

# Choisir son framework

Laurence  
X  NATIS



Compétence demandée :  
Savoir choisir son Framework

1. Popularité
2. Durabilité
3. Sécurité
4. Documentation
5. Communauté

# POPULARITE

Choisissez le framework bien connu,  
**reconnu, complet**, qui comprend les  
bonnes idées et surtout qui **comprend**  
**les fonctionnalités voulues.**

**DURABILITE**

Lors du choix d'un framework, assurez-vous qu'il pourra vous **suivre pendant toute la durée du projet**. Cela simplifie à la fois **la maintenance et la mise à niveau** de votre application.

**SECURITE**



Tout en choisissant un framework pour minimiser les risques, assurez-vous qu'il est **capable d'assurer des fonctions de sécurité**. Les frameworks non mis à jour peuvent présenter de gros problèmes de sécurité.

# DOCUMENTATION

Une documentation bien expliquée et détaillée vous permet de dépasser les limites technologiques que vous pouvez rencontrer.

# COMMUNAUTE

Choisissez le framework qui dispose d'une communauté amicale qui aide les nouveaux développeurs sur la plate-forme.

La communauté est une ressource exploitable qui liste de nombreux problèmes que vous serez peut-être amenés à rencontrer.

# Exemples

Laurence  
X  NATIS



Compétence demandée :  
Connaître des exemples de  
bibliothèques et frameworks

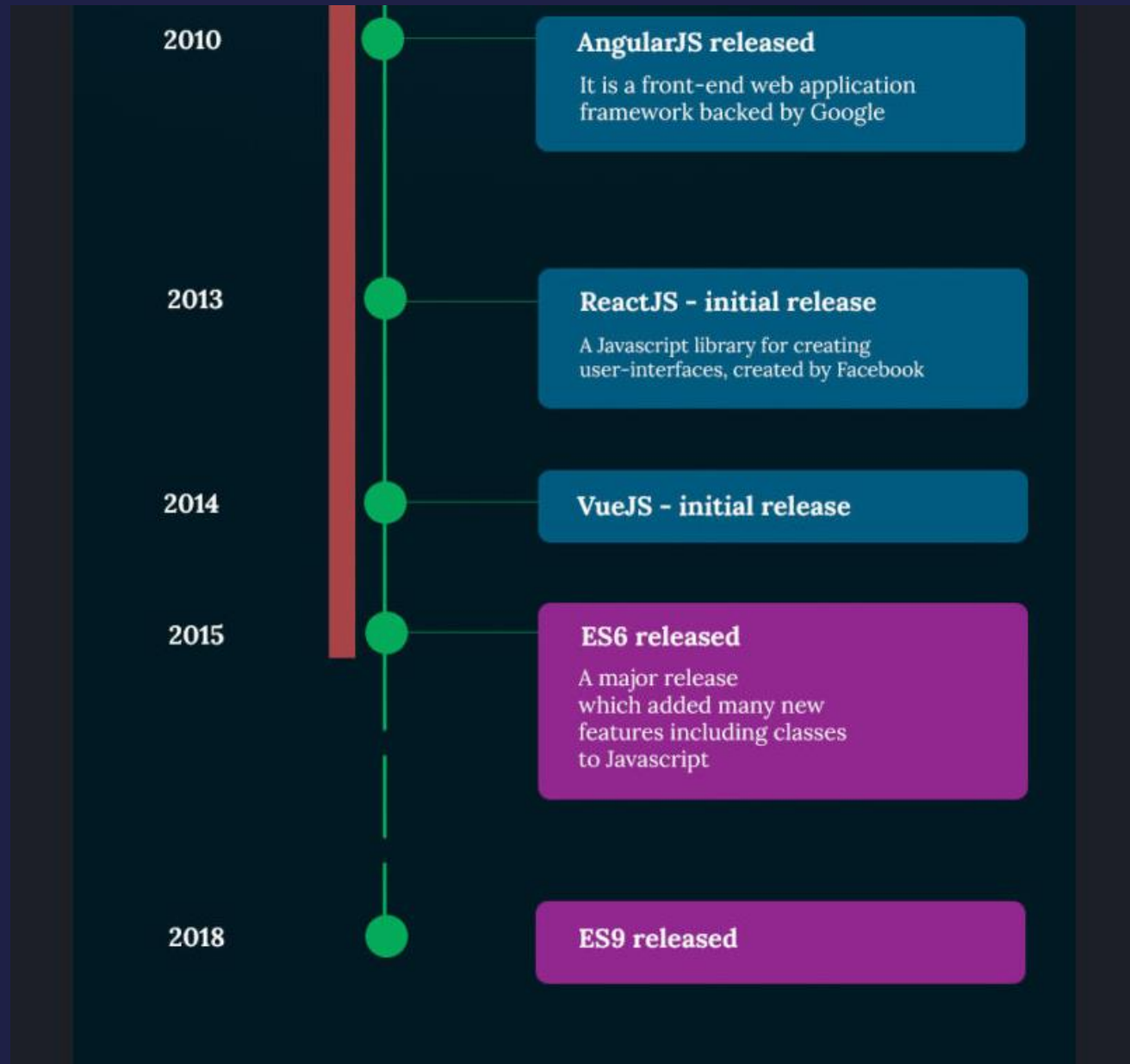
1. jQuery
2. Bootstrap
3. React JS
4. Symfony



# JQUERY

jQuery est une **bibliothèque JavaScript rapide**, petite et riche en fonctionnalités.

Cela **simplifie** considérablement la **navigation** et la **manipulation de documents HTML**, la **gestion des événements** et **l'animation**.



# BOOTSTRAP

Bootstrap est une large collection de morceaux de code pratiques et réutilisables écrits en HTML, CSS et JavaScript.

C'est également un cadre de développement frontend qui permet aux développeurs et aux concepteurs de créer rapidement des sites Web entièrement responsives.

# REACT JS

React permet aux développeurs de créer de grandes applications Web qui peuvent modifier les données, sans recharger la page.

L'objectif principal de React est d'être rapide, évolutif et simple. Il ne fonctionne que sur les interfaces utilisateur de l'application.

# SYMPHONY



Symfony est un **ensemble de composants PHP** ainsi qu'un **framework MVC** libre écrit en PHP.

Il fournit des fonctionnalités modulables et adaptables qui permettent de **faciliter et d'accélérer le développement** d'un site web.

MVC :  
Model  
View  
Controller

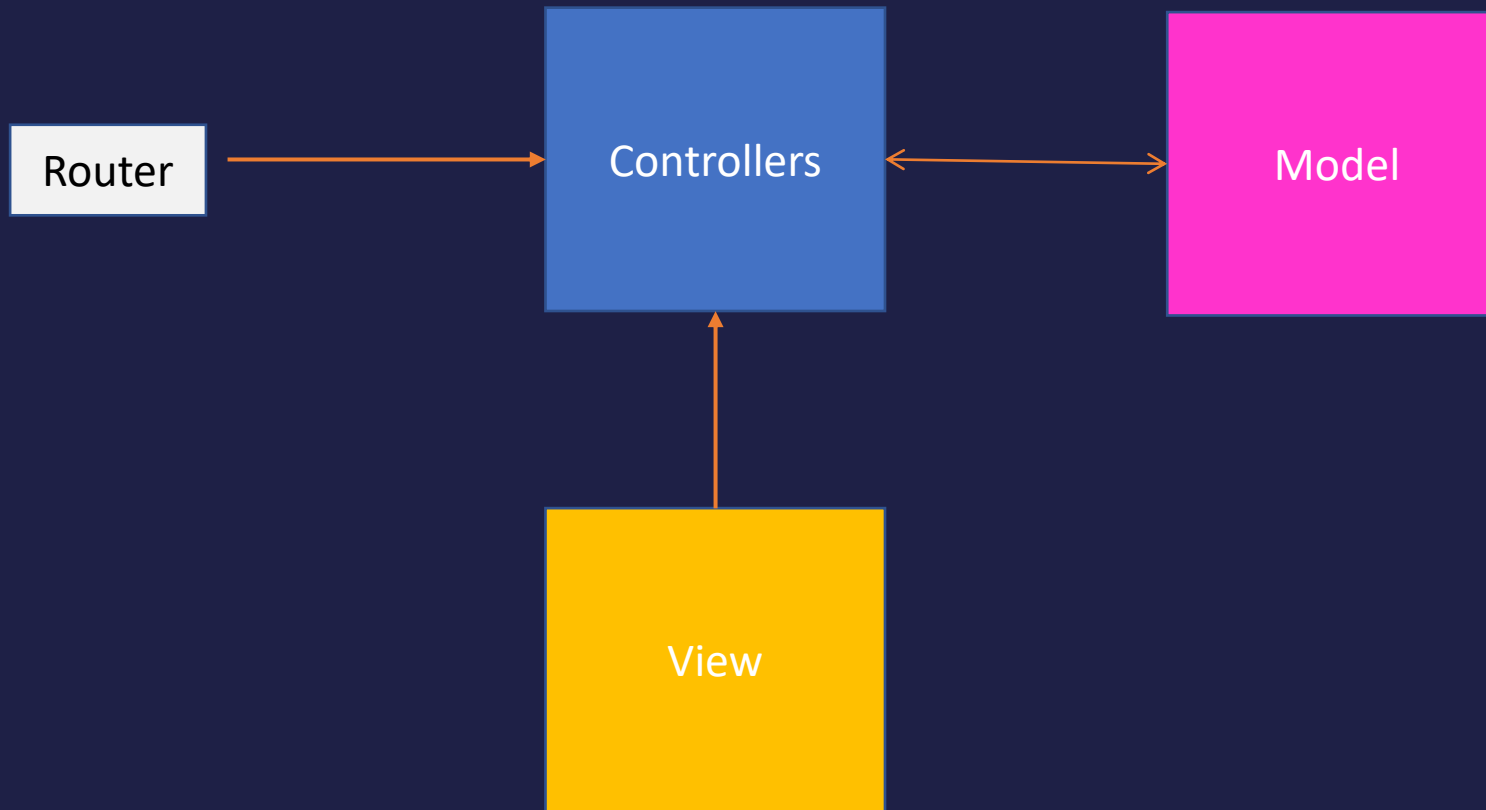
MVC :

Model = Entités

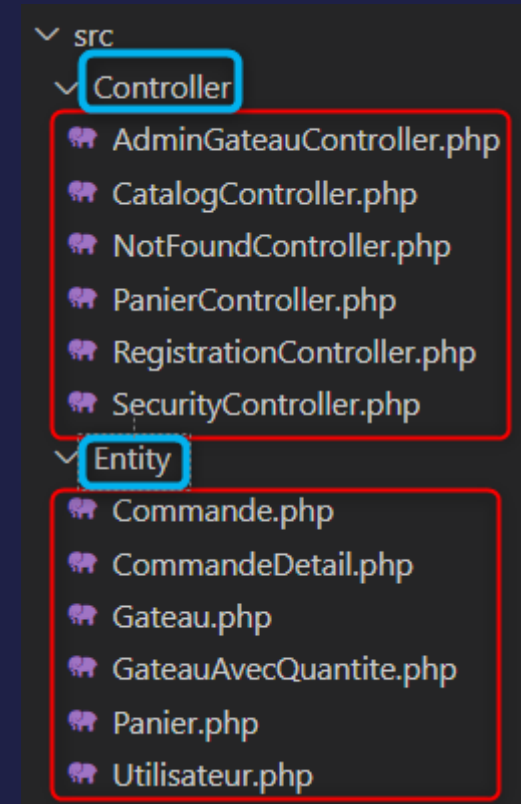
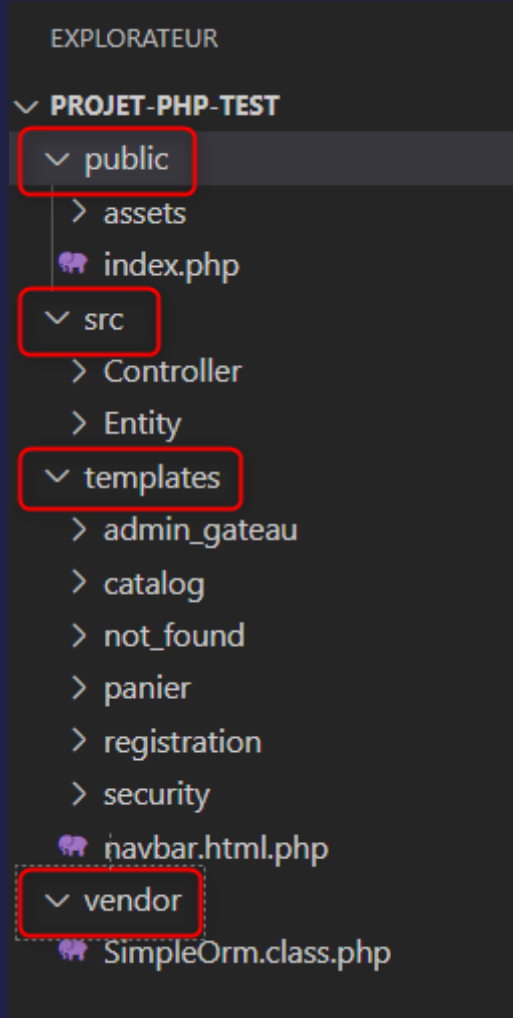
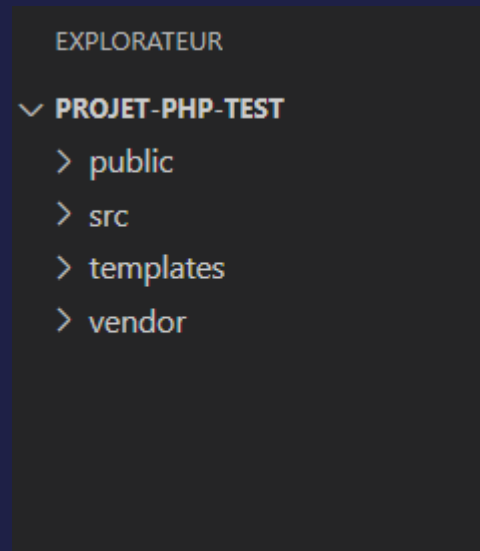
View = Code pour le visuel

Controller = Le reste

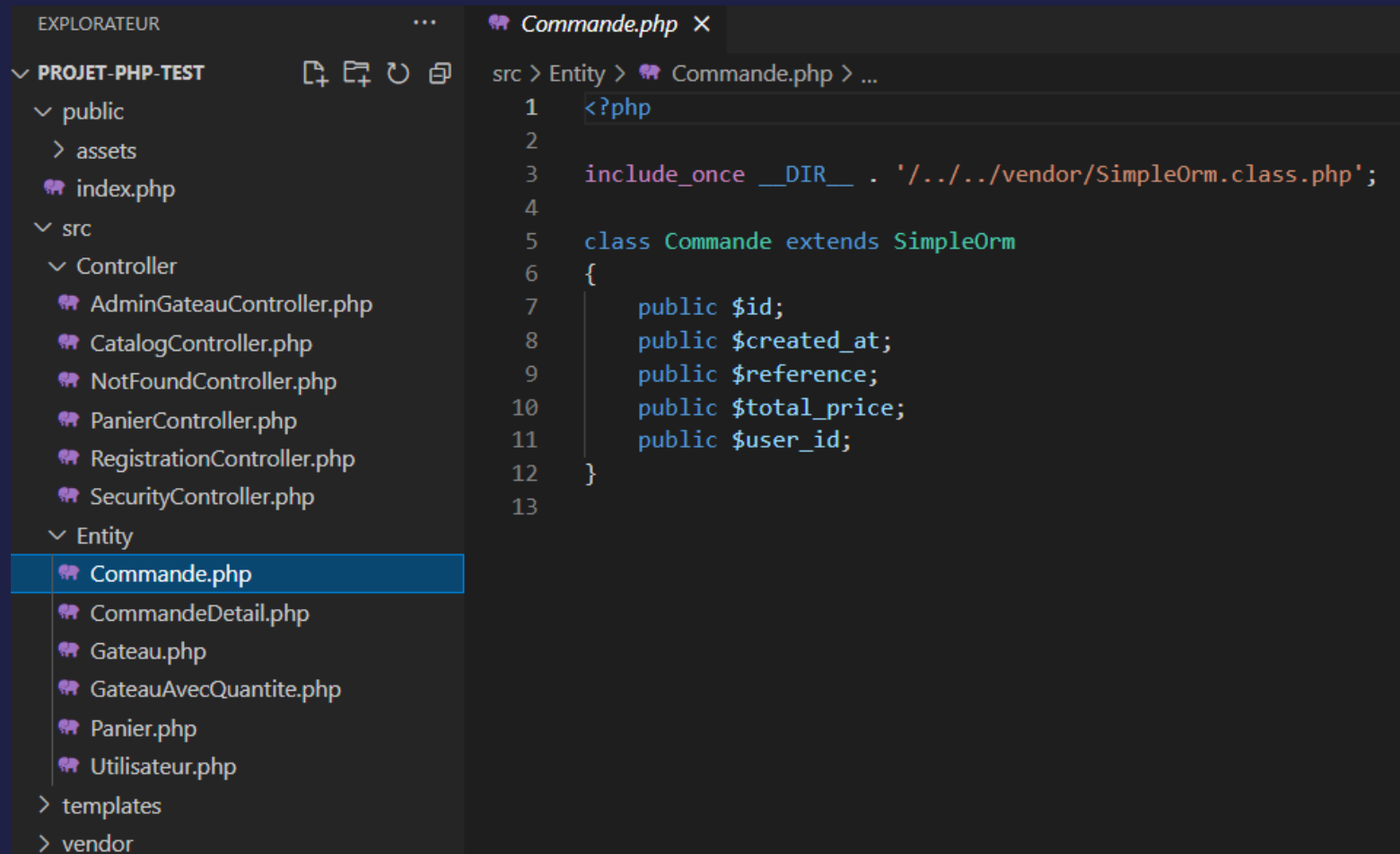
# Architecture MVC



# Architecture MVC



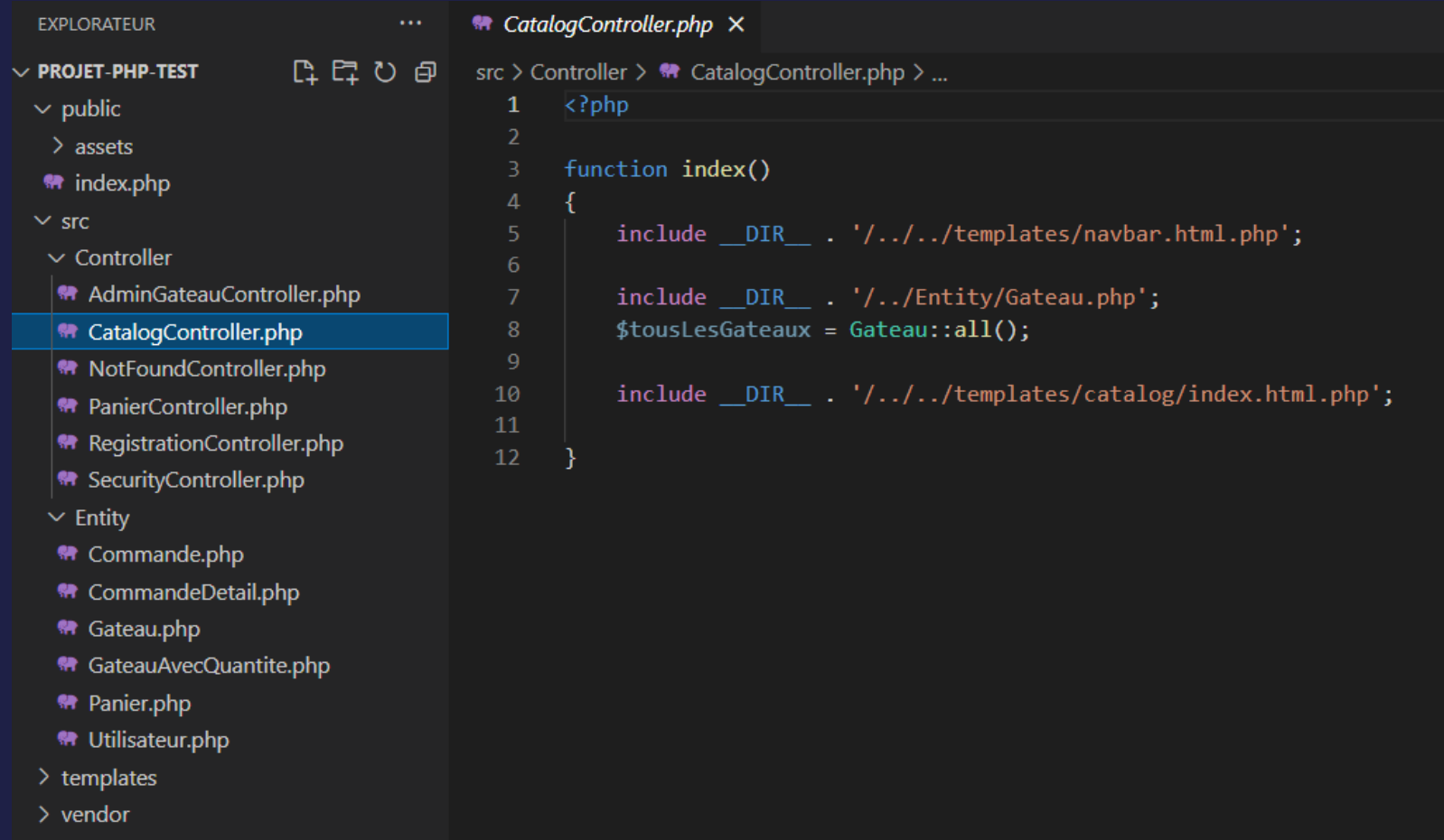
# Architecture MVC



```
EXPLORATEUR
PROJET-PHP-TEST
  public
  assets
  index.php
  src
    Controller
      AdminGateauController.php
      CatalogController.php
      NotFoundController.php
      PanierController.php
      RegistrationController.php
      SecurityController.php
    Entity
      Commande.php
      CommandeDetail.php
      Gateau.php
      GateauAvecQuantite.php
      Panier.php
      Utilisateur.php
  templates
  vendor

src > Entity > Commande.php > ...
1  <?php
2
3  include_once __DIR__ . '/../../vendor/SimpleOrm.class.php';
4
5  class Commande extends SimpleOrm
6  {
7      public $id;
8      public $created_at;
9      public $reference;
10     public $total_price;
11     public $user_id;
12 }
13
```

# Architecture MVC



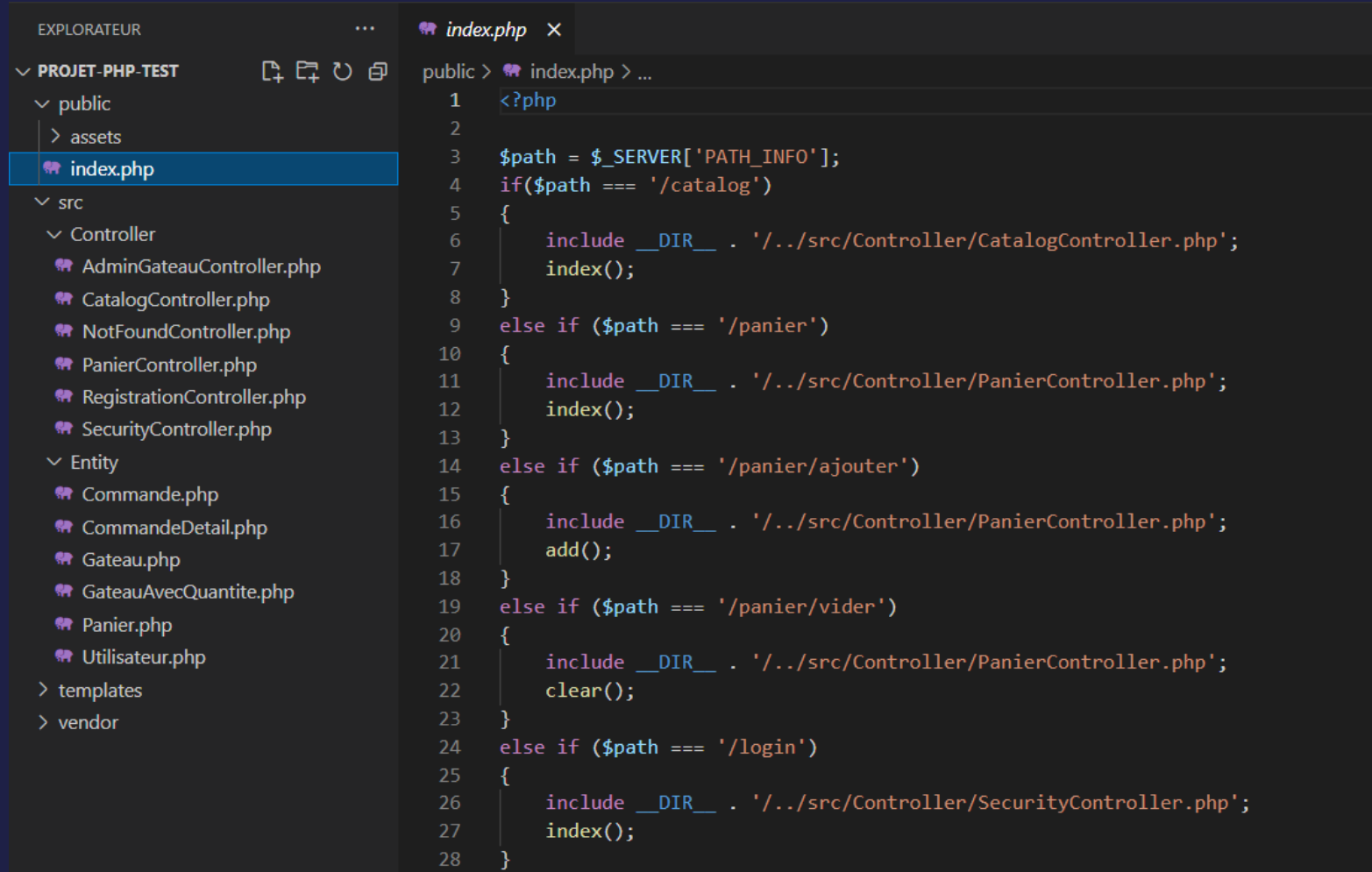
EXPLORATEUR

- PROJET-PHP-TEST
  - public
    - assets
    - index.php
  - src
    - Controller
      - AdminGateauController.php
      - CatalogController.php
      - NotFoundController.php
      - PanierController.php
      - RegistrationController.php
      - SecurityController.php
    - Entity
      - Commande.php
      - CommandeDetail.php
      - Gateau.php
      - GateauAvecQuantite.php
      - Panier.php
      - Utilisateur.php
    - templates
    - vendor

src > Controller > CatalogController.php > ...

```
1 <?php
2
3 function index()
4 {
5     include __DIR__ . '/../../templates/navbar.html.php';
6
7     include __DIR__ . '/../Entity/Gateau.php';
8     $tousLesGateaux = Gateau::all();
9
10    include __DIR__ . '/../../templates/catalog/index.html.php';
11
12 }
```

# Architecture MVC



```
EXPLOREUR
PROJET-PHP-TEST
  public
    assets
    index.php
  src
    Controller
      AdminGateauController.php
      CatalogController.php
      NotFoundController.php
      PanierController.php
      RegistrationController.php
      SecurityController.php
    Entity
      Commande.php
      CommandeDetail.php
      Gateau.php
      GateauAvecQuantite.php
      Panier.php
      Utilisateur.php
  templates
  vendor

index.php
1 <?php
2
3 $path = $_SERVER['PATH_INFO'];
4 if($path === '/catalog')
5 {
6     include __DIR__ . '/../src/Controller/CatalogController.php';
7     index();
8 }
9 else if ($path === '/panier')
10 {
11     include __DIR__ . '/../src/Controller/PanierController.php';
12     index();
13 }
14 else if ($path === '/panier/ajouter')
15 {
16     include __DIR__ . '/../src/Controller/PanierController.php';
17     add();
18 }
19 else if ($path === '/panier/vider')
20 {
21     include __DIR__ . '/../src/Controller/PanierController.php';
22     clear();
23 }
24 else if ($path === '/login')
25 {
26     include __DIR__ . '/../src/Controller/SecurityController.php';
27     index();
28 }
```