

SpringBoot

- Purpose: To build java application

Try to type code with me.

Pre-requisites:

OOP, classes, interfaces, inheritance, exception handling, collection framework

Must have installed:

JDK -> JDK 17 or higher to use springboot 3

IntelliJ IDE

- Provides large number of helper classes and annotations

The Problem with spring:

Traditional spring application building was tedious

Qs

1. Which JAR dependencies do I need?
2. How do I set up configuration? (xml or java)
3. How do I install the server? (Tomcat, JBoss etc)

& this is just getting started

- **SpringBoot is the Solution for this**
 - Easier for spring development
 - Minimize manual configuration (It performs the auto-configuration)
 - Resolve dependency conflicts
 - Provide an embedded HTTP server

- SpringBoot and Spring
 - SpringBoot uses Spring Behind the scenes.
 - SpringBoot simply makes it easier to use spring.

- Spring Initializer (SpringBoot provides it)

<http://start.spring.io>

- Quickly create a starter spring project
- Select dependencies
- Select maven/gradle
- Import project in IDE

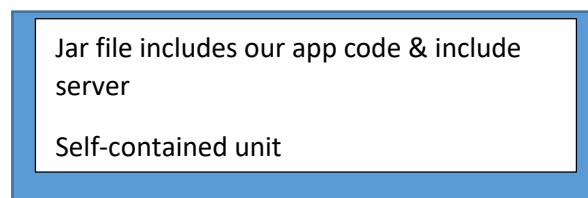
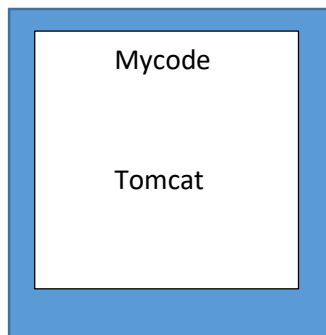
- SB Embedded Server

Provide embedded server

- Tomcat, JBoss, Undertow

No need to install server separately

firstapp.jar



FAQs

1. Does SB replace Spring MVC, Spring REST..?
No, Instead it uses these technologies
2. Does SB run code faster than regular Spring Code?
No, SB uses same code of spring framework

Maven:

- When building our project, we may need additional JAR files
Ex. Spring, Hibernate, JSON etc

1st Approach:

Download the JAR files from each project website

Manually add the JAR files to our build path/classpath

Maven is Solution

- Tell maven the projects we are working with(dependencies)
- Maven will go out and download the JAR files for those projects
- And Maven will make those JAR files available during compile/run
- We can say maven is our helper or personal shopper (shopping list)

Development Process:

1. Configure our project at spring initializer(dependency: Spring Web)
2. Download zip file
3. Unzip the file
4. Import the project into our IDE

Lets Create RestController

```
package com.flynaut.springboot.demo.firstapp.rest;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.time.LocalDate;

@RestController
public class FunRestController {

    //This method will handle GET request at "hello" endpoint
    @GetMapping("/hello")
    public String sayHello(){
        return "Hello Team!!!!!!";
    }

    @GetMapping("/date")
    public LocalDate date(){
        LocalDate localdate= LocalDate.now();
        return localdate;
    }

}
```

URL: Uniform Resource Locator

<http://localhost:8080>

<http://www.abc.com:8080/banking>

http: Application Layer Protocol (http : hypertext transfer protocol)

www.abc.com : DNS qualified host name/IP address(to resolve the host problem)

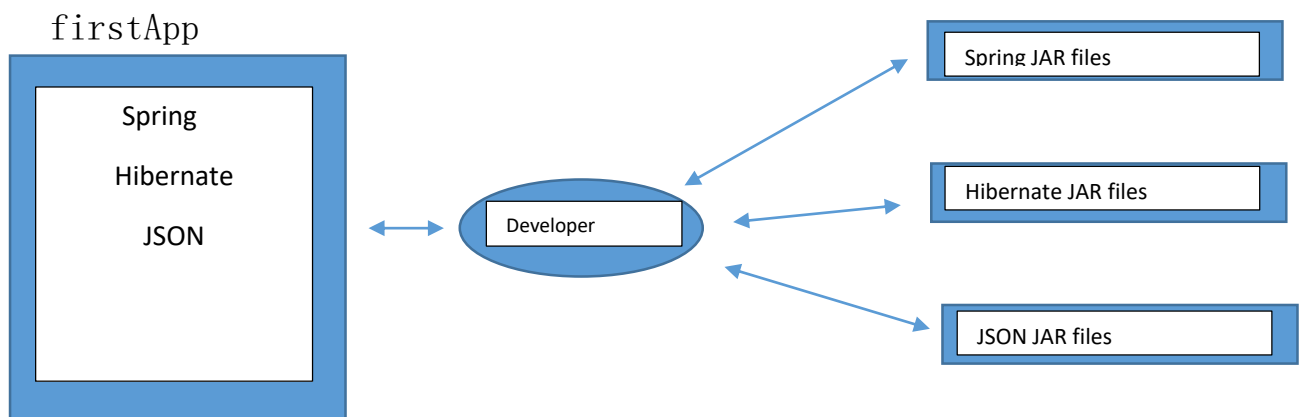
8080: TCP port (to identify the port)

/banking: path or URI (Uniform resource identifier)

Maven

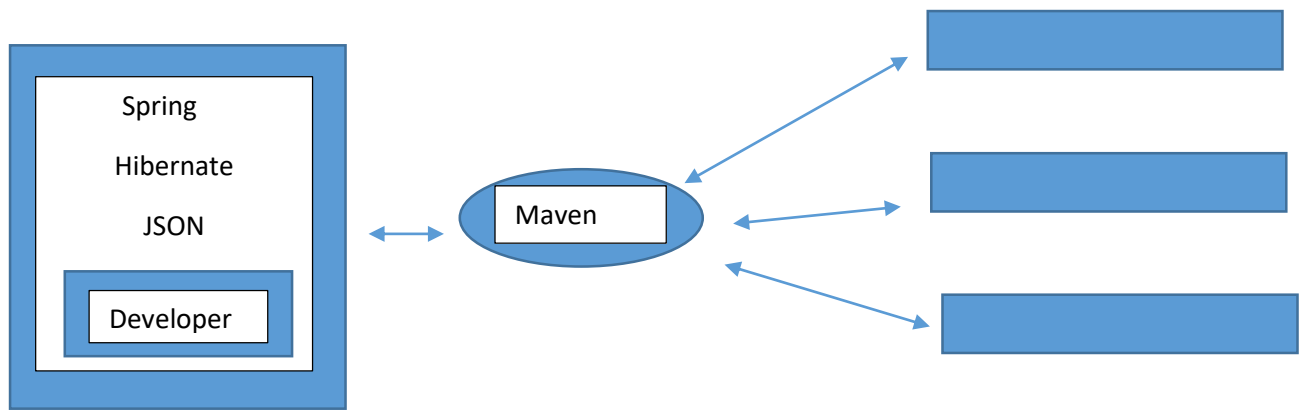
- What?
 - A project management tool(build tool)
 - Most popular use of Maven is for build management and dependencies
- What problems does maven solve?

1st Approach - Without using maven



2st Approach - With using maven

- Tell maven the projects we are working with (dependencies)
- Go out and download Jar Files for us



- **Maven Project Structure**

Maven follows standard directory structure.

- Normally when we join a new project
- Every development team used to make their own project directory
- And this is not ideal for new comers and not standardized

Directory	Description
<code>src/main/java</code>	Our java source code
<code>src/main/resources</code>	Properties/config files used by our app
<code>src/test</code>	Unit testing code and properties
<code>target</code>	Destination directory for compiled code(Automatically created by maven)
<code>pom.xml</code>	Maven configuration file

POM. xml

Project Object Model file

- Configuration file for our project
- Basically our “Shopping List” for Maven ☺

Located at root of maven project

- POM file Structure

pom. xml

