



MA PREMIERE APPLICATION WEB MVC EN PHP



TA.Walid

15 rue Wattu - 6700 STRASBOURG

☎ 07 69 34 40 70 ✉ waledo.1997@hotmail.com

www.elan-formation.fr

SOMMAIRE

I. Introduction de projet cinema	Erreur ! Signet non défini.
II. Looping.....	4
III. Cahier des charges	6
1. Création du dossier de l'application	47
IV. Page index.php.....	77
V. Mise en place de la page controller php	99
1. Session	9
2. Base de données MCD \ MLD	Erreur ! Signet non défini.
3. Exemple de fonctionnalité	11
4. La Vue	17

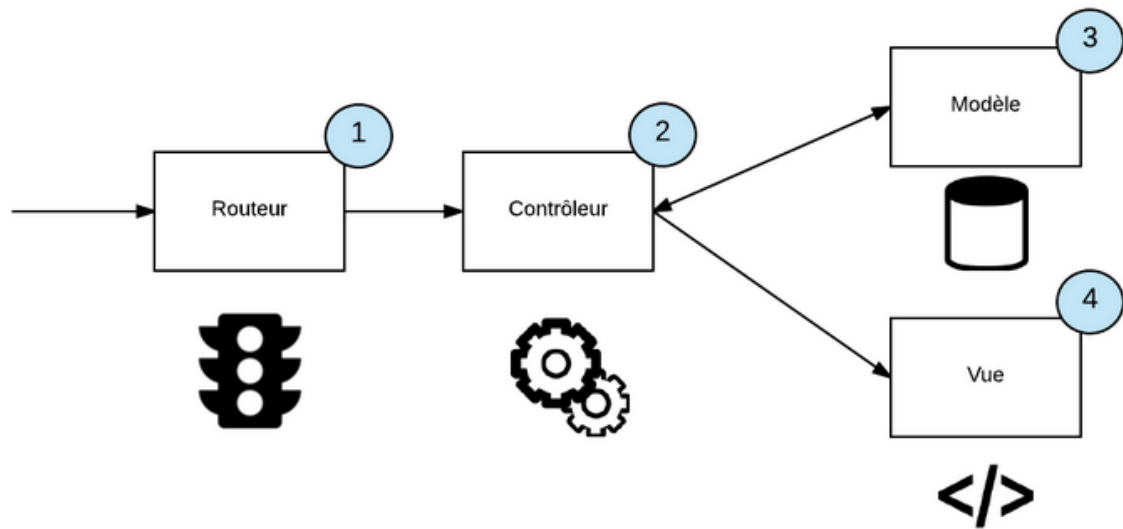
I. Introduction de projet cinéma

On va utiliser un schéma appelle MVC ce présent le chemin d'opération de nos parti et l'ordi parti jusqu'à l'affichage de résultat.

Nouvelle structure des fichiers

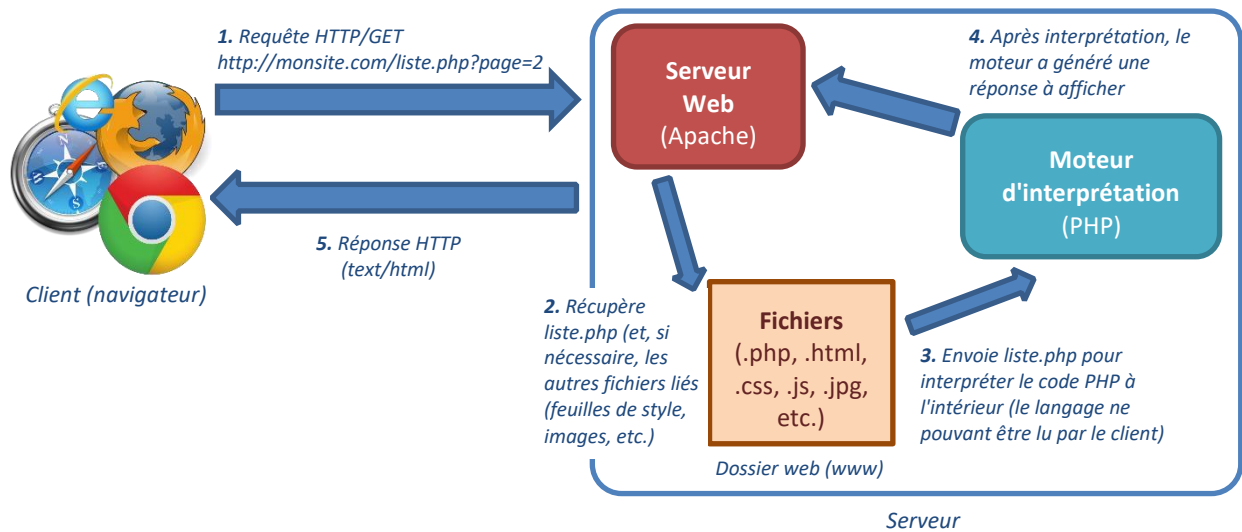


Pour faciliter la maintenance, il est plus simple de passer par un contrôleur *frontal*, qui va jouer le rôle de *routeur*. Son objectif va être d'appeler le bon contrôleur (on dit qu'il *route* les requêtes).

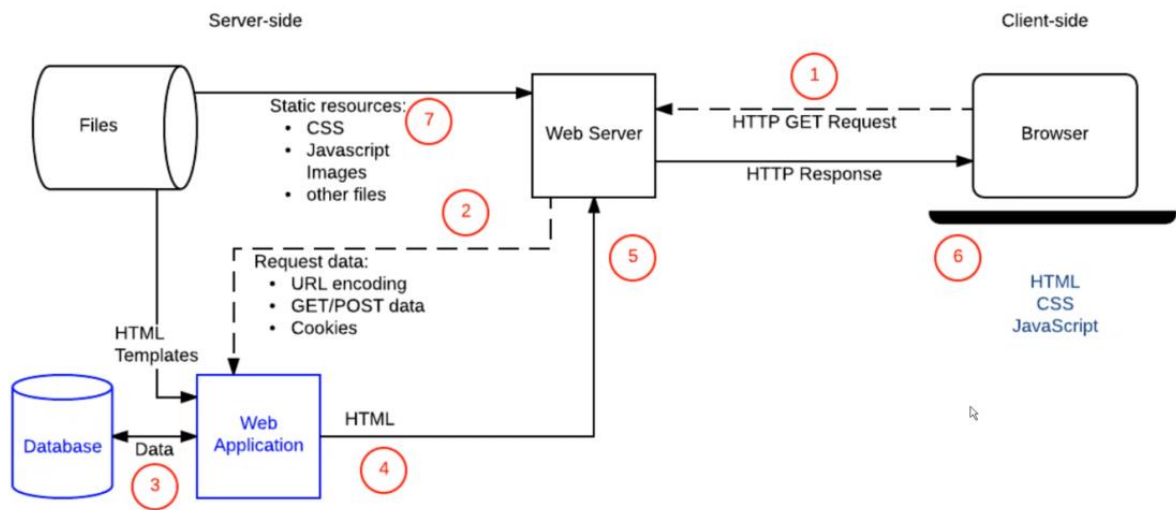


Le routeur appelle le bon contrôleur, qui appelle le modèle puis la vue

En tant que langage de script coté serveur, PHP implémente nativement diverses fonctionnalités permettant d'assurer une bonne communication des données dans un contexte client-serveur. Cette communication passe naturellement dans ce contexte par le protocole HTTP, comme des schémas l'illustre :



Un Autre Exemple :



Ainsi, le moyen de communication entre le client et le serveur (le protocole HTTP) représente en quelque sorte l'intermédiaire, l'interprète des deux parties lors de leurs échanges respectifs.

Dans l'exemple ci-dessus, le client soumet une requête au serveur qui se compose de plusieurs éléments présents dans l'URL :

<http://monsite.com/liste.php?page=2>

Le protocole employé pour la communication

Le nom de domaine du serveur sur le réseau Internet

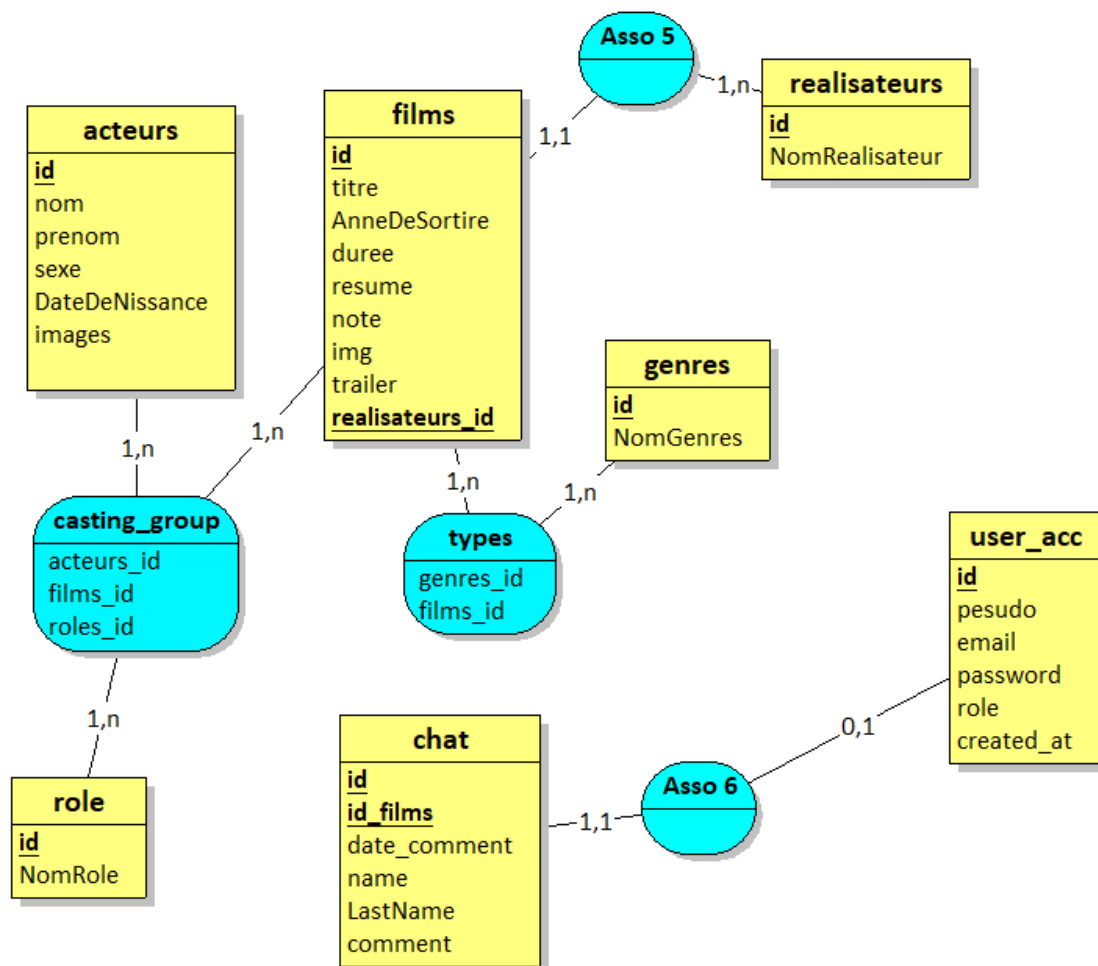
La ressource (fichier) demandée

Un paramètre de requête "page" avec pour valeur = 2

Donc on va utiliser ce type de schéma après ou moment de faire le Contrôleur.

II. Looping

Ça présente les types de communication entre des valeurs identifiées et les différentes valeurs des tableaux et ça est propre de ASSO qui permet de faire des connexions en utilisant la méthode et l'action des Contrôleur avec des index de chaque un.



Liste des connexions entre mes base de donnes :

- **Entre acteurs et casting group :** id de tableau acteur avec id des acteurs de Asso casting group pour avoir une liste asso qui permet devenir les acteurs qui sont présent aux films et chaque acteur avec son film et ça rôle
- **Entre Films et casting group :** id de tableau films avec id des films de Asso casting group pour avoir une liste asso qui permet devenir les films qui avait les acteurs et chaque acteur avec ça film et ça rôle
- **Entre rôle et casting group :** id de tableau rôle avec id des rôles de Asso casting group pour avoir une liste asso qui permet devenir les rôles qui sont présent par les acteurs et chaque acteur avec ça film et ça rôle
- **Entre Films et réalisateurs :** id de tableau réalisateur qui va asso au tableau films comme un id réalisateurs qui est un id secondaire
- **Entre Films et genres :** id de tableau genres qui va asso au tableau films comme un id genres qui est un id secondaire
- **Entre chat et user_acc :** id de tableau chat qui va asso au tableau user_acc comme un id chat qui est un id secondaire

c'est id asso va nous aide avec le base de donne et va nous aider a insère des valeurs de un tableaux de un autre

III. Cahier des charges

Dans la suite de ce cours, nous allons créer pas-à-pas une petite application web dont le cahier des charges est celui-ci :

L'application doit permettre à un utilisateur de renseigner différents produits par le biais d'un formulaire

Trois types de pages sont nécessaires à cela :

1. index.php

Présentera un page index comme un page permettant de renseigner plusieurs page au navigation par chaque un de ces pages avec ça index en Controller qui permette nous envoie dans un vu le schéma qui Jai présente avant (MVC) c'est chaque Controller avec sa propre vu et si on veut on peut ajouter un 2em vu comme en line qui permet nous envoie a un deuxième page de vu :

Je vais mettre tous les Screen shot étape par étape en dessus

2. Controller de chaque page avec ça manager et page de classe d'élément avec sa entité

On doit y connecter avec MySQL aux bases des données traitera la requête de base des données

En utilisant class mère **AbstractManager** qui fait plusieurs actions de fonction

Qui permet faire une requête ou getter des résultat etc...

Par contre on doit connecte de notre class (ex : Acteur manager) de ca class mère AbstractManager qui va exécuter cette déférent fonction et envoie le déférent résultat a la fonction index() qui permette envoie les résultat a la Controller qui (compare le type de donnees et combien existe il des paramètre des valeurs etc..) contrôle le base de donnees par le résultat attendus et le envoie oui tu passes a la vue ou non il y existe un **fatal error ou error 404** bien sûr il existait plusieurs **types de error** et par fois on fait un message de error nous même au envoie le client a un autre page.

3. vu de chaque page

Affichera toutes les valeurs des tableaux (sont dans Looping)

Et qui sont enregistré en session (et en détail) et présentera le total général de tous notre élément ajouté.

Et par fois permette affiche un Link qui permette nous envoie à l'autre page qui affiche par Example : les détail des donnees

1. Création du dossier de l'application

En fonction de la solution de développement que vous utilisez (WAMP, MAMP, Laragon...), **créons un dossier "cinema" dans notre serveur web**, au sein du dossier "www" ou "htdocs" selon cas.

Créons ensuite des fichiers travail.

IV. Page index.php


Voici le code de la page *index.php*, contenant le formulaire qui nous permettra de renseigner les produits à l'application :

```
1  <?php
2      namespace App;
3      use App\Router;
4
5      require_once "app\Autoloader.php";
6      Autoloader::register();
7
8      session_start();
9
10     define('DS', DIRECTORY_SEPARATOR);
11     define('ROOT_DIR', '..'.DS); // "../"
12     define('PUBLIC_PATH', ROOT_DIR.'public'.DS); // "../public/"
13     define('CSS_PATH', PUBLIC_PATH.'css'.DS); // "../public/css/"
14     define('IMG_PATH', PUBLIC_PATH.'img'.DS); // "../public/img/"
15     define('JS_PATH', PUBLIC_PATH.'js'.DS); // "../public/js/"
16
17     define('SERVICE_PATH', ROOT_DIR.'app'.DS); // "../app/"
18     define('CTRL_PATH', ROOT_DIR.'controller'.DS); // "../controller/"
19     define('VIEW_PATH', ROOT_DIR.'view'.DS); // "../view/"
20
21     define('ADMIN_MAIL', "admin@mvc.com"); //mail de l'administrateur
22
23
24     $result = Router::handleRequest($_GET);
25
26     \ob_start();
27
28     include $result['view'];
29     $page = ob_get_contents();
30
31     \ob_end_clean();
32
33     include VIEW_PATH."layout.php";
```

Quelques précisions concernant certaines lignes de ce code :

↳ Ligne 10 au 21 : qui définissent le chemin absolu des fichiers déferent

 **action** (qui indique la cible du formulaire, le fichier à atteindre lorsque l'utilisateur soumettra le formulaire) et **l'action corresponde a la id**

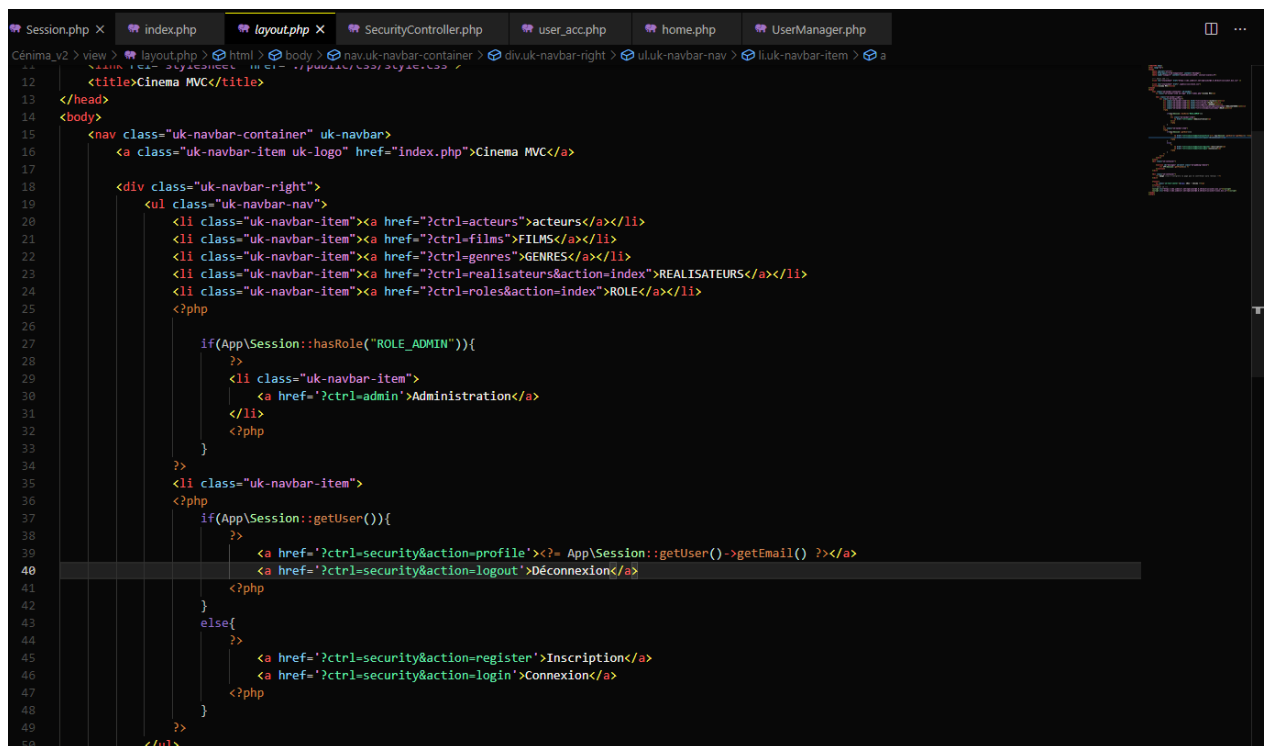
 **method** (qui précise par quelle méthode HTTP les données du formulaire seront transmises au serveur)

http://localhost/C%c3%a9nima_v2/index.php?ctrl=films

http://localhost/C%c3%a9nima_v2/index.php?ctrl=films&action=voirFilm&id=7

Lignes 26 au 31:

Un page (notre page index) qui contient un autre page (layout) qui nous envoie à la vue correspondent de notre index)



```
12 <title>Cinema MVC</title>
13 </head>
14 <body>
15 <nav class="uk-navbar-container" uk-navbar>
16 <a class="uk-navbar-item uk-logo" href="index.php">Cinema MVC</a>
17
18 <div class="uk-navbar-right">
19 <ul class="uk-navbar-nav">
20 <li class="uk-navbar-item"><a href="?ctrl=acteurs">acteurs</a></li>
21 <li class="uk-navbar-item"><a href="?ctrl=films">FILMS</a></li>
22 <li class="uk-navbar-item"><a href="?ctrl=genres">GENRES</a></li>
23 <li class="uk-navbar-item"><a href="?ctrl=realisateurs&action=index">REALISATEURS</a></li>
24 <li class="uk-navbar-item"><a href="?ctrl=roles&action=index">ROLE</a></li>
25 <?php
26
27 <?php
28 <li class="uk-navbar-item">
29 <a href="?ctrl=admin">Administration</a>
30 </li>
31 <?php
32
33 <?php
34
35 <li class="uk-navbar-item">
36 <?php
37 if(App\Session::getUser()){
38 <a href="?ctrl=security&action=profile">? App\Session::getUser()->getEmail() ?></a>
39 <a href="?ctrl=security&action=logout">Déconnexion</a>
40 <?php
41
42 }
43 else{
44 <a href="?ctrl=security&action=register">Inscription</a>
45 <a href="?ctrl=security&action=login">Connexion</a>
46 <?php
47
48 }
49
50 </ul>
51 </div>
52 </nav>
53 </body>
54 </html>
```

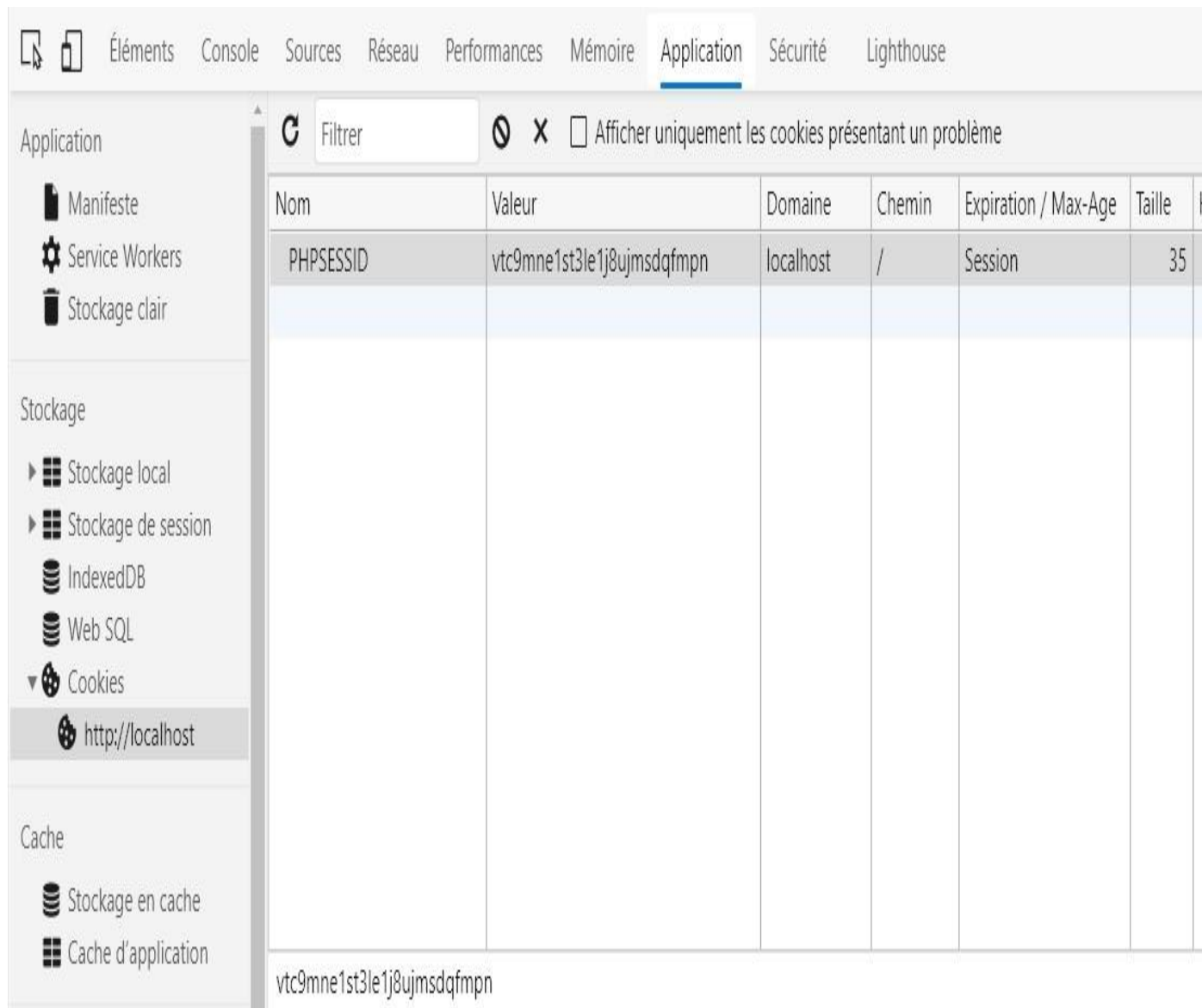
Ça définir déferent navigation de par qui nous envoie à notre méthode utilise par nous pour que le navigateur nous envoie *à la vue correspondent de notre index*

V. Mise en place de la page Controller PHP

1. Session

le serveur enregistrera un cookie PHPSESSID dans le navigateur client, contenant l'identifiant de la session appartenant à celui-ci. Les cookies sont transmis au serveur avec chaque requête HTTP effectuée par le client.

La capture suivante présente le cookie en question (onglet *Application* du DevTools de Google Chrome) :



La capture suivante présente mon code de App\session qui permet prendre en charge de class session :

```

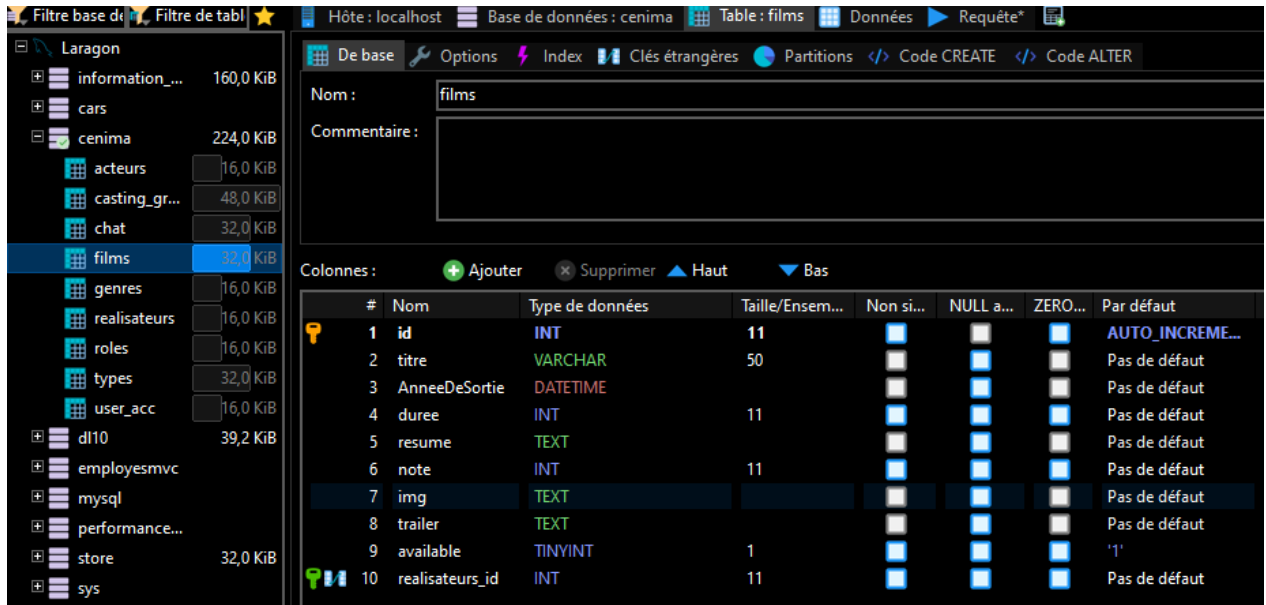
1  <?php
2  namespace App;
3  use model\Entity\User;
4  use model\Entity\Films;
5
6  abstract class Session
7  {
8      public static function getFlashes(){
9          if(isset($_SESSION['messages'])){
10             var_dump($_SESSION['messages']);
11             unset($_SESSION['messages']);
12         }
13     }
14
15     public static function addFlash($type, $message){
16         if(!isset($_SESSION["messages"])){
17             $_SESSION["messages"] = [];
18         }
19         if(!isset($_SESSION["messages"][$type])){
20             $_SESSION["messages"][$type] = [];
21         }
22         $_SESSION["messages"][$type][] = $message;
23     }
24     public static function getUser(){
25         if(isset($_SESSION["user"])){
26             return $_SESSION["user"];
27         }
28         return null;
29     }
30
31     /**
32      * Put the user param in session (aka. connect the user in the app)
33      *
34      * @param User $user the user object to put in session
35      * @return void
36      */
37     public static function setUser($user){
38         $_SESSION["user"] = $user;
39     }
40
41     public static function removeUser(){
42         unset($_SESSION["user"]);
43     }
44
45     public static function hasRole($role){
46         return isset($_SESSION["user"]) && $_SESSION["user"]->getRole() === $role;
47     }
48
49 }

```

2. Base de données MCD \ MLD

La 2 capture suivante présente HeidiSQL qui est notre base des données de notre serveur :

- Les paramètres



- Les valeurs :

#	id	titre	AnneeDeSortie	duree	resume	note	img	trailer
1	1	Transformers	2007-07-25 00:00:00	144	Une guerre sans merci oppose ...	3	https://static1.purebreak.com/a...	https://www.you
2	2	Le Transporteur 3	2008-09-26 00:00:00	104	Frank Martin est le spécialiste in...	3	https://encrypted-tbn0.gstatic.c...	https://www.you
3	3	Casino Royale	2006-11-14 00:00:00	145	Pour sa première mission, Jame...	4	https://ae01.alicdn.com/kf/HTB...	https://www.you
4	4	Taxi 3	2003-01-29 00:00:00	90	C'est décembre à Marseille. Da...	2	https://encrypted-tbn0.gstatic.c...	https://www.you
5	5	Thor : Ragnarok	2017-10-25 00:00:00	130	Privé de son puissant marteau, ...	4	https://encrypted-tbn0.gstatic.c...	https://www.you
6	6	Avengers : Endgame	2019-04-24 00:00:00	182	Le Titan Thanos, ayant réussi à...	4	https://d1fmx1rbmqxr.cloudfr...	https://www.you
7	7	Argo	2012-10-07 00:00:00	130	Le 4 novembre 1979, au summ...	4	https://images-na.ssl-images-a...	https://www.you

- La 2 capture suivante présente Mon code qui est notre base des données de notre serveur et comment on connecte au serveur et c'est quoi le Controller :
- De ↩ Ligne 6 au 20 : connexion a la base des données
- De ↩ Ligne 30 au 39 : faire une requête : permette de construire une requête qui contentent les données et d'obtenir notre résultat
- De ↩ Ligne 40 au 70 : c'est la partie qui donne les résultats.

```
Session.php AbstractManager.php X # style.css voir.php user_acc.php home.php UserManager.php
C nima_v2 > app > AbstractManager.php > PHP Intelephense > AbstractManager > getOneOrNullResult
1 <?php
2 namespace App;
3
4 abstract class AbstractManager
5 {
6     protected static $bdd;
7
8     protected function connect(){
9         //se connecter   MySQL
10         self::$bdd = new \PDO(
11             "mysql:host=localhost:3306;dbname=cinema",
12             "root",
13             "",
14             [
15                 \PDO::ATTR_ERRMODE => \PDO::ERRMODE_EXCEPTION,
16                 \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_ASSOC,
17                 \PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8'
18             ]
19         );
20     }
21
22     /**
23      * Permet d'effectuer une requ te   la base de donn es et d'en obtenir son r sultat
24      *
25      * @param string $query - la requ te SQL   ex cuter
26      * @param array|null $params - les param tres de la requ te si besoin
27      *
28      * @return \PDOStatement l'objet PDOStatement relatif   l'ex cution de la requ te
29      */
30     protected static function makeQuery($query, $params = null){
31         if($params){
32             $statement = self::$bdd->prepare($query);
33             $statement->execute($params);
34         }
35         else{
36             $statement = self::$bdd->query($query);
37         }
38
39         return $statement;
40     }
41 }
```

```
43     protected function getResult($classname, $query, $params = null){
44         $stmt = self::makeQuery($query, $params);
45         $results = [];
46         foreach($stmt->fetchAll() as $data){
47             $results[] = new $classname($data);
48         }
49
50         return $results;
51     }
52
53     /**
54      * R cup re un objet de la classe sp cifi e ou null
55      *
56      * @param string $classname - la classe de l'objet voulu
57      * @param string $query - la requ te SQL   ex cuter
58      * @param array|null $params - les param tres de la requ te si besoin
59      *
60      * @return Object|null l'objet de la classe attendue ou null
61      */
62     protected function getOneOrNullResult($classname, $query, $params = null){
63         $stmt = self::makeQuery($query, $params);
64         if($data = $stmt->fetch()){
65             return new $classname($data);
66         }
67         return null;
68     }
69
70 }
```

3. Exemple de fonctionnalité

Comment on Affiche la liste des Films ?

- Etape 1 :
On connecte en base des données comme est déjà expliqué en haut AbstractManager
Et le requête c'est déjà prise comme J'ai expliqué avec la fonction makeQuery () et getResult () ou getOneOrNullResult()
Notre résultat est préparé pour être prêt à être utilisé et il est retourné de manière class
- Etape 2:
Les résultats des Films sont déjà envoyés en ControllerFilms
Une astuce pour lire le code en dessus :
- ↪ Ligne 7 au 10 : renvoie à manager qui dit ok votre résultat est correct pour afficher
- ↪ Ligne 15: prendre de résultat (ici List de films) avec l'envoi à manager qui dit ok résultat prêt et sont trouvés all => Avec la fonction findAll().
- ↪ Ligne 13 au 21: c'est index de de films qui prend ce résultat (List de films) et renvoie à la vue
- La 2 capture suivante présente mon code ControllerManager :
↪ Ligne 9 au 11: connecte à base de données avec l'aide de AbstractManager qui est parent de la classe FilmsManager
- ↪ Ligne 13 au 21: prendre de résultat (ici List de films) => Avec la fonction findAll() et comparer avec de la classe entity que on a installé ,
Avec cette classe et ça getter et setter on peut installer les données de notre parti comme on veut et comparer avec la base de données par manager et renvoie à la vue.
- ↪ Ligne 13 au 21: c'est index de de films qui prend ce résultat (List de films) et renvoie à la vue
- ↪ Ligne 23 au 37: c'est une fonction qui permet de prendre le résultat de manager par son id et le rendre à la vue.

```

1  <?php
2  namespace Controller;
3
4  use App\AbstractController;
5  use Model\Manager\FilmsManager;
6
7  class FilmsController extends AbstractController
8  {
9      public function __construct(){
10         $this->manager = new FilmsManager();
11     }
12
13     public function index()
14     {
15         $films = $this->manager->findAll();
16
17         return $this->render("films/home.php", [
18             "films" => $films,
19             "title"  => "Liste des Films"
20         ]);
21     }
22
23     public function voirFilm($id)
24     {
25         if($id){
26
27             $Film = $this->manager->findOneById($id);
28             $casting = $this->manager->findcastingbyfilm($id);
29             $Type = $this->manager->findTypebyfilm($id);
30
31             return $this->render("films/voir.php", [
32                 "film" => $Film,
33                 "title" => $Film,
34                 "casting" => $casting,
35                 "Type" => $Type
36             ]);
37         }
38         else $this->redirectToRoute("films", "index");
39     }

```

- Manager :
- La 4 capture suivante présente Mon code mdl (model)

model

entity

acteurs.php

casting_group.php

films.php

genres.php

realisateurs.php

roles.php

types.php

user_acc.php

manager

ActeursManager....

casting_groupMa...

FilmsManager.php

GenresManager.p...

RealisateursMana...

RolesManager.php

TypesManager.php

UserManager.php

```

1 <?php
2 namespace Model\Entity;
3 use App\AbstractEntity;
4
5 class Films extends AbstractEntity
6 {
7     private $id;
8     private $titre;
9     private $AnneeDeSortie;
10    private $duree;
11    private $resume;
12    private $note;
13    private $realisateurs;
14    private $roles;
15    private $genres;
16    private $acteurs;
17    private $img;
18    private $star;
19    private $trailer;
20    private $Available;
21    public function __construct($data){
22        parent::hydrate($data, $this);
23    }
24    /**
25     * Get the value of id
26     */
27    public function getId()
28    {
29        return $this->id;
30    }
31
32    /**
33     * Set the value of id
34     * @return self
35     */
36    public function setId($id)
37    {
38        $this->id = $id;
39    }

```

```

1 <?php
2 namespace Model\Manager;
3 use App\AbstractManager;
4 use Model\Entity\Films;
5
6
7 class FilmsManager extends AbstractManager
8 {
9     public function __construct(){
10        parent::connect();
11    }
12
13
14    public function findAll(){
15        return $this->getResults(
16            "Model\Entity\Films",
17            "SELECT * FROM films ORDER BY titre"
18        );
19    }
20
21    public function findOneById($id){
22        return $this->getOneOrNullResult(
23            "Model\Entity\Films",
24            "SELECT * FROM films WHERE id = :num",
25            [
26                "num" => $id
27            ]
28        );
29    }
30    public function findcastingbyfilm($id){
31        return $this->getResults(
32            "Model\Entity\Casting_group",
33            "SELECT * FROM Casting_group WHERE films_id = :num",
34            [
35                "num" => $id
36            ]
37        );
38    }

```

```

39     public function findTypebyfilm($id){
40         return $this->getResults(
41             "Model\Entity\Genres",
42             "SELECT g.id as id, NomGenres FROM types t, genres g WHERE t.genres_id= g.id AND films_id = :num",
43             [
44                 "num" => $id
45             ]
46         );
47     }
48
49     public function insertFilms($titre, $annee_de_sortie , $duree, $resume, $note){
50         return $this->executeQuery(
51             "INSERT INTO films (titre, annee_de_sortie , duree, resume, note) VALUES (:titre, :annee_de_sortie , :duree, :resume,:note)",
52             [
53                 "titre" => $titre,
54                 "annee_de_sortie" => $annee_de_sortie,
55                 "duree" => $duree,
56                 "resume" => $resume,
57                 "note" => $note,
58             ]
59         );
60     }
61
62     public function deleteFilms($id){
63         return $this->executeQuery(
64             "DELETE FROM films WHERE id = :id",
65             [
66                 "id" => $id
67             ]
68         );
69     }
70 }

```

Avec ManagerFilms:

Qqe astuce pour lire le code en haut :

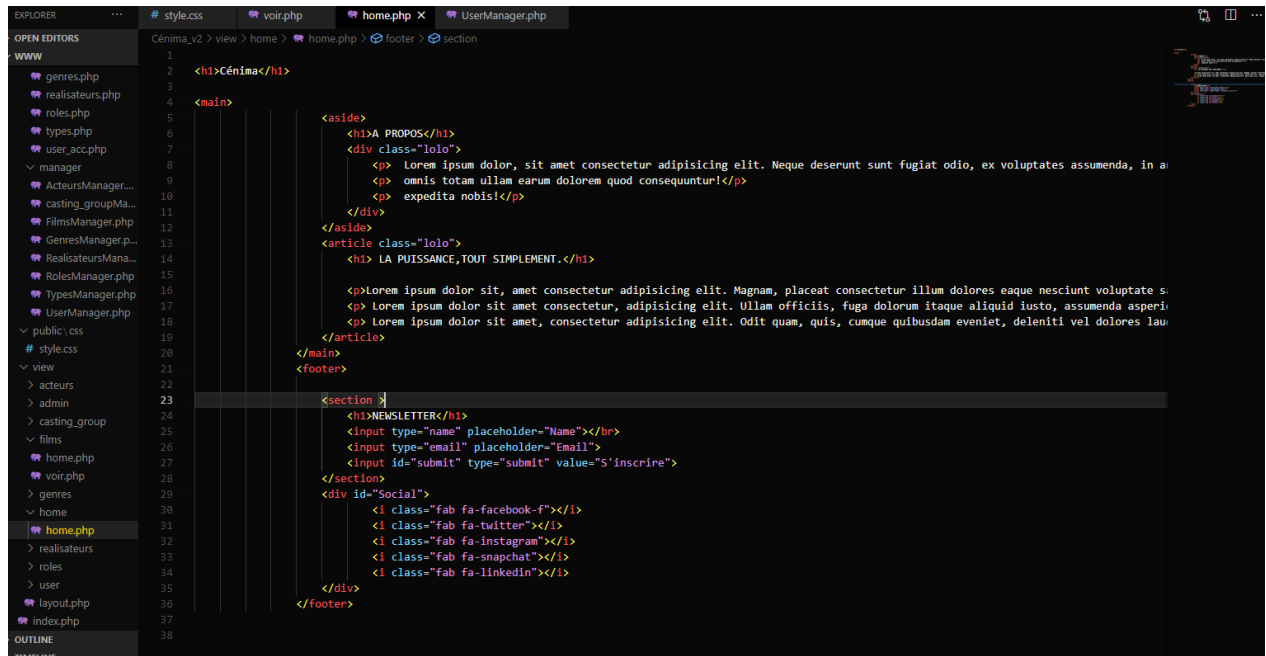
Pour ManagerFilms le deux dernier Image on a :

- ➤ Ligne 9 au 11: connecte a base de donnees avec l'aide de AbstractManager qui est parant de la classe FilmsManager
- ➤ Ligne 14 au 19: prendre de resultat (ici List de films) => Avec le fonction findAll() en prendre tout les films et on l'envoie a la affichage de un lise trier par ca titre de a -z
- ➤ Ligne 21 au 29: c'est un fonction pour films qui prendre cette resultat (List de films) et les donnees par ca id (un resultat ici que 1 films et ça donne) et renvoie à la 2em vue qui représente ça.
- ➤ Ligne 39 au 47: c'est un fonction pour films qui prendre cette resultat (List de films) et le donne par ca id (un resultat ici que 1 films et ça donne) et renvoie à la 2em vue qui représente resultat de 1 films avec un donnees dedans qui permet dis voilà le genres qui est arrive de notre base de donnees et qui est en point commun avec cette film =>le deux tableaux de films et genres sont Asso (si vous réfléchir au MDL).
- ➤ Ligne 49 au 69 : deux fonction 1 pour insère le films et 1 pour le supprime .

4. vu

Avant de commence la vue de notre Example : Films

Je veux montre le page home qui est très simple au niveaux code et affichage .



```
1 <h1>C nima</h1>
2
3 <main>
4
5 <aside>
6 <h2>A PROPOS</h2>
7 <div class="lolo">
8 <p> Lorem ipsum dolor, sit amet consectetur adipisicing elit. Neque deserunt sunt fugiat odio, ex voluptates assumenda, in a
9 <p> omnis totam ullam earum dolorem quod consequuntur!</p>
10 <p> expedita nobis!</p>
11 </div>
12 </aside>
13 <article class="lolo">
14 <h2> LA PUISSANCE, TOUT SIMPLEMENT.</h2>
15 <p> Lorem ipsum dolor sit, amet consectetur adipisicing elit. Magnam, placeat consectetur illum dolores eaque nesciunt voluptate s
16 <p> Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ullam officiis, fuga dolorum itaque aliquid iusto, assumenda asperi
17 <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odit quam, quis, cumque quibusdam eveniet, deleniti vel dolores lau
18 </article>
19 </main>
20 <footer>
21
22 <section>
23 <h3>NEWSLETTER</h3>
24 <input type="text" placeholder="Name"><br>
25 <input type="email" placeholder="Email">
26 <input id="submit" type="submit" value="S'inscrire">
27 </section>
28 <div id="Social">
29 <i class="fab fa-facebook-f"></i>
30 <i class="fab fa-twitter"></i>
31 <i class="fab fa-instagram"></i>
32 <i class="fab fa-snapchat"></i>
33 <i class="fab fa-linkedin"></i>
34 </div>
35 </footer>
```



- En passe a notre Exemple :

```

1  <?php
2  $films = $result['data']['films'];
3  ?>
4
5  <h1>Films</h1>
6
7  <main id="films-list">
8      <table class="uk-table uk-table-striped">
9          <thead>
10             <tr>
11                 <th>Titre</th>
12                 <th>Annee_de_sortie</th>
13                 <th>Duree </th>
14                 <th>Resume </th>
15                 <th>Note </th>
16                 <th>Realisateurs</th>
17                 <th>supr&add</th>
18             </tr>
19         </thead>
20         <tbody>
21
22             <?php
23             foreach ($films as $film) { ?>
24                 <tr>
25                     <td><a href="?ctrl=films&action=voirFilm&id=?= $film->getId() ?"><?=$film->getTitre() ?></a></td>
26                     <td><?=$film->getAnneeDeSortie() ?></td>
27                     <td><?=$film->getDuree() ?></td>
28                     <td><?=$film->getResume() ?></td>
29                     <td class="uk-icon-link"> Note :<?=$film->getStar() ?></td>
30                     <td><?=$film->getRealisateurs() ?></td>
31                     <td>
32                         <a href="#" class="uk-icon-link uk-icon="trash"></a>
33                         <a href="#" class="uk-icon-link uk-margin-small-right" uk-icon="file-edit"></a>
34                     </td>
35                 </tr>
36             </td>
37             </tbody>
38         </table>
39     </main>

```

Dans ce code on a :

- un table de nav-bar ➡ Ligne 8 au 18.
- ➡ Ligne 22 au 36: un blucle qui permete compte le films et chaque un des cest film et donne nous td qui est utilise comme un bloque de tableaux ici et affiche le :
 - Le nom de film .
 - Date de sortie .
 - Le resume .
 - Le note par Star (note c'est un nombre on crée fonction dans notre entité de films ((on a dit comme on veut afficher si veut reficher)) et l'affiche comme un start .
 - Le réalisateur .
 - Et modifie et trash qui modifie le bloque ou supprime (Dans mon code ca affiche sans fonctionne désolé)

Et ca donne cette type de affichage

Casino Royale



realisateur : Martin Campbell

Titre : Casino Royale

Année de sortie: 14-11-2006

Durée :145

avis: ☆☆☆☆☆

Trailerlink: "https://www.youtube.com/watch?v=36mnx8dBbGE"

Forbidden

You don't have permission to access
/C/Cinema_v2?url=http://www.youtube.com/watch?v=36mnx8dBbGE
on this server.

Genres : Action/Aventure

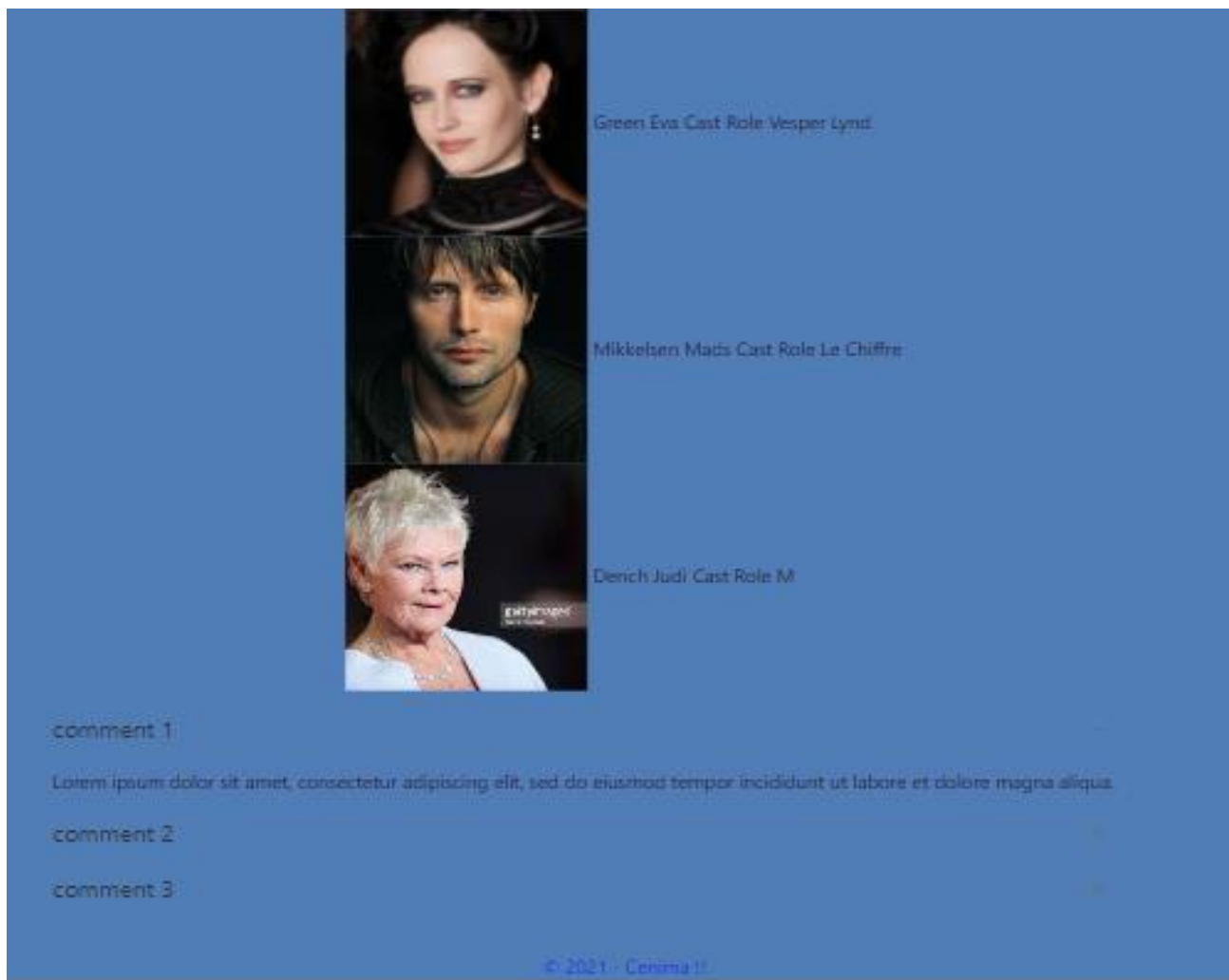
Casting



Craig Daniel Cast Role James Bond



Green Eva Cast Role Vesper Lynd



Dans Mon site il y a :

- Nav bar tout en haut qui contient le lien de coction ca deviens de l'index qui en commun avec tout les autre page ex : CinémaMVC (Page home), acteur, film, genres , rôle , etc..
- Titre de Films H1
- Image de base qui présent le film
- Nom réalisateur
- Anne de sortir
- La durée en minute
- Le note par star
- Trailer Link
- Trailer vidéo (dans mon site il donne pas l'accès au YouTube mais le Link fonction
- List de genres ,List de casting qui permette affiche les acteurs présente par une image et le nom et il caste le rôle qui lui jeu, aussi le commentaire pas arrive a avoir le temps pour bounes+ , il y a aussi le Footer qui dire Elan ou dans mon site 2021 cinéma !!

Films

TITRE	ANNEE_DE_SORTIE	DUREE	RESUME	NOTE	REALISATEURS	SUPR&ADD
Argo	07-10-2012	130	Le 4 novembre 1979, au summum de la révolution iranienne, des militants envahissent l'ambassade américaine de Téhéran, et prennent 52 Américains en otage. Au milieu du chaos, six Américains réussissent à s'échapper et à se réfugier au domicile de l'ambassadeur canadien. Sachant qu'ils seront inévitablement découverts et probablement tués, un spécialiste de l'exfiltration de la CIA du nom de Tony Mendez monte un plan risqué visant à les faire sortir du pays.	Note : ☆☆☆☆☆	ben Affleck	🗑️ ✎️
Avengers : Endgame	24-04-2019	182	Le Titan Thanos, ayant réussi à s'approprier les six Pierres d'Infinité et à les réunir sur le Gantelet doré, a pu réaliser son objectif de pulvériser la moitié de la population de l'Univers. Cinq ans plus tard, Scott Lang, alias Ant-Man, parvient à s'échapper de la dimension subatomique où il était coincé. Il propose aux Avengers une solution pour faire revenir à la vie tous les êtres disparus, dont leurs alliés et coéquipiers : récupérer les Pierres d'Infinité dans le passé.	Note : ☆☆☆☆☆	Anthony Russo	🗑️ ✎️
Casino Royale	14-11-2006	145	Pour sa première mission, James Bond affronte le tout-puissant banquier privé du terrorisme international, Le Chiffre. Pour achever de le ruiner et démanteler le plus grand réseau criminel qui soit, Bond doit le battre lors d'une partie de poker à haut risque au Casino Royale. La très belle Vesper, attachée au Trésor, l'accompagne afin de veiller à ce que l'agent 007 prenne soin de l'argent du gouvernement britannique qui lui sert de mise, mais rien ne va se passer comme	Note : ☆☆☆☆☆	Martin Campbell	🗑️ ✎️

Le Transporteur 3	26-09-2008	104	Frank Martin est le spécialiste incontesté des livraisons très spéciales et très risquées. Menacé de mort, il doit cette fois-ci transporter deux grands sacs ainsi qu'une jeune femme d'origine ukrainienne depuis Marseille jusqu'à Odessa. Il ignore tout du contenu des sacs et de l'identité de sa passagère.	Note : ☆☆☆☆☆	Olivier Megaton	🗑️ ✎️
Taxi 3	29-01-2003	90	C'est décembre à Marseille. Daniel rajoute des options à son taxi et Lili voit rouge. Le commissaire Gilbert est aveuglé par Qiu, sa stagiaire japonaise, et son enquête piétine. Emilien voit des pères Noël partout et Petra s'impatiente. De Marseille à Tignes, de la Canebière aux pistes de ski, l'aventure peut commencer.	Note : ☆☆☆☆☆	Gérard Krawczyk	🗑️ ✎️
Thor : Ragnarok	25-10-2017	130	Privé de son puissant marteau, Thor est retenu prisonnier sur une lointaine planète aux confins de l'univers. Pour sauver Asgard, il va devoir lutter contre le temps afin d'empêcher l'impitoyable Hela d'accomplir le Ragnarök, ou la destruction de son monde et la fin de la civilisation asgardienne. Pour y parvenir, il va d'abord devoir mener un combat titanesque de gladiateurs contre celui qui était autrefois son allié au sein des Avengers : l'incroyable Hulk	Note : ☆☆☆☆☆	Taika Waititi	🗑️ ✎️
Transformers	25-07-2007	144	Une guerre sans merci oppose depuis des temps immémoriaux deux races de robots extraterrestres : les Autobots et les cruels Decepticons. Son enjeu : la maîtrise de l'univers. Dans les premières années du 21ème siècle, le conflit s'étend à la Terre, et le jeune Sam Witwicky devient, à son insu, l'ultime espoir de l'humanité. Semblable à des milliers d'adolescents, Sam n'a connu que les soucis de son âge : le lycée, les amis, les voitures, les filles.	Note : ☆☆☆☆☆	Michael Bay	🗑️ ✎️

Alors on a encore le code notre vue

```

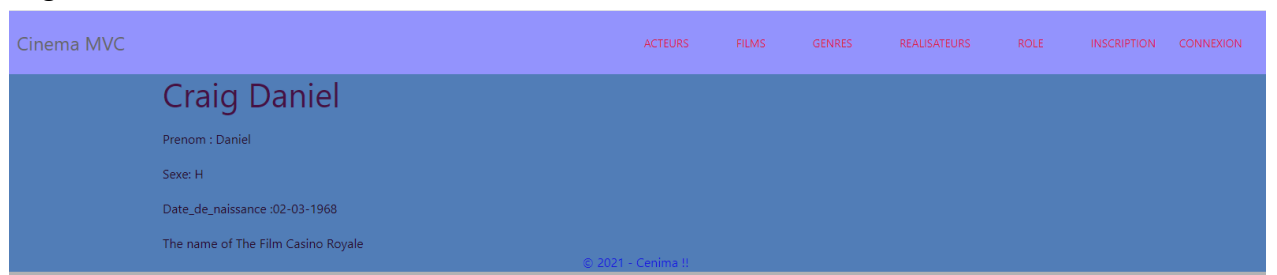
1 <?php
2 namespace App;
3 use Model\Entity\Types;
4
5 $film = $result['data']['film'];
6 $casting = $result['data']['casting'];
7 $Type = $result['data']['Type'];
8
9 <body>
10
11 <h1><?=$film->getTitre() ?></h1>
12 <img class="img" src=<?=$film->getIng() ?>" alt="img">
13 <p>realisateur : <?=$film->getRealisateurs() ?></p>
14
15 <p>Titre : <?=$film->getTitre() ?></p>
16 <p>Annee_de_sortie: <?=$film->getAnneeDeSortie() ?></p>
17 <p>Duree :<?=$film->getDuree() ?></p>
18 <p class="uk-icon-link"> Note :<?=$film->getStar() ?></p>
19 <p class="Type">
20 <p>TrailerLink:<a href=<?=$film->getTrailer() ?>" > "<?=$film->getTrailer() ?>"</a></p>
21
22 <iframe src="url(<?=$film->getTrailer() ?>)" frameborder="0" width="320" height="240" controls></iframe>
23 <p>
24 <?php
25
26     foreach ($Type as $t) { ?>
27         Genres : <?=$t ?>
28
29         <?php ?>
30     }
31 </p>
32 <h1 id="cast">Casting </h1>
33 <div class="content">
34
35 <p>
36
37 <?php
38
39     foreach ($casting as $c) { ?>
40         <?php ?>

```

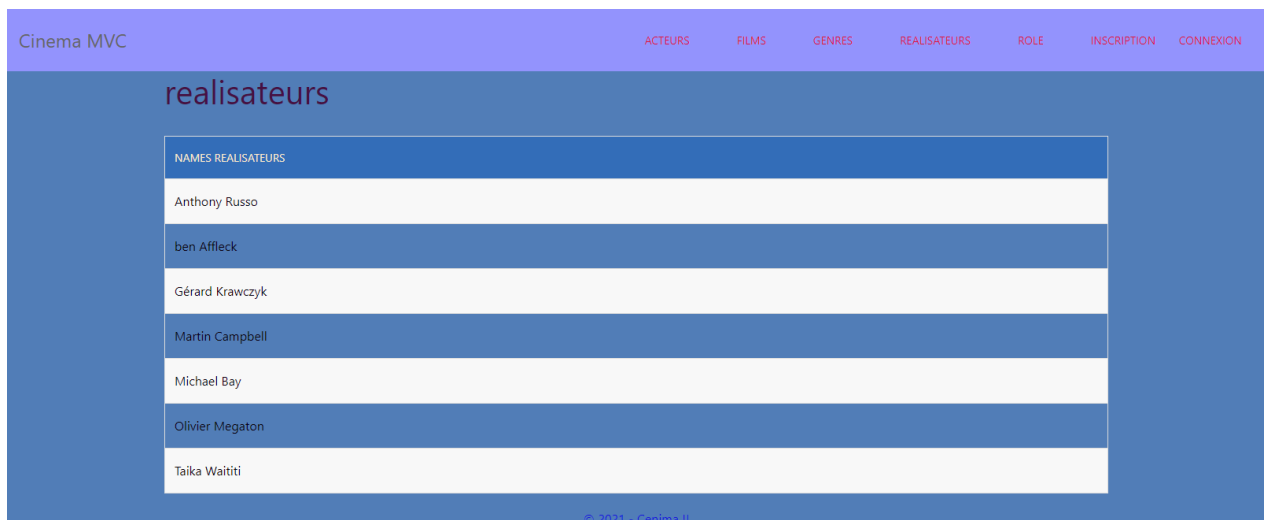
C'est fait au même manier du home. PHP de film
 Et voilà comme ça on a passée au site de la liste de film avec ça code.
 Et le deuxième code de vue qui est dans mon site au moment vous Click le Titre dans mon site de List de films home. PHP ça vas vous envoie au 2em vue.
 En va passe a quelle que page comme Exemple sans explication :



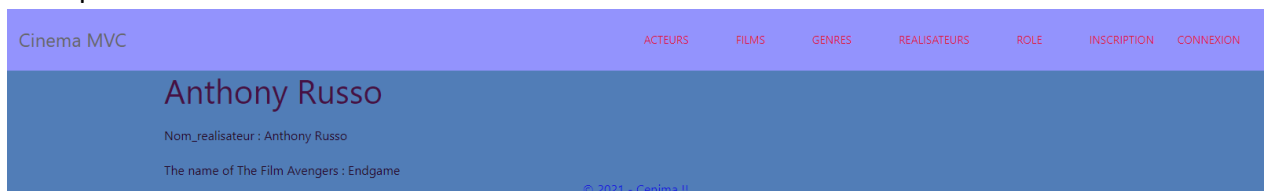
Page de acteur



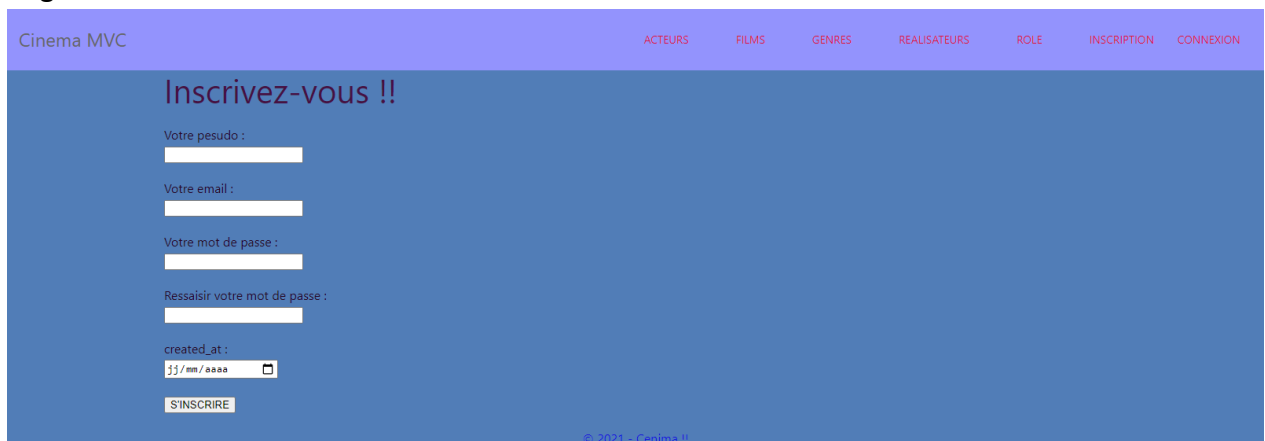
Page de réalisateurs : donne un List de réalisateurs comme page de films avec le titre que on click pour accède au information détaillée .



Exemple vue 2 :



Page insertion :



Page de connexion :

