The ultimate Guide to Spring Boot: Spring Boot for Beginners

<https://www.youtube.com/watch?v=Nv2DERaMx-4>

Gradle

-build and dependency management tool

-Groovy and Kotlin are programming languages

A screen shot of a computer code

Description automatically generated

-Possible because of Spring Web

-Built a REST API

-these are considered endpoint

Maven

-helps devs manage and build their projects

-builds/tests our project

A black screen with white text

Description automatically generated

A black background with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

Target Directory (15:00)

-directory that Maven uses where it puts all of the stuff that it’s processed

Persistence Layer 30:26

-Handle interaction with our DB

-Entities

* Java Objects which represents our domain and often map to tables in DB

Repository Pattern vs DAO (Data Access Objects) 30:50

Service Layer (31:40)

-To use all the functionality exposed by the persistence layer

-We don’t want our presentation layer talking to persistence, calls must always go by service layer

Presentation Layer

-Goal is to take all of the data as a result of the service layer and expose that to user (REST APIS)

Layers

-Persistence -> Service -> Presentation

Modularity (33:16)

4.1 Inversion of Control

Example:

1. A class depends on 3 other classes
2. One way is we can create these 3 classes with new keyword
3. When we change dependency, we need to change the class

So better way is not create those 3 classes inside but simply just declare the different interface and inject it? (Dependency Injection) Work example later

4.2 Beans (37:40)

-Remember, leave it to the framework to provide he concrete classes where we declare our interfaces

Beans is the Spring framework concrete classes

A screenshot of a computer

Description automatically generated

-Database Driver will make a Java database driver available for you

-JDBC (Java Database Connectivity)

* Low level API, meaning you interact with the DB by using SQL queries
* Spring JDBC builds open JDBC and make it easier

-JPA (Java Persistence API)

* Allows you to interact with a DB using Java objects
* Handles all of the generation of the SQL and the mapping to and from the Java objects
* Build on top of JDBC so a high level API
* You can swap DBs and not have to change a lot of the code
* Specification, the actual implementation used by the Spring Ecosystem is called Hibernate. Devs will use JPA/Hibernate synonymosuly

Hibernate

-refers to an ORM (Object relational mapper)

* Maps from SQL to Java Objects and back again

Later

21: learn how packages work

-What is Test Driven Development?

25:56

-running it by jar file?