ATELIER II :

Questions :

1. Qu’est ce que des Services Web Full Rest ? Quelles sont les contraintes imposées par ce type de service ?

Un service Web Full REST est un service web qui utilise la structure REST. Les services web RESTful sont conçus pour fonctionner au mieux sur le Web.

REST est un style architectural qui spécifie les contraintes, telles que l’interface uniforme, qui si appliquée à un service Web induisent des propriétés souhaitables, telles que la performance, l’évolutivité et la modifiabilité, qui permettent aux services de mieux fonctionner sur le Web.

Contraintes :

* Client-serveur : Les responsabilités sont séparées entre le client et le serveur.
* Sans état : La communication client–serveur s'effectue sans conservation de l'état de la session de communication sur le serveur entre deux requêtes successives.
* Avec mise en cache : Les clients et les serveurs intermédiaires peuvent mettre en cache les réponses.
* En couches : Un client ne peut habituellement pas dire s'il est connecté directement au serveur final ou à un serveur intermédiaire.
* Avec code à la demande (facultative) : Les serveurs peuvent temporairement étendre ou modifier les fonctionnalités d'un client en lui transférant du code exécutable.
* Interface uniforme :
  + Indentification des ressources dans les requêtes : Chaque ressource est identifiée dans les requêtes
  + Manipulation des ressources par des représentations : Chaque représentation d'une ressource fournit suffisamment d'informations au client pour modifier ou supprimer la ressource
  + Messages autodescriptifs : Chaque message contient assez d'information pour savoir comment l'interpréter.

1. Qu’est ce qu’un gestionnaire de dépendance ? Maven est-il le seul ? Quel est l’avantage d’utiliser un gestionnaire de dépendances ? Quelles sont les grandes étapes de fonctionnement de Maven ?

Un gestionnaire de dépendance sert à gérer toutes les dépendances d’un ou des projets comme des bibliothèques ou des frameworks.

Maven n’est pas le seul gestionnaire de dépendances. Il en existe d’autres comme Composer ou Pear.

Avec un gestionnaire de dépendances, Il est plus facile de rajouter une dépendance au projet.

Maven distingue deux types de dépôts : local et distant (remote). Ces dépôts peuvent être gérés à plusieurs niveaux :

* Parmi les repositories distants il y en a des publics et des privés (repository d’entreprise par exemple qui met à disposition des artifacts au sein de l’entreprise). Les repositories mettent à disposition des librairies (des fichiers JAR mais pas seulement) en téléchargement. On appelle ces libraires artifacts.
* Un repository local est une copie en local de repository distant contenant tous les artifacts téléchargés dans le passé (mise en cache).

Après avoir rajouté les dépendances pour le projet, on valide la cohérence du pom ( « mvn validate ») et ramènera les dépendances sur le projet. Il ira ensuite chercher le ou les repositories configurés puis ira les copier dans notre repo local.

Toutes les dépendances ne doivent pas forcément être utilisées de la même manière dans le processus de build ou lors de l'exécution de l'artéfact. Maven utilise la notion de portée (scope) pour préciser comment la dépendance sera utilisée.

Pour mettre à disposition notre artifact sur le repo distant configuré, on applique la commande « mvn deploy ».

Qu’en à la commande « mvn install », elle va compiler, exécuter les tests puis déployer dans le repo local. On peut donc tester notre dépendance en local avant de la mettre à dispo avec « mvn deploy ».

1. Qu’est ce qu’une application classique Entreprise Java Application ? Donner un exemple d’usage avec web service, JSP, JavaBean, EJB et JMS.

Java EE signifie Java Enterprise Edition, qui signifie essentiellement les applications d'entreprise. Cela comprend le stockage sécurisé des informations ainsi que leur manipulation et leur traitement. Ces applications peuvent avoir plusieurs interfaces utilisateur, telles que l'interface Web d'un client, accessible sur Internet, et une interface graphique s'exécutant sur les ordinateurs de l'entreprise sur le réseau privé de l'entreprise. (factures clients, calculs d'amortissement, réservation de vols, etc)

1. Qu’est ce que Spring ? qu’apporte Spring boot vis-à-vis de Spring

Spring est un framework open source ayant pour but de construire et définir l’infrastructure d’une application Java, dont il facilite le développement et les tests.

SpringBoot est une pré configuration d’un environnement Spring afin de créer et exécuter un projet Java rapidement.

1. ? Qu’est ce que Spring Boot ? Quels sont les points communs/différences entre JEE et Spring Boot ?

SpringBoot est une extension de l’utilisation de Spring permettant la pré-configuration d’application Spring autonomes facilement exécutable. La plupart des applications SpringBoot nécessitent très peu de configuration Spring.

SpringBoot est un Spring « déjà » configuré, alors que Spring lui-même a beaucoup de modules à configurer, ce qui peut être long à faire. La simplification de la configuration est aussi un atout majeur de SpringBoot, alors que Spring a énormément de chose customisable ou de configuration. Spring est basé sur l’injection de dépendances, alors que SpringBoot utilise un « Starter », des modules déjà configurés (Gestion de dépendances). SprinigBoot utilise beaucoup les annotations et un serveur embarqué (Apache Tomcat)

1. Qu’est ce qu’une annotation ? Quels apports présentent les Annotations ?

Les annotations SpringBoot sont une forme de métadonnées qui fournissent des données sur un programme. En d'autres termes, les annotations sont utilisées pour fournir des informations supplémentaires sur le programme. Cela ne fait pas partie de l'application développée. Cela n'a aucun effet direct sur le fonctionnement du code qu'ils annotent. Cela ne change pas le fonctionnement du programme compilé non plus. En fait cela nous permet d’ajouter des objets dans un contexte d’application sans être intrusif dans le code des classes concernées.

1. Comment fait-on pour créer un Web Service Rest avec Spring Boot ?

* On crée le modèle de donnée
* On ajoute le controller (Annotation @RestController) et une méthode qui va prendre en charge la requête. L’annotation @RequestMapping(‘’/…’’) de la méthode créée fait le mapping entre le path et notre méthode.

1. Qu’est ce qu’un container de Servlet ? Comment fonctionne un container de Servlet ?

Un conteneur de servlet est une implémentation de la spécification Java Servlet, utilisée principalement pour exécuter des applications web reposant sur les technologies servlets et JSP (Java Server Pages).

1. Expliquer la philosophie « Convention over Configuration » de Spring boot ?

Spring a toujours privilégié « Convention over Configuration », ce qui signifie qu'il prend en compte la majorité des cas d'utilisation et qu'il s'en tient plutôt que de choisir une configuration exacte et les dépendances requises pour le développement d'une application spécifique.

1. Expliquer ce qu’il se passe lors de l’exécution «SpringApplication .run(App.class,args) » Qu’est ce qu’un DAO ?

Springboot va lancer l’exécution de l’application avec comme base, le fichier App.class.

1. En quoi est-ce intéressant d’utiliser ce pattern ? Qu’est ce qu’un Singleton ? Que permet de réaliser les Entity dans Spring boot ? Est-ce spécifique à SpringBoot ?

Le DAO (Data Access Object) est un modèle qui fournit une interface abstraite à un certain type de base de données ou à un autre mécanisme de persistance. En mappant les appels d'application à la couche de persistance, le DAO fournit certaines opérations de données spécifiques sans exposer les détails de la base de données. Cet isolement soutient le principe de responsabilité unique. Il sépare les données dont l'accès à l'application a besoin, en termes d'objets et de types de données spécifiques au domaine (l'interface publique du DAO), de la manière dont ces besoins peuvent être satisfaits avec un SGBD spécifique, un schéma de base de données, etc.

1. Combien d’instances avez-vous crées lors de l’usage de «Service » en Spring boot? Pourquoi ?

Une seule, SpringBoot gère un injecteur de dépendance, fonctionnement sous la forme d’un singleton par classe et donc par service.

1. Que fournit le CRUD Repository de Spring boot ? Que sont les CRUD ?

Le CrudRepository est une interface Spring Data pour les opérations CRUD génériques sur un référentiel d’un type spécifique. Il fournit plusieurs méthodes prêtes à l’emploi pour interagir avec une base de données ( Create, Read, Update et Delete).

1. Qui réalise l’implémentation de la méthode findByProperty lors de la création d’un repository en Spring Boot ?

C’est le développeur qui doit préciser l’implémentation du findByProperty en définissant une query

1. Comment gère -t-on les relations One To One, One to Many et Many to Many avec JPA ?

JPA définit des modèles de relation qui peuvent être déclarés par annotation. Les relations sont spécifiées par les annotations : @OneToOne, @ManyToOne, @OneToMany, @ManyToMany.

1. Qu’est ce qu’une Architecture SOA ? Qu’est ce qu’une architecture Micro Service ? Il y a-t-il d’autres architectures existantes ? Quels sont leurs avantages/inconvénients ?

L'architecture orientée services (SOA) est un style architectural qui prend en charge l'orientation services. Par conséquent, elle s'applique également dans le domaine de la conception de logiciels où des services sont fournis aux autres composants par des composants applicatifs, à travers un protocole de communication sur un réseau.

L'architecture des microservices (souvent abrégée en microservices) fait référence à un style architectural de développement d'applications. Les microservices permettent de séparer une grande application en parties indépendantes plus petites, chaque partie ayant son propre domaine de responsabilité.