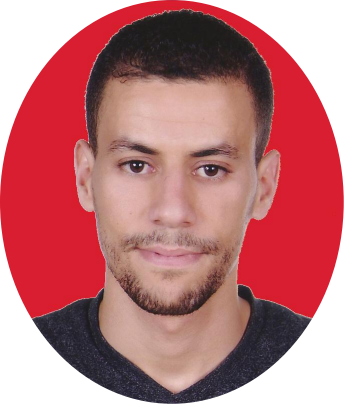


AngularJs





Jamel Eddine
MEJRI



Ingénieur FrontEnd chez Wevioo

Formateur JavaScript

Ingénieur GLSI from ISI

Co-Founder of CreativeLab Club

mejri.jameleddine@gmail.com

Session de Formation Angular

9am ---- 1pm

1pm ---- 3pm

25h Totale

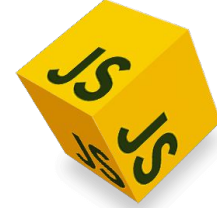


Théorique,
Pratique,
Exercices,
Evaluation.

Quelques rappels pour bien commencer



LANGUAGES
& SKILLS
FOR ASPIRING
FRONT-END
WEBDEVELOPERS



iDE



Model

Controller

View



Architecture 1 tier

Une application se présente sur différentes parties :

- La couche interface homme machine
- La couche de traitement
- La couche de gestion des données. Et toute application possède ces trois parties.

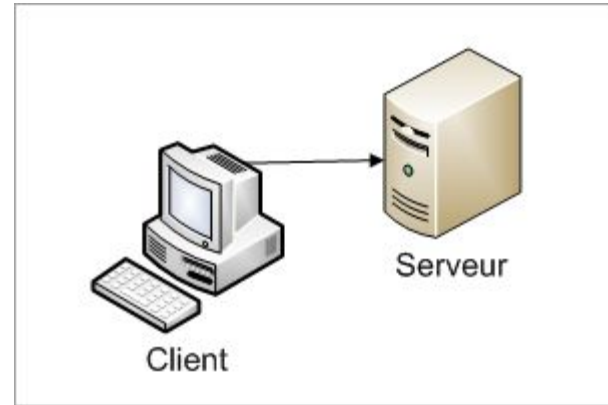
Jusqu'a 1990 ces trois couches étaient la plupart de temps sur une seule machine



Architecture 2 tiers

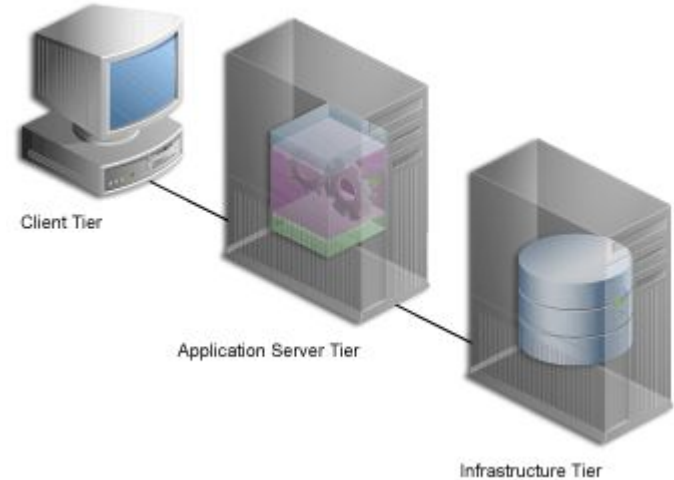
1. Le client

2. Le serveur qui se contente de répondre aux requêtes du client



Architecture 3 tiers

Avec l'apparition des technologies Web, il est possible de séparer la couche présentation de la couche applicative



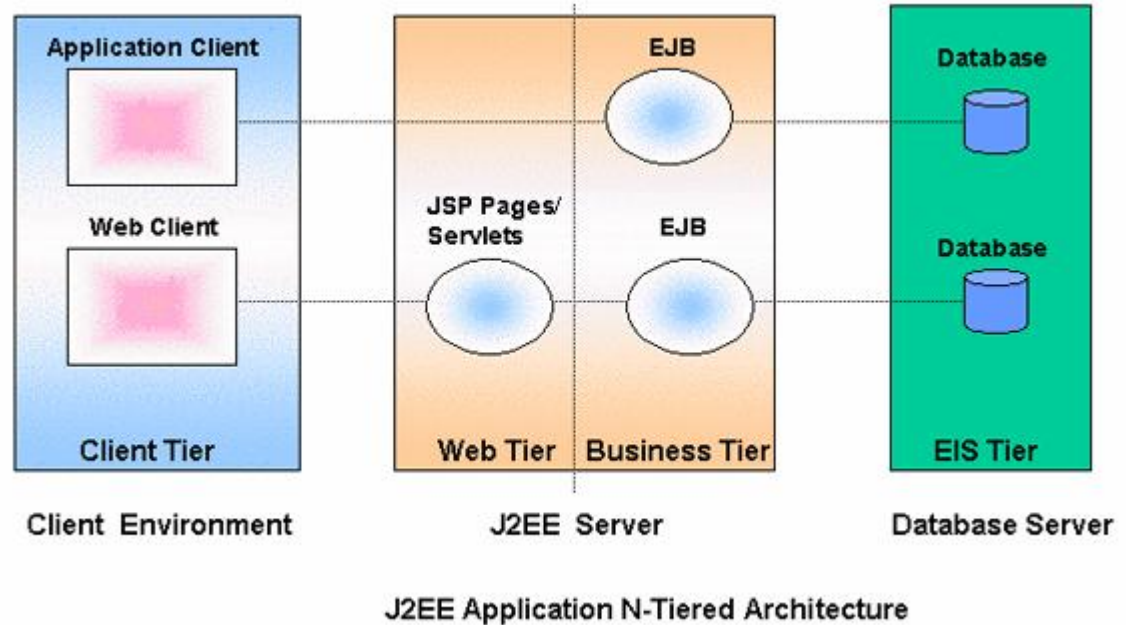
Architecture n tiers

L'architecture n-tiers est aussi appelée architecture distribuée ou architecture multi-tiers.

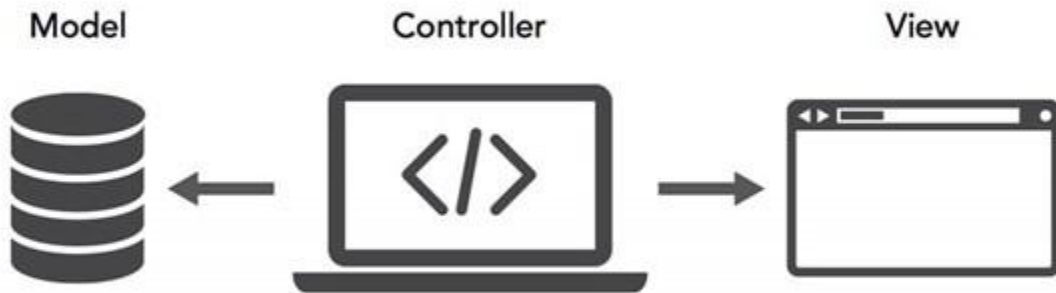
Le tier client : - Un navigateur Web

Le tier Applicatif : Java Servlets - JSP - PHP, JavaScripts

Base de données via SQL, JDBC,



MVC



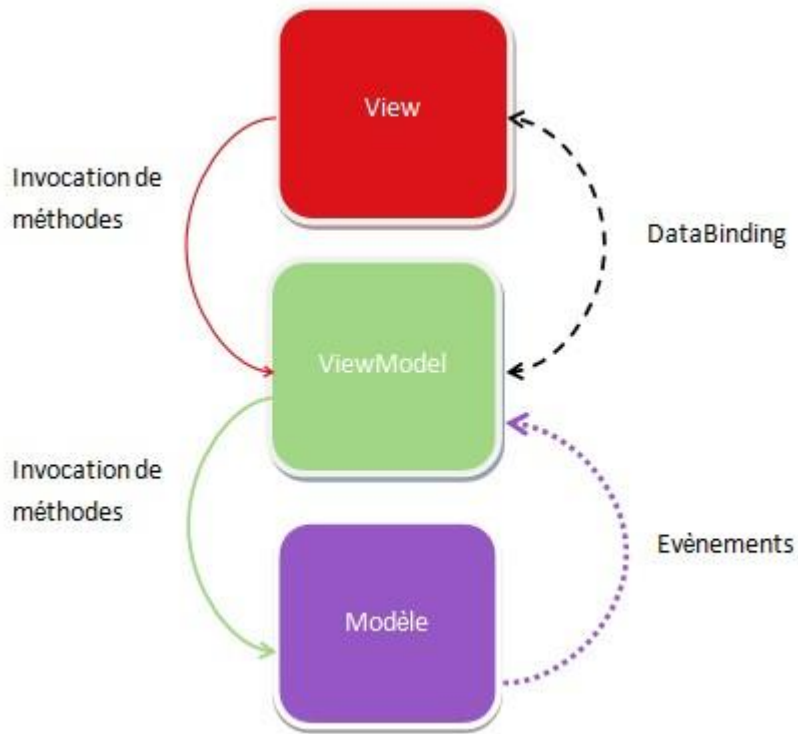
MVC est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. C'est un modèle qui a été conçu au départ pour des applications dites « client lourd », c'est-à-dire dont la majorité des données sont traitées sur le poste client

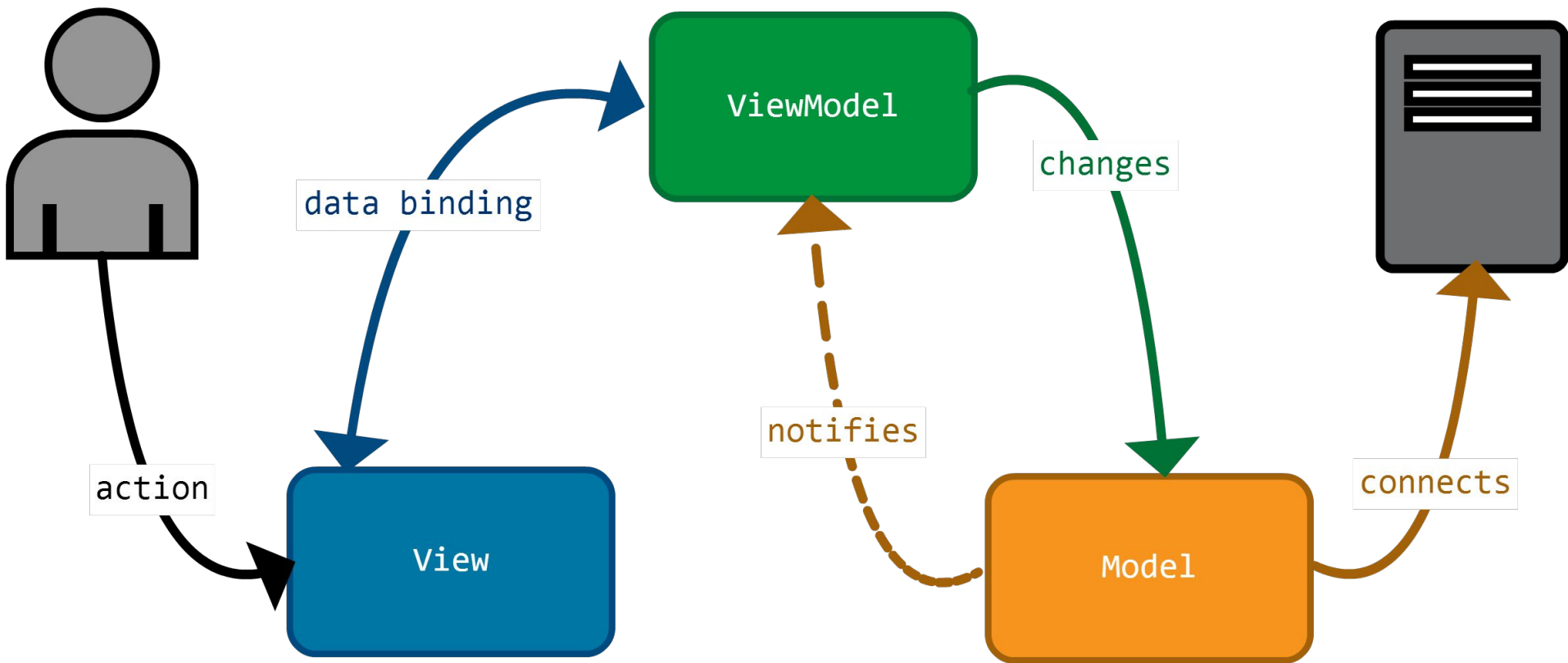
MVVM

La **vue** est couplée aux données via du DataBinding et invoque les méthodes du ViewModel. (ng-click)

Le **ViewModel** invoque les méthodes du modèle (CRUD). Il contient la data spécifique à la gestion de l'écran et les méthodes de réponses aux interactions utilisateurs. Il contient également une référence vers un ou des modèles.

Le **modèle** contient la data et les méthodes de manipulation de cette dernière (calculs, appels de services, ...).





Planning

- ❑ introduction à angularjs et aux SPA
- ❑ première application avec angularjs
- ❑ les modules
- ❑ les contrôleurs et modèles
- ❑ Vues et Binding
- ❑ formatage des données
- ❑ les Scopes
- ❑ service provider et factory
- ❑ structure et organisation d'un projet
- ❑ npm, bower et gulp
- ❑ Routing navigation
- ❑ promise et requêtage (consommation de web services)
- ❑ les directives
- ❑ internationalisation
- ❑ test unitaires debug
- ❑ introduction à typescript
- ❑ Angular

AngularJs 101



Angular has nifty features such as two-way data binding, the ability to make custom HTML tags/attributes/comments that encapsulate functionality, and built-in directives that extend the functionality of normal HTML. Angular is also really versatile (polyvalent), it can be used for large, one-page app projects or for a small little element; there's not much setup required to start using Angular's built-in directives.

Pourquoi AngularJs

Angular est un socle technique qui se veut extensible et qui pousse vers un développement structuré. Il s'inscrit dans un mouvement d'innovation côté front-end, dont le but est d'éviter le chargement d'une nouvelle page à chaque action demandée.



1. Une navigation plus fluide pour le visiteur
2. Une meilleure gestion de contenu dynamique
3. Une plateforme extensible et modulaire

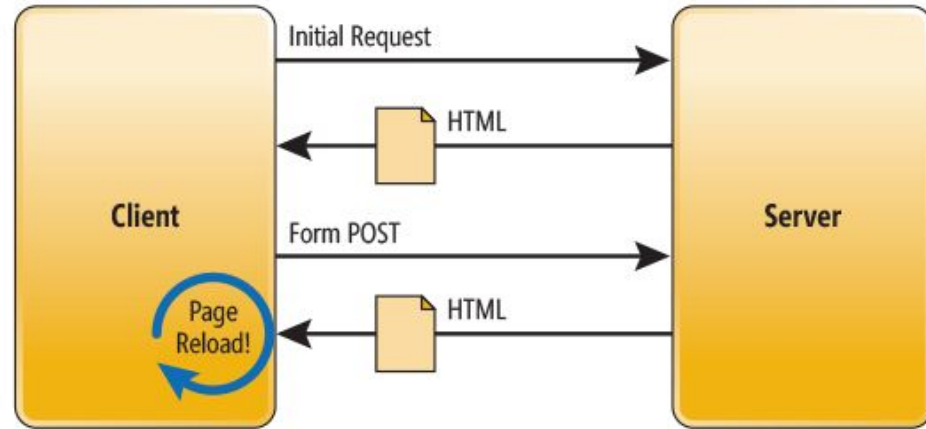
Pourquoi AngularJs

- Permet d'implémenter le pattern MVW pour structurer le code, ce qui améliore la qualité de code
- Favorise la collaboration entre les développeurs et les intégrateurs HTML
- Meilleure expérience utilisateur avec les applications monopages
- Apporte aux applications web les services traditionnellement apportés côté serveur
 - Injection de dépendances
 - MVC
 - Data-binding bidirectionnel
 - Routing
 - Templating

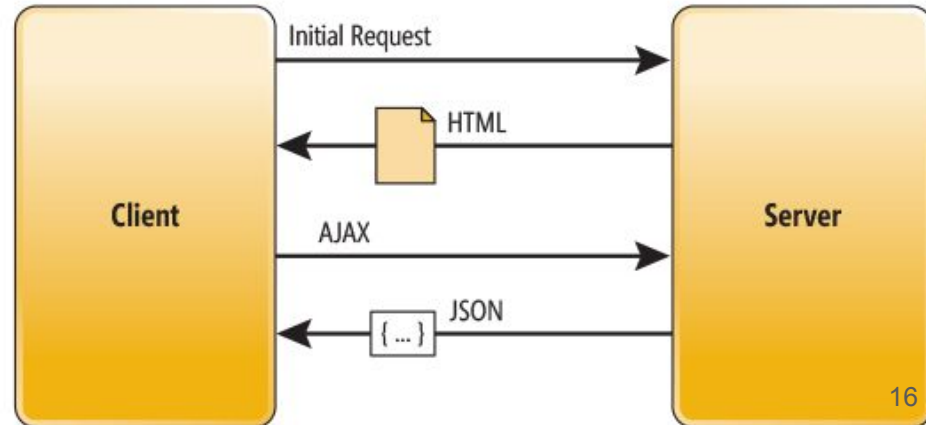
SPA (Single Page Application)

The majority of interactions are handled on the client without the need to reach a server.

Traditional Page Lifecycle



SPA Lifecycle



Installation (IDE)



Add Angularjs to app

<http://angularjs.org>

Hello World

$\{\{3*3\}\}$

Syntaxe AngularJs

```
var app = angular.module('Nom-Module', [ ]);
```

```
{ {} }
```

```
ng-model, ng-bind, ng-click
```

```
$scope, $rootScope
```

```
controller, service, factory, provider, filter, run, config, module, directive,
```

**HTML -> app.js -> angularjs
->Module -> Injection -> Test ->
scope**

.....

.....



Pour bien démarrer avec JS

```
var chaine1 = "Bonjour ";  
var chaine2 = "tout le monde";  
var chaine3 = chaine1+chaine2;
```

```
3 === 3    // true
```

```
3 === '3'  // false
```

```
1 == 1;    // true
```

```
"1" == 1;  // true
```

```
var cars = new Array("Saab", "Volvo", "BMW");
```




.....

.....

.....

☐

.....

☐
☐

.....

☐

1



2

3



$$1+2.$$

$$2+3$$

$$3*10$$

.....

.....