Art of Heavy lifting with Spring Boot

By Sunit Parekh

Challenges

Repetitive Boilerplate Code

- Every NEW project has to define all the common stuff again and again
- Automated build and deployment is always a challanging
- Dependency management is nightmare to manage across projects

Too Much Configuration

- Configurability is good design, in Spring Core everything is configurable. However, default settings also needs to be configured
- Too many options for configuration

Build and Deployment

- Manually take care of compatible dependencies
- Leverage on IDE support for running application
- How to run application locally and downstream environments?

What we need?

Full Stack Framework

- like **Rails** in Ruby or **django** and Python or Symphony in PHP
- Heavy and blots the application with too many components
- Sometime not better way to have different approach in implementation
- What you get is what you can use

Sprint Boot

- Reduce repetitive boilerplate code for every new project
- Provides default configuration with inbuilt beans with env specific property files
- Embed Tomcat, Jetty or Undertow support for running application
- Automated way of build and deployment (Cloud Support)
- Inbuilt support for production-ready features such as metrics, health checks
- Better support for Unit Testing
- Framework with convention over configuration

How to get started?

Spring Boot Starters

- Include only what you want using Maven POM
- Provides default implementation with capability of configurability and override ability

https://github.com/spring-projects/spring-boot/tree/master/spring-boot-starters
http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using-boot-starter

Live Coding

Spring Boot to Build RESTful Web Services

Live Coding

Spring Boot Properties and Profiles

Live Coding

Unit Testing with Spring Boot

Unit Testing Support

- First Class support for Unit Testing with mocking support
  - https://spring.io/blog/2016/04/15/testing-improvements-in-spring-boot-1-4
  - http://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-testing.html
-

Unit Tests

- AAA of Unit Testing
- Clear separation between Arrange, Act and Assert
- Give descriptive method name for each test
- One test one scenario
- Focus on Readability

```
@Test
public void longDescriptiveName() {

    // arrange  (Setup)
    // Arrange all necessary preconditions and inputs.

    // act      (Execute)
    // Act on the object or method under test.

    // asset  (Verify)
    // Assert that the expected results have occurred.

}
```

Live Coding

Working with Databases using Spring Boot

Live Coding

Spring Boot Actuator for Production Support

Live Coding

Secure Development with Spring Boot

Live Coding

Build and Deployment with Spring Boot

Live Coding

Spring Boot working with JMS

Spring Maven Plugin

- Creates executable Jar packaging with ability to run as linux service
- Goals
  - `spring-boot:run`

http://docs.spring.io/spring-boot/docs/current/maven-plugin/

Read More **Code** Less

by Sunit

Code on Github

https://github.com/morning-school/spring-boot-demo

Spring Boot Resources

- Spring Boot Properties http://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html
- Sample Projects https://github.com/spring-projects/spring-boot/tree/master/spring-boot-samples
- Spring Boot Dependencies http://docs.spring.io/spring-boot/docs/current/reference/html/appendix-dependency-versions.html

Future Reading

- https://www.infoq.com/presentations/spring-boot-web
- https://www.youtube.com/watch?v=FXJPA3IqANg
- http://docs.spring.io/spring-boot/docs/1.4.1.RELEASE/reference/htmlsingle/
- https://spring.io/guides/gs/spring-boot/
- https://spring.io/guides/gs/rest-service/

Thanks!

Like to reach us,

Sunit Parekh
parekh.sunit@gmail.com

Sarthak Makhija
sarthak.makhija@gmail.com