3 blocks distincts

Front -end (Angular, html, css,…)

Back-end (Spring et bd)

Persistance

## Application multi-tier:

SKELETON FULLSTACK =/= MODEL MVC car MODEL MVC est dans le codage

MVC va se voir en lisant le code. (Design Pattern = façon de coder)

SKELETON CONCERNE L’ARCHITECTURE 🡪 Installation de java, bootstrap, intellij, node.js

**Fullstack** = Un développeur qui va tout faire (Front-end, back-end et persistance(bd))

Couche front-end: HTML, CSS, Angular, Typescript,…

\*LA COUCHE FRONT ET BACK se parle avec un Web Service : Un web service sert à exposer les méthodes du back-end au front-end(qui va les call) à l’aide d’un protocole http.

**JDCB** est un protocole de communication entre JAVA et les base de données 🡪 **JPA(outils utilisant JDBC)** pour éviter de tout recoder. Exécuter des requêtes à travers JAVA.

Dans ce skeleton, jdbc est le prot. De comm. Entre back-end et la base de données

Couche back-end : Spring, Spring boot(Methode ajouter, supprimer, update,…)

BD: h2, persistence

\*LA VUE NE PARLE PAS À LA BD

http :PORT

JDBC :PORT

H2 est une base de donnée embarqué qui va utiliser le port choisi pour les requêtes http(8080 par défaut).

Spring : 1) CRUD = repository est fourni.

**Transient** est un mot clé java pour dire que la variable ne sera pas prise dans la sérialisation : ne fera pas parti de la table.

Tous les objets du model doivent avoir une interface équivalente qui héritent de **JpaRepository** ou **CrudRepository**. Pour éviter de retaper tous les méthodes du CRUD (C,R,U,D).

**PagingAndSortingRepository** =(Paging) Permet de faire des paquets si on veut afficher par paquet (Sorting) va trier les enregistrements de la table.

CrudRepository + PagingAndSortingRepository = **JpaRepository(Java Persistence API**

**)**

NE JAMAIS PARTIR AVEC UNE BD VIERGE! Les premiers enregistrements doit être écrit par le développeur!

POJO : Plain Old Java Object @Entity @Data

Spring.jpa.hibernate.ddl-autocreate-drop = Drop et recréez.

**JPA(Java persistance API)**

Spring Data

Est compose de.

* JDBC: protocole de communication entre java et un SGBDR
* Hibernate: Ensemble de classe JAVA qui génère le code SQL.

Web Service (2 choix)

REST Controller: Représentation state Transfer(Utilise les fichiers JSON ) et passe par HTTP

SOAP: Utilise les fichiers XML

WebService fait le lien entre le front-end et le back-end

Controller🡪 @RestController 🡪Injection de dépendance d’un controlleur en IOC

On doit injecter une interface = pour qu’elle soit fourni au besoin (sur demande)

En utilisant @. = Injection de dépendance = IOC(Inverse of Control)

@Autowired = détecte lorsqu’

@RequetMapping() 🡪Appelé par HTTP

Public List<Employee> getAll(“/getAll’, method = RequestMethod.GET){

Return repository.findAll();

}

Tableau intelligent qui accepte les doublons

Npm = equivalent gradle et maven pour Node.js

@CrossOrigin{origin = ‘’http://localhost :4200/} 🡪 serveur de deploiement accepte que les requetes venant de l’origin ci-dessus.