AP2

## AP2.2 – Projet Netcar Mission 1 : Découverte de l'architecture MVC



MH

Liens vers le	Bloc 1	Support et mise à disposition des services informatiques
référentiel	Compétence 1.3	Développer la présence en ligne de l'organisation
referencier	Sous-compétence	Participer à l'évolution d'un site Web exploitant les données de
		l'organisation
	Notions	Langage d'interrogation de données - SQL partie LID :
	Bloc2	Conception et développement d'applications
	Compétence 2.1	Conception et développement d'une solution applicative
	Sous-compétence	Intégration continue des versions d'une solution applicative
	Bloc2	Conception et développement d'une solution applicative
	Compétence 2.1	Participer à la conception de l'architecture d'une solution applicative
	Sous-compétence	Architecture MVC – Modèle Vue Controleur

L'architecture MVC (Modèle - Vue - Contrôleur) est un patron de conception (design pattern).

Un patron de conception est un modèle ayant fait ses preuves par le passé et qui est considéré comme une bonne pratique pour un objectif spécifique.

Les avantages d'un patron de conception sont :

- fournir une structure standard qu'il suffit d'adapter à son besoin
- accélérer le processus de développement
- augmenter la qualité (code optimisé et fiabilité)
- faciliter la maintenance grâce à une structure connue

## **1. Principe**

Dans l'architecture MVC, l'organisation des sites web ressemble à ceci :

📕 images 🛛 🛛	Le point d'entrée est toujours le fichier <i>index.php</i> , il joue le rôle d'aiguilleur dans le site.
Css	Le dossier <b>Vues</b> contient toutes les pages fournissant le contenu à afficher (interface graphique)
<ul> <li>Controleurs</li> <li>Modeles</li> <li>Vues</li> <li>index.php</li> </ul>	Le dossier <i>Controleurs</i> contient les fichiers qui vont contenir les règles métier du site (spécifiques aux données et procédures de l'entreprise) Le dossier <i>Modeles</i> contient les fichiers interagissant avec les données stockées (Base de données, fichiers XML, fichiers Ison,)

# 2. Les 3 composants

### 2.1. Les vues

### **1.1.1.Vue externe**

L'aperçu global du site visible par l'utilisateur s'appelle la vue externe. Techniquement, elle est obtenue à partir de plusieurs vues assemblées, par exemple : l'en-tête, le contenu principal et le bas de page.

### 1.1.2.Sous-vues

Il s'agit des différentes pages permettant de construire la vue externe visible par l'utilisateur. En général, une partie de la vue externe est identique sur l'ensemble de la navigation et seule une zone change en fonction de la demande de l'utilisateur. Cette zone variable est une vue adaptée à l'action spécifiée en paramètre et réalisée par le sous-contrôleur spécifique.

# **3. Les contrôleurs**

### 3.1. Contrôleur principal

La page index.php joue le **rôle d'aiguilleur.** Ceci signifie qu'elle va orchestrer le travail et/ou l'affichage des composants du site pour répondre à la demande de l'utilisateur. On l'appelle le contrôleur principal.

Dès que l'utilisateur clique sur une fonctionnalité, l'url du site se complète afin de donner les informations nécessaires à la page index.php.

L'url respecte systématiquement la structure suivante :

http://adresseDuSite/index.php?controleur=gestionAgence&action=consulterLesAgences

L'url comporte 2 paramètres passés par la méthode  $\ensuremath{\textbf{GET}}$  :

- controleur : indique quel élément du site est concerné (en général, un élément correspond à une table de la base de données métier)
- **action** : indique l'opération souhaitée par l'utilisateur sur l'élément passé en premier paramètre (en général : consultation, ajout, suppression, modification)

Le contrôleur principal dirige la demande au sous-contrôleur spécifique selon la valeur du paramètre « controleur » .

### 3.2. Sous-contrôleurs

Chaque sous-contrôleur, appelé par le contrôleur principal, va orchestrer le travail nécessaire à la réalisation de l'action passée en paramètre (valeur du paramètre action). Ceci peut nécessiter :

- la récupération de données via les modèles
- l'affichage de résultats à l'utilisateur via les vues

### 4. Les modèles

Il s'agit des fichiers permettant d'accéder aux données nécessaires à l'application web. Ces fichiers contiennent la connexion mais également toutes les méthodes utiles aux sous-contrôleurs pour réaliser les actions - **sur les données -** demandées par l'utilisateur.

L'avantage de ce composant est de rendre l'application moins dépendante du support de stockage des données. Ainsi, si les données sont stockées dans une base de données relationnelles, puis que plus tard, elles sont basculées dans des fichiers xml, seule la partie Modèles sera à adapter. Le reste de l'application restera inchangé.

# 5. Observation détaillée du mécanisme

### 5.1. Préparer le projet et tester son fonctionnement

- 1. Lancer Netbeans et ouvrir le projet T:\AP2\AP2.2ProjetPHPNetcarMVC (cf séance précédente)
- 2. Lancer WampServer (il est conseillé de se replacer sur le même ordinateur afin de ne pas à avoir à reparamétrer votre projet et à réimplanter la base de données).
- 3. Vérifier le bon fonctionnement du site.

← → C <sup>(i)</sup> localhost.8	3000/index.php		Q \$
NETCAR - La loc	ation de véhicules a	adaptée à VOS besoins	Version : V1 - Site initial + personnalisation de la vue externe
	Confiez-nous vot	re besoin!	
Retour à l'accueil	NET		URES-
Consulter nos agences	INCI	LE MEILLEUR DE LA LOCATION	DE VOITURE
	<u> @Lę</u>		
	and the	Il est actuellement : 10:16	
Site développé par Netrar et modifié par mol ()			
← → C ③ localhost:80	00/index.php?controleur=g	gestionAgence&action=consulterLes	Agences 🍳 🕁
NETCAR - La loca	tion de véhicules a	adaptée à VOS besoins	Version : V1 - Site initial + personnalisation de la vue externe
	Nos agences		
	Nom	Ville	
Retour à l'accueil	La Baule	LA BAULE	Louer un véhicule
Consulter nos agences	La bordelaise	BORDEAUX	Louer un véhicule
	La bordelaise du lac	BORDEAUX	Louer un véhicule
	La nantaise	MANTES	Louer un véhicule
	Saint-Nazaire	MERIONAL SAINT NAZAIRE	Louer un véhicule
	And a statement of		Louer un véhicule
	Netcar comporte 6 a	gences pour être au plus proche de	vous.

# 6. Analyser la structure du site fourni

4. Ouvrir le fichier index.php dans Netbeans et répondre aux questions :



### 5. Ouvrir, <u>dans Netbeans</u>, le fichier **vues\v\_layout\_entete.php** (vue)

Indiquer le rôle des instructions situées aux lignes : 5, 12, 14, 23-24, 27-28, 37-38

Indiquer le role des instructions situées aux lignes : 5, 12, 14, 23-24, 27-28, 37-38	
1	
4 (meta charset="utf-9" />	
s <pre>demonstration = "width=device-width, initial-scale=!" /&gt;</pre>	
<pre>ctitleNFTClRc/title&gt;</pre>	
4 Clink href="bootstrap.css/bootstrap.min.css" rel="stylesheet" type="text/css">	
$10 \rightarrow 2$	
11	
12 <pre>dink href="https://cdn.jsdelivr.net/nnm/bootstrap\$5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xW2eSR50omGNYDhhzAbDs0XxcvSN1TPprVMTNDbiYZCxYb0017+AMvvTG2x" crossorigin="anonymy"</pre>	us">
13	
14 <li>link href="/css/netcar.css" rel="stylesheet" type="text/css"&gt;</li>	
15	
16	
17	
18 📩 dbody>	
19	
😪 📩 🗯 📶 🖓 🙀 🖓 🔄 🖓 🔄 🖓 🔄 🖓 🔄 🖓 👘	
21 cheader> à ajouter : en-tête	
Guity id="entete" class="ħj;     Class="ħ	
<pre>aa <div> <img alt="NetCar_icone" id="icone" src="./inages/netCar_icone.jpg"/> </div></pre>	
<pre></pre>	
25 /div>	
א 📋 🛛 <div class="text-success" id="version"></div>	
<pre>27</pre>	
28 (div> Version : V1 - Site initial + personnalisation de la vue externe	
29 _	
30	
al _	
Ma L	
as - (nav class=">	
$\frac{26}{10}$	
47 Cline inter-inter.php / Record a racuerity and and the segment computer and approach (a) (1)	
All A met- mex.phy.concluten-gestionalien-constiteriesagences /constiteriesagences/a//ii/	
a) (July )	
42 <	
Ligne 12 ·	

Ligne 12 : Ligne 14 : Ligne 23-24 : Ligne 27-28 : Ligne 37-38 :

### 6. Ouvrir le fichier controleurs \c\_gestionAgence.php

<u> </u>		php</th <th></th> <th></th>		
2		include 'Modeles	/AgenceDAO.php';	
3		<pre>\$objAgence= new</pre>	AgenceDAO();	
Â		if(!isset(\$ GET[	['action']))	
៊		Saction = 'c	consulterLesAgences';	
6		else		
O.		Saction = \$	GET['action']:	
8		10000000000000000000000000000000000000		
9		switch(Saction)		
10		{		
11	H	. case !co	ngulterLeslgences! •	
12		1		
13			(les]gences = (obj]gence_>getIes]gences().	
14			<pre>vicoxgenees voojkgenee vgebleokgenees();</pre>	
15			include/"www.w.congulterLeg/gences.nhn").	
16			include("vues/v_consulterLesAgences.php");	
17				- ŝeve su plus preste de pres ".
17			<pre>smessage = "NetCar comporte ". count(siesAgences). " agences pou </pre>	r etre au plus proche de vous.";
18			<pre>\$stylePresentation = "text-success";</pre>	
19			break;	
20		}		
21				
22	L	}		

a.	A quoi sert la structure alternative 'if' présente dans ce fi- chier ?	
b.	A quoi sert la structure alternative 'switch' présente dans ce fichier ?	
c.	Décrire les instructions appelées dans le case `consulterLesAgences'	
d.	Retrouver où les variables \$message et \$stylePresentation sont utilisées par la suite.	
e.	A l'aide d'internet si besoin, trouver où est défini la classe CSS « text-success ». Retrouver comment cette bibio- thèque est intégrée dans le projet Netcar.	

**Bootstrap est un framework CSS**. Un framework correspond à un ensemble de librairies regroupées dans un but précis et possédant des règles internes que doivent suivre les utilisateurs.

En d'autres termes, et pour le dire très simplement, Bootstrap est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble et qu'on va pouvoir utiliser pour créer des design complexes de manière relativement simple.

Pour utiliser Bootstrap, nous allons donc simplement devoir utiliser ces fichiers. On va pouvoir faire cela de deux façons. On peut : ≻soit **télécharger les fichiers Bootstrap (CSS et JavaScript)** sur le site <u>https://getbootstrap.com/</u> puis les lier à nos fichiers HTML comme n'importe quel autre fichier CSS et JavaScript ;

>soit utiliser un CDN (Content Delivery Network ou réseau de distribution de contenu) et inclure le lien vers les fichiers dans nos fichiers HTML.

<u>C'est cette seconde solution qui a été choisie. Cependant, les fichiers ont été aussi importés pour pouvoir tester les deux méthodes de fonctionnement.</u>

7. Ouvrir, dans Netbeans, le fichier Modeles \ConnexionBD.php

- 92		php</th		
0	)	class ConnexionBD		
3	Ę.	Image: private \$connexion;		
4				
5		// constructeur de la classe BdPdo : en PHP le constructeur se nomme toujoursconstruct		
6		public functionconstruct()		
7	þ.	{		
8	白	try{		
9		<pre>\$this-&gt;connexion=new PDO('mysql:host=localhost;dbname=bdNetCar','root','');</pre>		
10		<pre>\$this-&gt;connexion-&gt;query("SET CHARACTER SET utf8");</pre>		
11		//echo " connexion réussie. ";		
12	-	}		
13	白	<pre>catch (PDOException \$erreur) {</pre>		
14		echo "Erreur de connexion à la base de données ".\$erreur->getMessage();		
15	-	}		
16	-	}		
17				
18	白	<pre>protected function getConnexion() {</pre>		
19		<pre>return \$this-&gt;connexion;</pre>		
20	-	}		
21	L	}		

Combien y a-t-il d'attributs définis dans la classe (partie a. privée) ? Comment s'appelle la méthode qui permet de construire b. objet (instance) de classe un la Est-ce que la convention de nommage de cette méthode est identique à celle vue en langage java ? Repérer l'erreur présente dans le commentaire ligne 5. c. Le corriger. Enregistrer et remarquer le marqueur bleu affiché par Netbeans. Quel est son but ? Si besoin, cliquez sur le marqueur.

d.	Rappeler le rôle de l'instruction try/catch.	
e.	Que contient la variable \$connexion ? Quelle est la classe évoquée à la ligne 9 ? Combien de paramètres sont passés lors de l'instancia- tion de le la variable \$connexion ?	
f.	Concernant la fonction getConnexion() : - quelle est sa visibilité ? Qu'est-ce que cela signifie ? - quel est son rôle ?	

8. Ouvrir, dans Netbeans, le fichier Modeles\AgenceDAO.php

- 🐕 두	p</th <th>hp</th>	hp
2	cla	ss AgenceDAO extends ConnexionBD
3 🖨	_ }	
4		//cette classe hérite de la propriété \$connexion
5		// la propriété \$connexion contient un objet PDO permettant de se connecter et d'interagir
6		// avec une base de données
7		
8		// cette classe a accès à l'accesseur getConnexion() qui retourne la propriété \$connexion
9		// grâce à cet objet \$connexion de la classe PDO, la fonction query() sera accessible
10		// la fontion query() permet de demander l'exécution d'une requête au SGBD
11		
12		
13		public function getLesAgences()
14		
15		<pre>\$req = "select * from agence order by nom asc;";</pre>
16		
17		// \$this: représente l'objet Agence dans lequel on est
18		// getConnexion() est la fonction qui retourne la propriété \$connexion
19		
20		<pre>\$resultatDeLaRequete = \$this-&gt;getConnexion()-&gt;query(\$req);</pre>
21		
22		<pre>\$tableauDesLignesResultat = \$resultatDeLaRequete-&gt;fetchAll();</pre>
23		return <pre>\$tableauDesLignesResultat;</pre>
24		}
25	• }	
26		
8	22	
	- E	Panérer et comprendre l'utilité des informations ins-
	a. r	rites à la linne 2 (anrès la nom de la classe)
ł	b. (	Quel est le rôle de la méthode getLesAgences ?
-		······································
	-	
(	c. [	Decrire les instructions qui la composent.
l I		

#### 9. Ouvrir le fichier vues\v\_consulterLesAgences.php



# 6. Représentation du mécanisme avec un diagramme de séquence (UML)



# 7. Test des fonctionnalités et observation de la modification de l'url

a) Noter les un après chaque action ci-dessous :		
Clic sur « Accueil »	http://localhost/projetphpnetcarmvc <b>/index.php</b>	
Clic sur « Consulter les Agences »		
Puis Clic sur « La nantaise »		
Retour sur « Les Agences »		
puis Clic sur « Louer un		
véhicule » <u>sur la première</u> <u>ligne</u>		

b) Pour chaque url ci-dessus, retrouver le « circuit » réalisé dans les composants du site pour déterminer le bloc d'instructions à réaliser : s'appuyer sur le code des différents fichiers appelés.

http://localhost/projetphpnetcarmvc/index.php	
index.php?controleur=gestionAgence&action=consulterLesAgences	
index.php?controleur=gestionAgence&action=consulterLesHoraires&agence=1	
index.php?controleur=gestionLocation&action=faireUneLocation&agence=1	

c) Compléter le diagramme de séquences complet des différentes fonctionnalités

#### Consulter les horaires d'une agence :

.

. . .



#### Faire une location dans une agence :



d) En fonction des fichiers déjà présents et de leur contenu, en déduire les éléments à rajouter pour que les fonctionnalités ci-dessus fonctionnent.

- => contrôleur à créer : .....
- $\Rightarrow$  actions à définir dans nouveau contrôleur : .....
- ⇒ actions à rajouter dans contrôleur existant: .....
- $\Rightarrow$  modèle DAO à rajouter : .....
- $\Rightarrow$  méthode à définir dans ce modèle DAO : ......
- $\Rightarrow$  méthodes à rajouter dans le modèle DAO existant :