@RestController This is the class which provides rest responses to HTTP requests which are rest requests.

Starting Spring Boot

1) Set Up default configuration.
2) Starts spring application context
3) Performs class path scain
4) Starts Tomcat Server

Lets add a controller

a) A Java class b) Marked with annotations
c) Has info about
- What URL access triggers it?
- What method to run when accessed?

The @RequestMapping maps only to the GET method by default. To map to other HTTP methods you'll have to specify it in the annotation
whatever you return will be converted to json

The generated JSON has key names corresponding to property names of the Topic class. The JSON values are the values of those properties.

Embedded Tomcat Server.
- Convenience - Servlet container config is now application config
- Standalone application
- Useful for microservice architecture

In Spring Business services are typically Singleton's. When the application starts up Spring creates an instance of this service and keeps that in memory.

Unit - 3

Index

⇒ Starting a Spring Boot App
→ Spring Initializer → Spring Boot CLI
→ STS IDE
⇒ Config

Customizing Spring Boot

application.properties → customize it to customize your spring application

To change server port
server.port = 808

Unit - 4

(4) Spring Data JPA : The Data Tier

JPA (Java Persistence API) it's basically a specification that lets you do ORM (Object-Relational Mapping) when you are connecting to a database.

ORM lets you map entity classes into SQL tables
Class ↔ Table

Spring Data JPA
@Entity marks the class as the row of table
@Id marks the data member as primary key