

Sprint 3

Formation POEI

Developpeur Java

Dumortier Steven, Menadjlia Nouha, Mollion Maéva



Le projet fil rouge: AppliCRM

But: Créer une application CRM (Customer Relationship Management) permettant de gérer des clients et des commandes.

- Créer, Lire, Mettre à jour, et Supprimer (CRUD) des clients
- CRUD des prestations (orders) sachant qu'une commande est liée à un client.

Pour cela, on veut que les actions soient possibles via des pages web.

Architecture

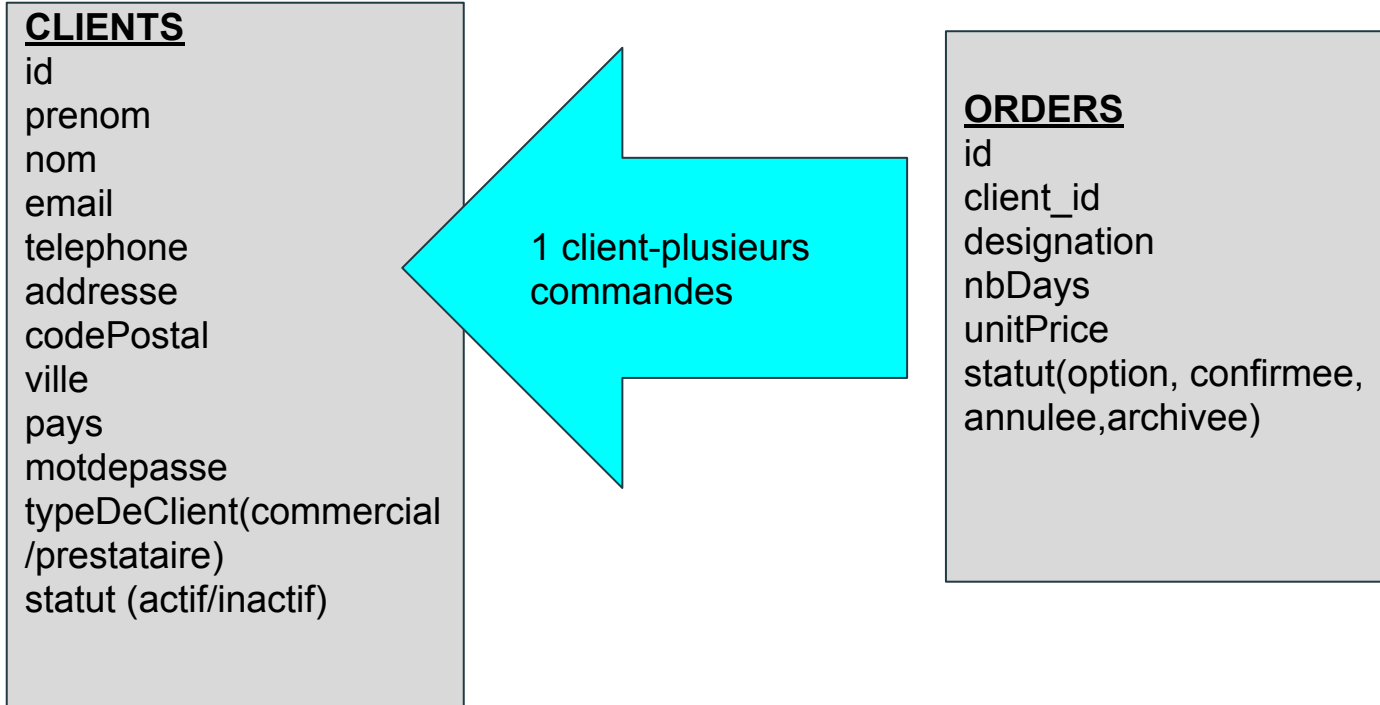
Base de données avec MySQL

Serveur pour faire le lien entre les pages web et l'application: TomCat

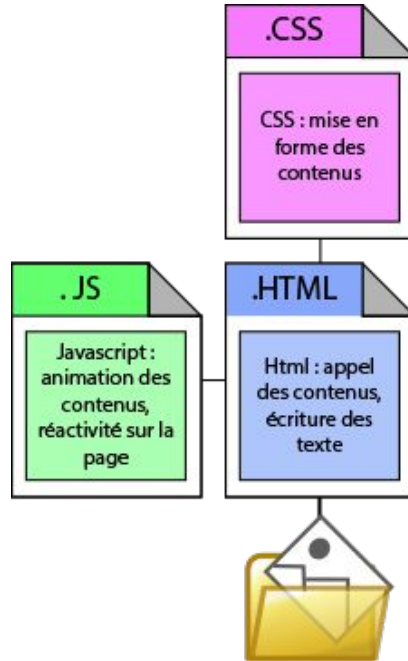
Backend avec Java JEE en utilisant Spring

FrontEnd avec Angular

La structure de la base de données



Les éléments essentiels d'un site web



The mockup shows a web page titled 'Ma page' with a yellow sun icon. It features a header with a picture icon, a main content area with a paragraph of Lorem Ipsum text, and a sidebar with the letters 'PUB'. Below the text are three input fields, a 'VALIDER' button, and a red circular arrow icon. To the right of the input fields is a red circular arrow icon with the text 'Vérification si tous les champs sont bien saisis'. Below the 'VALIDER' button is a red double-headed arrow icon with the text 'Envoie des données - AJAX'.

css Bootstrap



- Framework CSS à l'origine développé par les programmeurs de Twitter.
- Il impose un cadre de travail rigoureux pour la définition de la structure d'une page HTML et leurs liens avec les définitions CSS .
- Basé sur l'approche responsive du Web Design .
- l'utilisation du Framework ce fait à travers deux méthodes:
 - =>Le lien CDN et l'utilisation des classes Bootstrap dans notre HTML.
 - =>Téléchargement du thème.

Javascript

JS

Langage de programmation, comme java

Typage faible

langage de script orienté objet, interprété

Permet de rendre un site web dynamique en interagissant avec le DOM (Document Object Model) donc la page web. Peut être lu par le navigateur. À l'extérieur du navigateur, on utilise Node.js

```
1 <html>
2 <head>
3 <title>t1</title>
4 <script type="text/javascript">
5     function addNode()
6     {var newP = document.createElement("p");
7       var textNode = document.createTextNode(" This is a new text node");
8       newP.appendChild(textNode);
9       document.getElementById("firstP").appendChild(newP); }
10 </script> </head>
11 <body> <p id="firstP">firstP<p> </body>
12 </html>
13
```

Backend: spring



Introduction de Spring



Objectif : Création de la partie Back End d'une application d'entreprise en langage de programmation java.



Fonctionnement : **Incorporer** tel ou tel **module** de Spring dans le projet en fonction de ces besoins.



Définition :

- ❑ Un module correspond à une fonctionnalité de Spring
- ❑ Spring possède environ 20 modules différents



Exemple : Beans : le module qui permet de manipuler les objets Java (@Autowired)

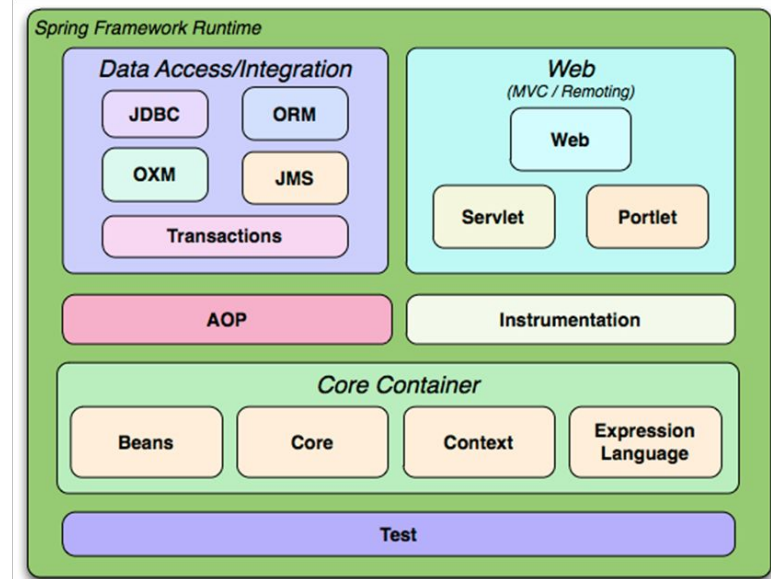


Schéma représentatif des modules de Spring

Introduction de Spring Boot



Objectif : Création de la partie Back End d'une application web en langage de programmation java



Fonctionnement : Incorpore et configure les modules de manière **automatique**.



Définition :

- ☐ Spring Boot est une **extension** du framework Spring.
- ☐ Spring Boot est un **facilitateur**



Exemple photo du gâteau :

- ☐ À gauche les **ingrédients** d'un gâteau représentant un **module**, la **recette** représentant les **configurations** (Spring)
- ☐ À droite un **gâteau** déjà préparé (Spring Boot)



spring boot



Photo illustrant la différence entre Spring et Spring Boot

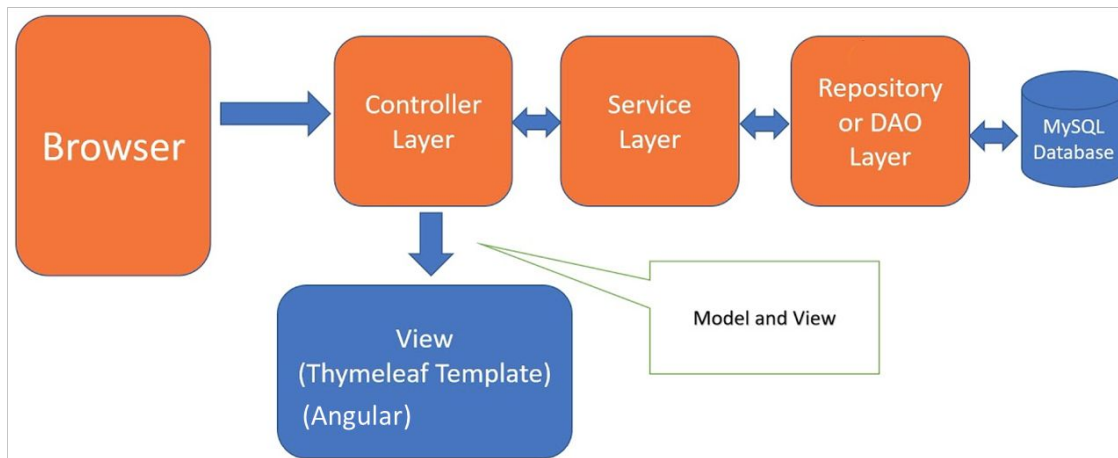
Comparaison de Spring et Spring Boot



Spring	Spring Boot
Développer des applications d'entreprise.	Développer des API REST.
Configuration manuelle	Configuration automatique
Écrire beaucoup de code.	Ecrire peu de lignes de code.
Long a mettre en place	Rapide à mettre en place
Niveau ingénieur expérimenté	Tout niveau



L'architecture de Spring-Boot



Controller :

- Interagit directement avec l'utilisateur. (Par requête HTTP)
- Valide les paramètres de la requête avant de les transmettre à la couche service.
- Met à jour le modèle et le transmet à une vue pour ensuite être affiché à l'utilisateur.

Repository :

- Fais la connexion entre les POJO et la BDD.
- Réalise automatiquement les recherches SQL dans la BDD.
- Transmet les informations de la BDD au service.

Service :

- Fourni le code métier (les fonctionnalités du site web).

Spring Boot dans notre code CRM

Controller

@Controller / @RestController
@AutoWired (pour service)
@(Get/Post/Delete/Put)Mapping

Service

@Service
@AutoWired (pour repository)

Repository

x2

@Repository
⚠ Interface ⚠
extend JpaRepository<>

POJO class client

@Entity
@Table()
@Id
@GeneratedValue()
@Column()
@OneToMany
mappedBy / fetch

POJO class order

@Entity
@Table()
@Id
@GeneratedValue()
@Column()
@ManyToOne
@JoinColumn

Application.properties

pom.xml

Architecture Spring Boot

POJO (objet simple) pour le CRM

DTO/mapper

Fichier de dépendance Maven

Mapper (traduire en DTO ou en POJO)







DTO (Adapter un POJO pour la view) ex : Enlever l'ID
dans l'affichage d'un client

Front end:

Accueil

Clients

Orders

Intitule	type	Prix	...	
Angular	formation	a euros		  
Spring	coaching	b euros		  

Ajouter une prestation

Angular



Feature	Angular	React	Vue
UI / DOM Manipulation	✓	✓	✓
State Management	✓	✓	✓
Routing	✓	✗	✓
Form Validation & Handling	✓	✗	✗
Http Client	✓	✗	✗

Angular

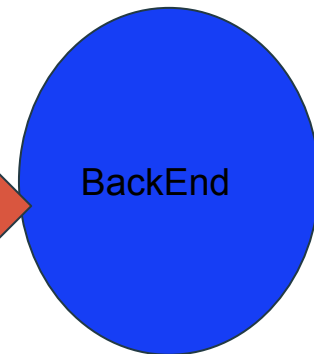
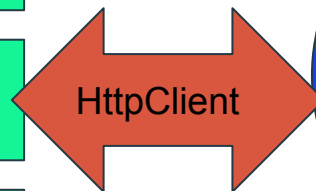
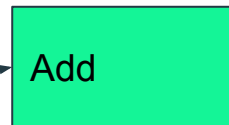
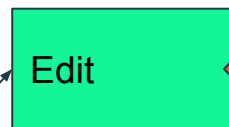
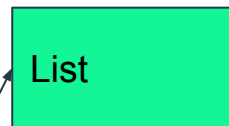
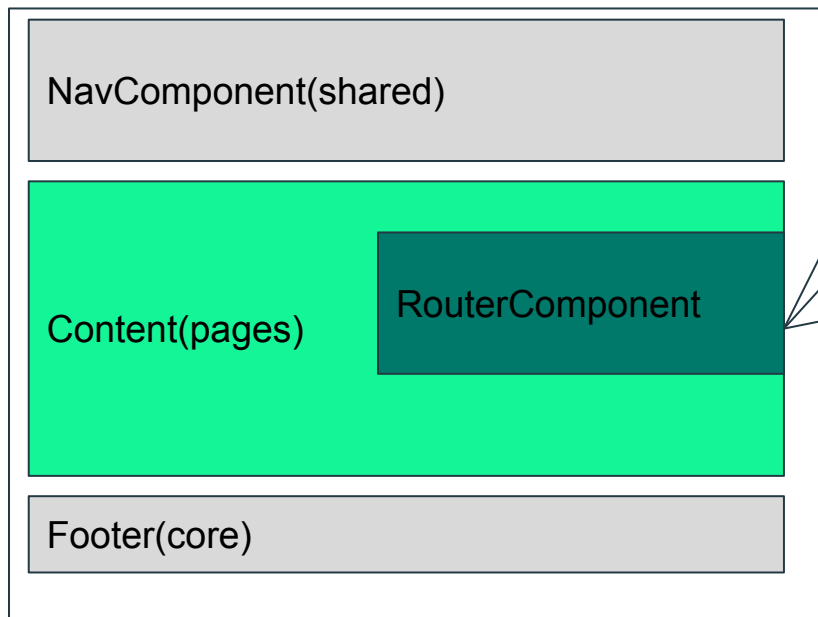


Framework (cadre de travail) pour faire du front-end

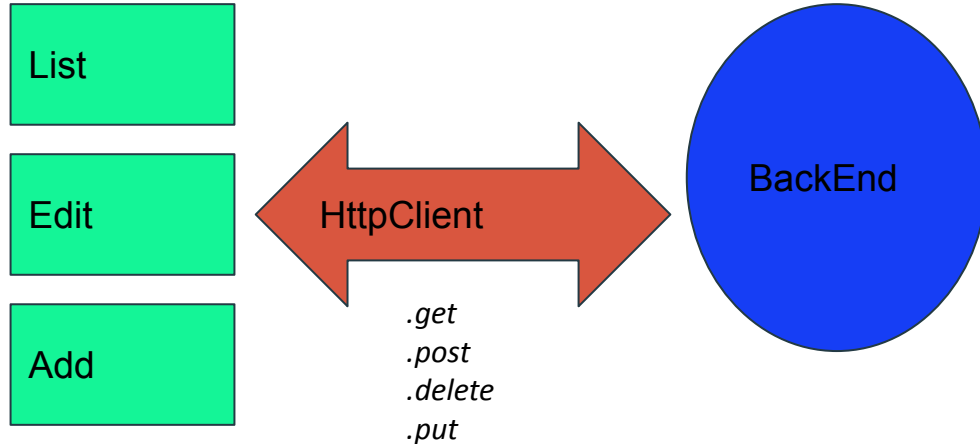
Utilise des **WebComponents**: fonctionnalités réutilisables pour les sites web

Single Page Application: pas besoin de recharger tout le site, mais seulement une partie de la page.

Angular



Angular



exemple de template:

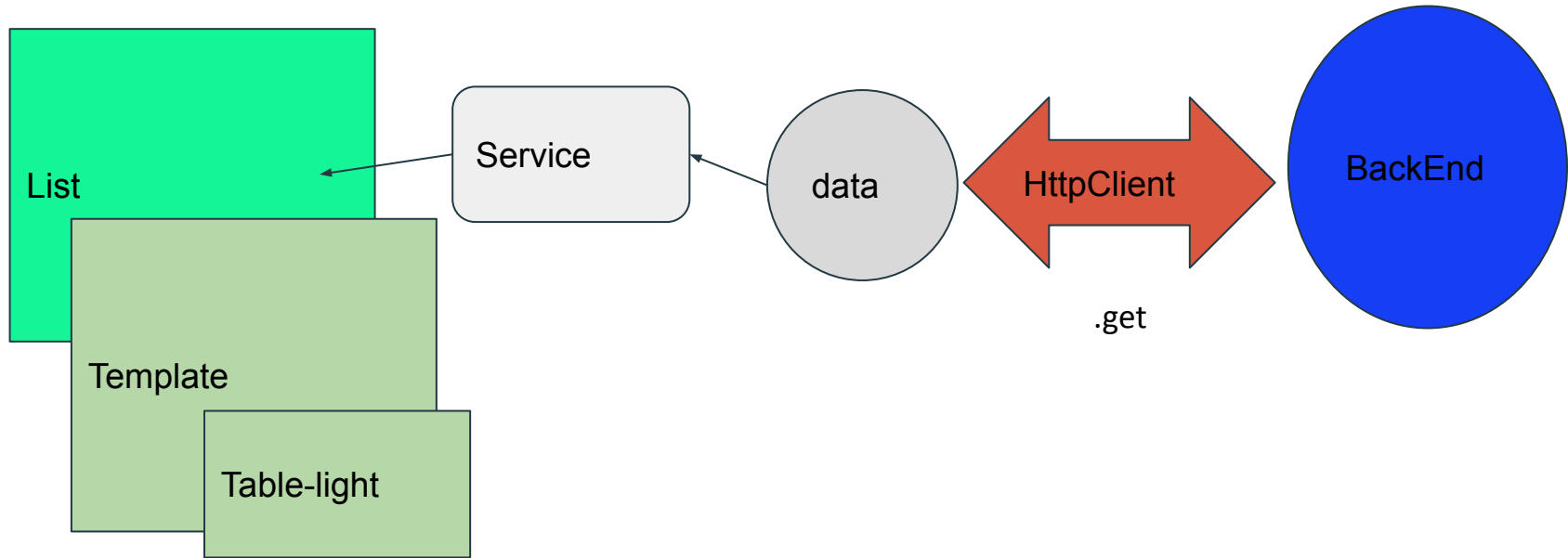
```
<p *ngFor="let client of clients">
  {{client.nom}}
```

```
  {{client.prestation}}
```

```
</p>
```

Formate les données pour les afficher
dans les pages à l'aide d'un template et
de la string interpolation

Angular



Front-end angular

Bonnes pratiques d'agilité appliquées:

Réutilisabilité + Transformabilité+ Documentation : la structure du code est claire et décomposée, permettant de modifier facilement les éléments ciblés.

Légèreté : on ne charge les composants qu'au moment où l'utilisateur en a besoin.



Merci pour votre attention et pour la formation !



Christian
Alex
JC
Sonia
Christophe

