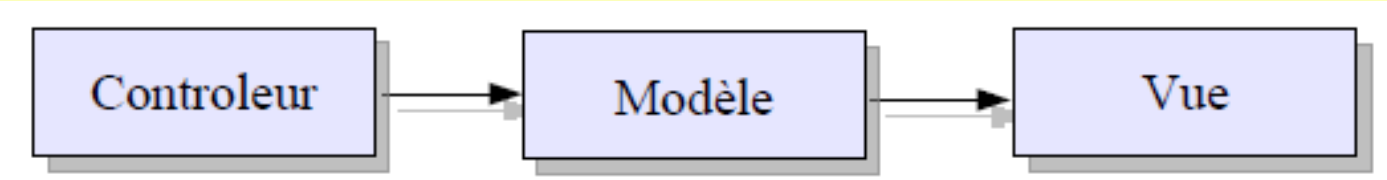
Le Modèle MVC

MVC

Le modèle MVC



Contrôleur : agit sur demande de l'utilisateur et modifie le modèle.

Modèle : contient les données du programme sous une forme compréhensible par la vue, notifie la vue des changements.

Vue : composant graphique affichant le modèle.

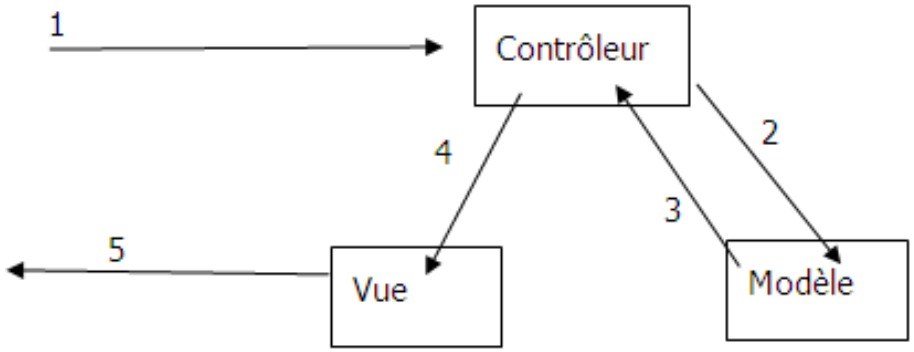
MVC

- **Architecture**

Packages	Commentaires
Métier	Classes issues de la base de données
Routeur	Le routeur concernant les services sera écrit dans la classe Api du dossier Routes
Contrôleurs	Chaque classe métier aura un contrôleur spécifique dont le rôle sera - Appeler la couche Service spécifique du dossier **DAO(Data Access Object)** - Retourner au client soit les données (requête GET soit un compte rendu requête POST)
DAO	**Classes Services** - ServiceFrais - ServiceLogin Les méthodes de ces classes contiennent les accès à la base de données.

MVC

Fonctionnement de MVC simplifié

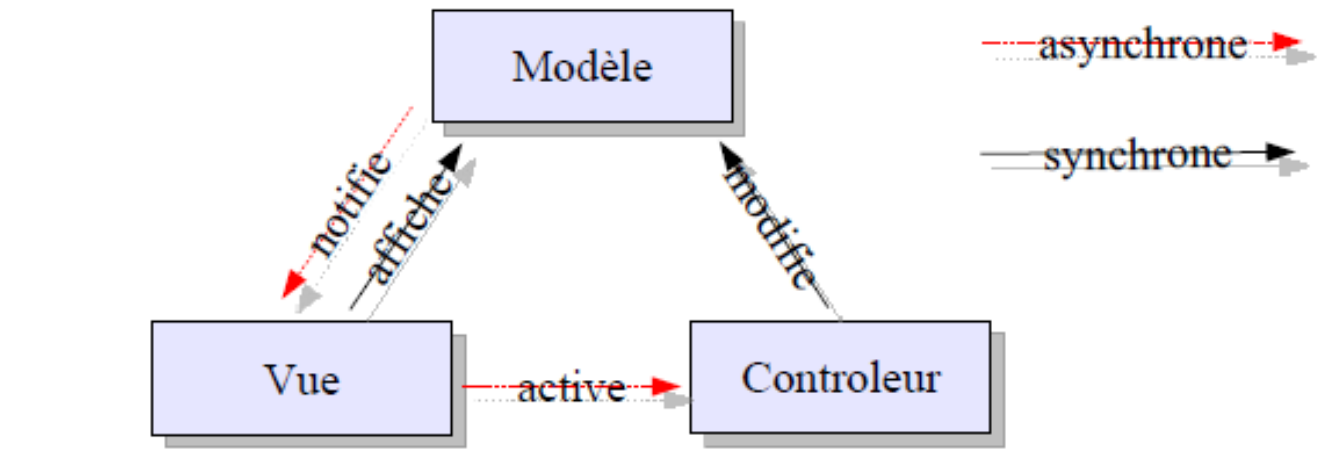


1) l'utilisateur remplit le formulaire et clique sur le bouton Submit, il envoie donc une requête HTTP vers le Contrôleur (Servlet) associée à l'attribut action du formulaire.
2) le Contrôleur récupère les données du formulaire, appelle le Modèle (Javabean) correspondant au traitement demandé, en fonction de l'origine de la demande et éventuellement des paramètres.
3) le Javabean effectue le traitement et restitue un résultat au Contrôleur auquel il rend la main.
4) le Contrôleur stocke le résultat dans un espace qui sera accessible aux pages JSP, puis il appelle la page JSP correspondant au résultat.
5) la Vue (page JSP) récupère le résultat, construit la page HTML et retourne à l'utilisateur la réponse HTTP.

MVC

Le modèle MVC

Pour un modèle, il peut y avoir **plusieurs vues et plusieurs contrôleurs.**



La notification des changements d'état

- entre le modèle et les vues et
- entre la vue et les contrôleurs

se font de façon **asynchrone (par *listener*)**

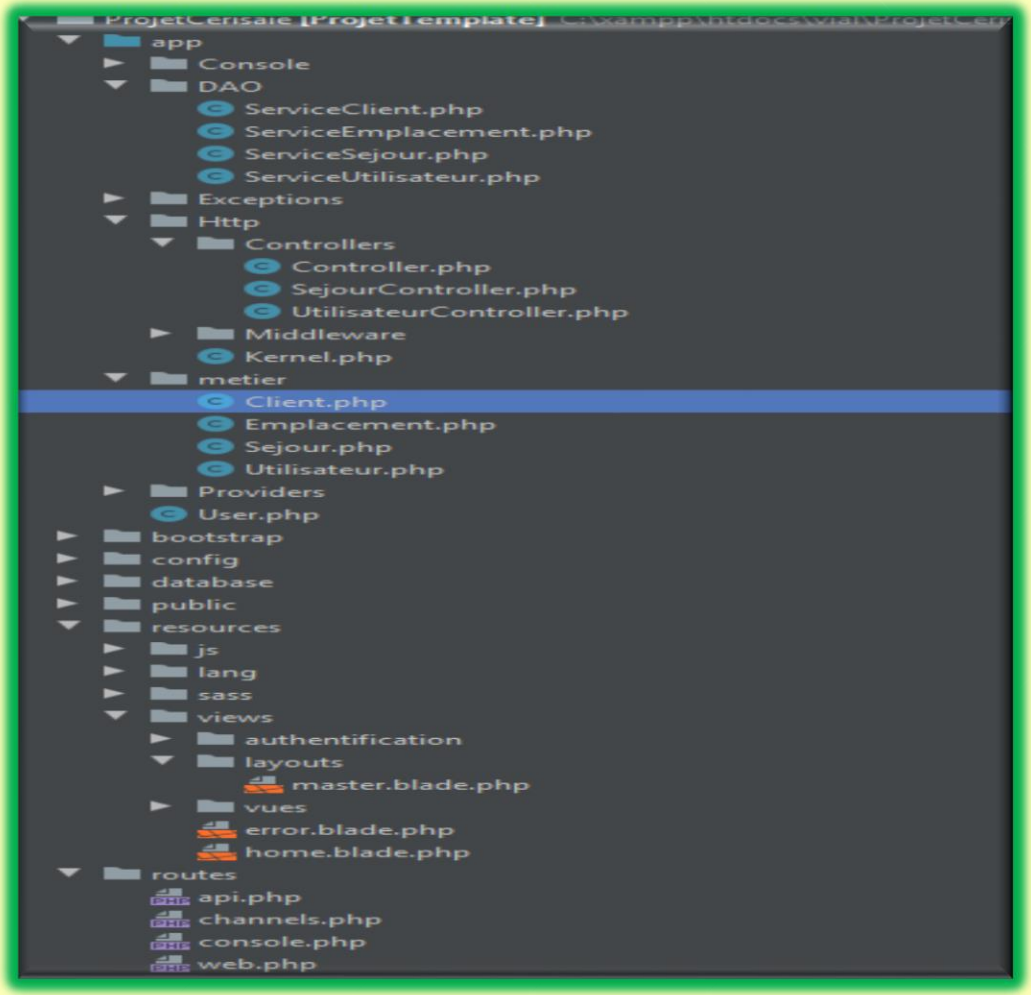
MVC

Fichiers générés

```php
<?php

/*...*/

namespace App\metier;

use ...

class Sejour extends Model {

    //On déclare la table Sejour
    protected $table = 'sejour';
    public $timestamps = false;
    protected $fillable = [
        'NumSej',
        'NumCli',
        'NumEmpl',
        'DatedebSej',
        'DateFinSej',
        'NbPersonnes'
    ];
    public $timetamps = true;


}
```

MVC

- **Conception approche distribuée : Gestion d'un stage**



MVC

- ## Code d'un contrôleur

```php
<?php

/*...*/

namespace App\Http\Controllers;

use ...

class SejourController extends Controller {

    // On recherche tous les séjours
    public function listeSejours() {
        try {
            $unSejour = new ServiceSejour();
            $mesSejours = $unSejour->getListeSejours();
            return view( view: 'vues/listerSejours', compact( varname: 'mesSejours'));
        } catch (MonException $e) {
            $erreur = $e->getMessage();
            return view( view: 'Error', compact( varname: 'erreur'));
        } catch (Exception $ex) {
            $erreur = $e->getMessage();
            return view( view: 'Error', compact( varname: 'erreur'));
        }
    }
```

MVC

- **Code d'un traitement de la couche DAO**

```php
//SELECT * FROM SEJOUR
public function getListeSejours() {
    try {
        $mesSejours = DB::table('sejour')
            ->Select('NumSej', 'NomCli', 'NumEmpl', 'DatedebSej', 'DateFinSej', 'NbPersonnes')
            ->join('client', 'sejour.NumCli', '=', 'Client.NumCli')
            ->get();
        return $mesSejours;
    } catch (QueryException $e) {
        throw new MonException($e->getMessage());
    }
}
```

MVC

- ## Code d'une page de la vue

```
@extends('layouts/master')
@section('content')
<h1>Liste Séjours</h1>
<div class="well">
    <table class="table table-bordered table-striped">
        <thead>
            <tr>
                <th>Numéro Séjour</th>
                <th>Nom Client</th>
                <th>Numéro Emplacement</th>
                <th>Date début</th>
                <th>Date de fin</th>
                <th>Nombre de personnes</th>
                <th>Modification</th>
                <th>Supprimer</th>
            </tr>
        </thead>
        @foreach($mesSejours as $unSejour)
        <tr>
            <td>{{$unSejour->NumSej}}</td>
            <td><b>{{$unSejour->NomCli}}</b></td>
            <td>{{$unSejour->NumEmpl}}</td>
            <td>{{ Carbon\Carbon::parse($unSejour->DatedebSej)->format('d/m/Y')}} </td>
            <td>{{Carbon\Carbon::parse($unSejour->DateFinSej)->format('d/m/Y')}}</td>
            <td>{{$unSejour->NbPersonnes}}</td>
            <td style="..."><a href="{{ url('/modifierSejour')}}/{{$unSejour->NumSej}}">
                    <span class="glyphicon glyphicon-pencil" data-toggle="tooltip" data-placement="top" title="modfier"
            </td>
            <td style="...">
                    <a class="glyphicon glyphicon-remove-sign" data-toggle="tooltip" data-placement="top" title="Suppr
                      onclick="javascript:if (confirm('Suppression confirmée ?')) window.location ='{{ url('/supprimer
                    </a>
            </td>
```