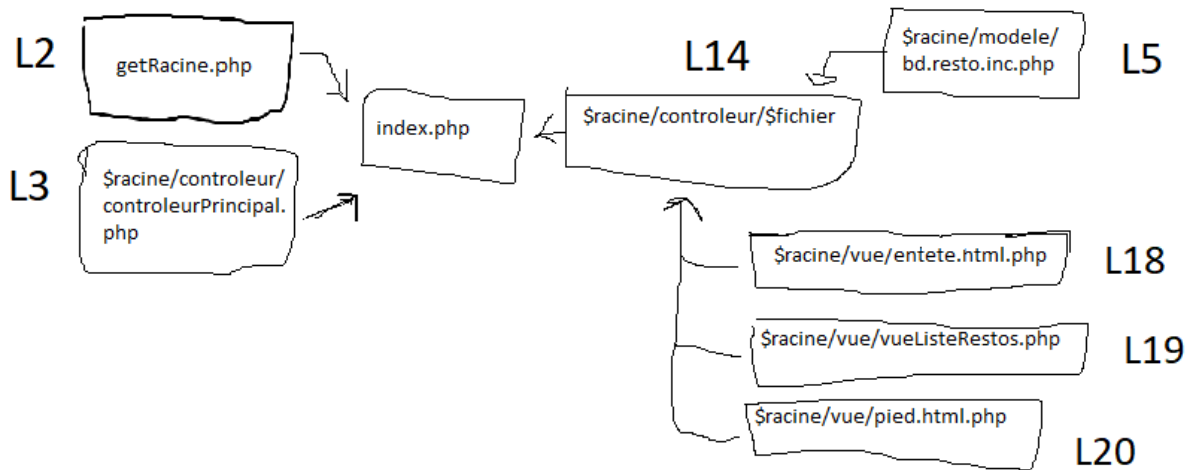
1.2. Déterminer la valeur de la variable `$fichier` en affichant son contenu à la ligne précédant la balise de fin de PHP dans `index.php`.

Je n'arrive pas à lancer la page `index`
c'est bon.

`echo $fichier` renvoie `listeRestos.php`, je comprends pas ce que fait la fonction `controleurPrincipal()`, je devine qu'elle permet de récupérer ce que demande l'utilisateur, puisqu'il y a un autre fichier `detailResto.php` dans le même répertoire, et que le `include` qui affiche le tout fonctionne ainsi : `"$racine/controleur/$fichier"` donc ici `"site P1/controleur/listeRestos.php"`, mais qui pourrait aussi être `site P1/controleur/detailResto.php`

1.3. En partant du fichier `index.php`, schématiser l'ensemble des fichiers utilisés (`include` ou `include_once`) pour afficher la liste des restaurants à l'aide de :

- rectangles portant le nom de chaque fichier,
- flèches pointant vers le fichier utilisé (inclus). Sur la flèche, indiquer la ligne qui pointe vers le fichier.



1.4. Relever l'ensemble des requêtes SQL contenues dans les fonctions déclarées dans le fichier `bd.resto.inc.php`.

Fonction	Requête
<code>getRestoByIdR</code>	<code>select * from resto where idR=:idR</code>
<code>getRestos</code>	<code>select * from resto</code>
<code>getRestosByNomR</code>	<code>select * from resto where nomR like :nomR</code>
<code>getRestosByAdresse</code>	<code>select * from resto where voieAdrR like :voieAdrR and cpR like :cpR and villeR like :villeR</code>

1.5. Quels sont les points communs de ces requêtes SQL ?

C'est que des select pour pointer sur des données dans la table resto*

1.6. En observant le nom des fonctions, le résultat affiché, et la requête contenue dans la fonction, expliquer d'une manière très générale ce que fait chacune des fonctions présente dans le script `bd.resto.inc.php`.

La fonction `getRestos()` permet de récupérer tous les restos dans la base. Son résultat d'exécution est :

```

getRestos() :
Array
(
    [0] => Array

```

```
(  
  [idR] => 1  
  [nomR] => l'entrepote  
  [numAdrR] => 2  
  [voieAdrR] => rue Maurice Ravel  
  [cpR] => 33000  
  [villeR] => Bordeaux  
  [latitudeDegR] => 44.7948  
  [longitudeDegR] => -0.58754  
  [descR] => description  
  [horairesR] => <table>...</table>  
)
```

[1] => Array

```
(  
  [idR] => 2  
  [nomR] => le bar du charcutier  
  [numAdrR] => 30  
  [voieAdrR] => rue Parlement Sainte-Catherine  
  [cpR] => 33000  
  [villeR] => Bordeaux  
  [latitudeDegR] =>  
  [longitudeDegR] =>  
  [descR] => description  
  [horairesR] => <table>...</table>  
)
```

[2] => Array

```
(  
  [idR] => 3  
  [nomR] => Sapporo  
  [numAdrR] => 33  
  [voieAdrR] => rue Saint Rémi  
  [cpR] => 33000  
  [villeR] => Bordeaux  
  [latitudeDegR] =>  
  [longitudeDegR] =>  
  [descR] => Le Sapporo propose à ses clients de délicieux plats typiques japonais.  
  [horairesR] => <table>...</table>  
)
```

////////// extrait tronqué //////////

[10] => Array

```
(  
  [idR] => 11  
  [nomR] => Trinquet Moderne  
  [numAdrR] => 60
```

```

        [voieAdrR] => Avenue Dubrocq
        [cpR] => 64100
        [villeR] => Bayonne
        [latitudeDegR] =>
        [longitudeDegR] =>
        [descR] => description
        [horairesR] => <table>...</table>
    )
)
getRestoByIdR(idR) :
Array
(
    [idR] => 1
    [nomR] => l'entrepote
    [numAdrR] => 2
    [voieAdrR] => rue Maurice Ravel
    [cpR] => 33000
    [villeR] => Bordeaux
    [latitudeDegR] => 44.7948
    [longitudeDegR] => -0.58754
    [descR] => description
    [horairesR] => <table>...</table>
)

```

La fonction `getRestosByNomR()` permet de récupérer les restos par nom avec un like. L'argument est placé dans le like et permet de récupérer dans la base toutes les chaînes de la colonne `NomR` qui contiennent l'argument.

`getRestos` par adresse fait la même chose mais au lieu d'aller chercher l'argument dans la colonne `nomR`, il va les chercher dans trois colonnes différentes, la colonne `voieAdrR`, la colonne `cpR`, et la colonne `villeR`.

1.7. Quelle fonction définie dans le script `bd.resto.inc.php` est utilisée dans `listeRestos.php` ?
`getRestos ()`

QUESTIONS PARTIE 2 :

2.1. Trouve-t-on des éléments de CSS ou de HTML dans les fichiers des dossiers contrôleur et modèle ?

Non, il n'y a que du php.

2.2. Dans quel dossier sont contenus les fichiers gérant les éléments de HTML et de CSS ?

Dans les dossiers vue et css.

2.3. Le code PHP contenu dans `vueListeRestos.php` est-il toujours visible après réception par le navigateur ? Par quoi est-il remplacé ?

Par rien du tout, le code est exécuté mais la page n'affiche que du html ?

2.4. Rappeler le point commun entre toutes les fonctions définies dans le script `bd.resto.inc.php` contenu dans le dossier modèle.

Elles font que des select ?

2.5. Trouve-t-on dans d'autres fichiers que ceux du dossier modèle des références à la base de données ?

Seulement dans le contrôleur

Synthèse : expliquer globalement le rôle des scripts contenus dans les dossiers suivants

modèle : Lie la page à la base de données.

vue : Affiche la page à l'utilisateur.

QUESTIONS PARTIE 3 :

3.1. Quelle fonction d'accès aux données est utilisée dans ce contrôleur ? Que permet-elle de récupérer dans la base de données ?

La fonction `ControleurPrincipal`, prenant pour argument la variable `$action`, qui change en fonction de ce sur quoi l'utilisateur clique, et renvoie par défaut à la page `listeRestos.php`, sinon à la page que désigne `$action`

3.2. Les données récupérées sont-elles affichées à l'écran ? L'affichage est-il fait dans le script contrôleur ?

Non, elles le seront après, dans la vue, le contrôleur n'affiche rien, il récupère juste les données du modèle.

3.3. Quels scripts sont inclus dans les dernières lignes du contrôleur ? Quels sont leurs rôles ?

A garder la structure du site lorsque que l'utilisateur change de page?

3.4. Quelle donnée est transmise au contrôleur en méthode GET ?

Le contenu de la variable `$action`, donc l'endroit où l'utilisateur veut aller dans le site.

3.5. Rappeler le rôle de la fonction `getRestoByIdR()`. Pourquoi cette méthode a-t-elle besoin d'un paramètre ?

Elle récupère un restaurant en utilisant son `idR`, unique à chaque restaurant dans la table, elle a besoin d'un argument car sans, elle ne peut pas le mettre dans sa requête SQL. (La variable `$unResto` est créée lors de l'appel à la fonction `getRestoByIdR()`.)

3.6. Explorer les fichiers "vue" inclus à la fin du fichier contrôleur et repérer les lignes où sont utilisées cette variable.

```
<h1><?= $unResto['nomR']; ?></h1>

<span id="note"></span>

<section>    Cuisine <br /> </section>

<p id="principal">    <?= $unResto['descR']; ?></p>

<h2 id="adresse">    Adresse</h2>

<p> <?= $unResto['numAdrR']; ?>

    <?= $unResto['voieAdrR']; ?> <br />

    <?= $unResto['cpR']; ?>

    <?= $unResto['villeR']; ?> </p>

<h2 id="photos">    Photos</h2>

<ul id="galerie"></ul>

<h2 id="horaires">    Horaires</h2>

<?= $unResto['horairesR']; ?>

<h2 id="crit">Critiques</h2>

<ul id="critiques"></ul>
```

ce code est extrait de vueDetailResto.php, et la variable \$unResto sert à afficher des infos sur le resto en question.

3.7. Deux autres variables créées dans le contrôleur sont affichées dans la vue. Lesquelles ?

\$listeRestos et \$menuBurger

3.8. Le contrôleur gère-t-il directement l'accès aux données ?

Non, son rôle est seulement de gérer la circulation sur le site, c'est le modèle qui gère les données.

Où sont affichées les données créées ou récupérées dans le contrôleur ?

dans la vue

Les données utilisées par le contrôleur peuvent provenir de 2 sources : l'utilisateur ou la base de données.

Comment ces données sont transmises au contrôleur :

de l'utilisateur vers le contrôleur ?

grace au include dans index.php

de la base de données vers le contrôleur ?

grace aux fonctions dans bd.resto.inc.php et bd.inc.php

Le nom du composant "contrôleur" est justifié par son rôle dans le patron de conception MVC :

il fait travailler ensemble la partie Affichage (vue) et la partie données (modèle) en récupérant les données transmises par l'utilisateur et en effectuant éventuellement des traitement sur ces données.

Le fonctionnement du contrôleur sera étudié plus en détail dans la partie suivante.