API Workshop

Build REST APIs with Spring Boot

O'REILLY®
Software Architecture

AGENDA

- Spring Framework Introduction
- Spring Boot Features
- Web services, APIs and REST
- Lab 1 - Build a ToDo list application using Spring Boot
- https://github.com/jpgough/api-workshop

WHAT IS SPRING?

"Make the right thing easy to do"

Rod Johnson

WHAT IS SPRING?

- **Dependency injection** framework for Java
- Lightweight, Open Source
- Layered framework
- Simplifies application development
- Powerful, consistent **abstractions** for common patterns

- **Data Access / Transaction Management** support (JTA, JDBC, Hibernate, JPA)
- **Web Frameworks** - **Spring MVC** and the newer **WebFlux**
- **Unit / Integration testing** using mocks
- **JVM Multiple language support** (Java / Groovy / Kotlin)

BUILDING SERVICES USING JAVA

IN THE OLD(ISH) DAYS

<XML />

<XML />

<XML />

GOALS OF SPRING BOOT

- Make it easy to create stand-alone **production-grade** Spring applications
- Facilitate rapid development through Spring Boot Starters
- Provide common non-functional features
  - Embedded servers
  - Security
  - Metrics
  - Health checks
  - Externalized configuration

WHAT IS SPRING BOOT?

Opinionated

<XML />

Stand Alone

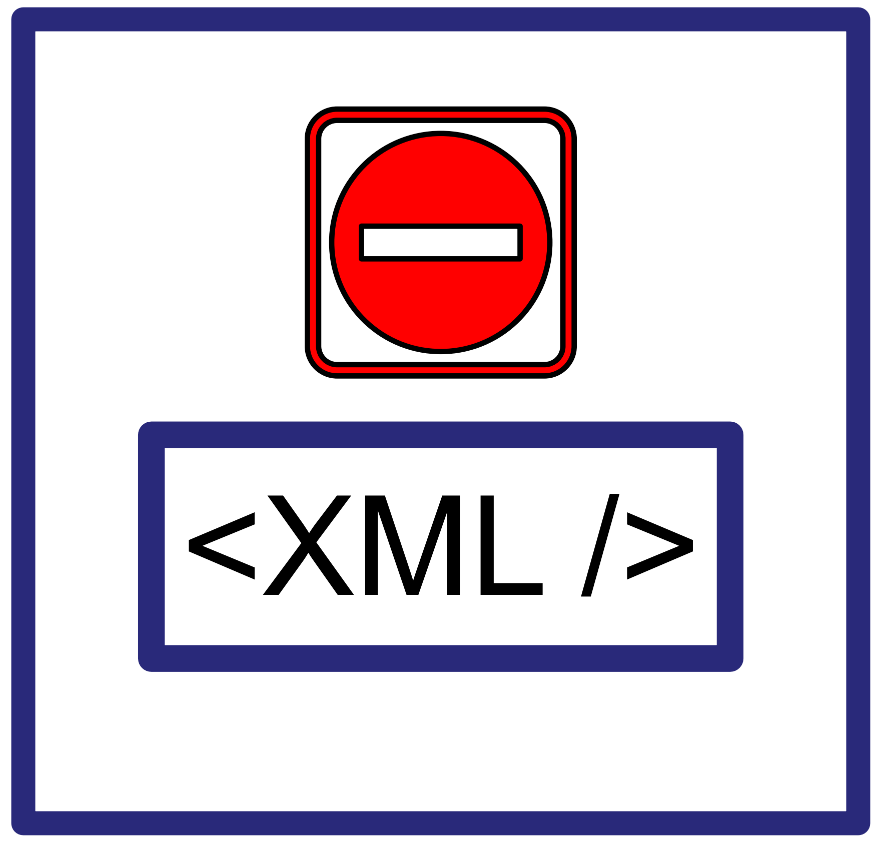
OPINIONATED

- Opinionated view of Spring Platform and third-party libraries
- Favours convention over configuration
- Reduces need to write boiler-plate code
- Quick to start-up, Hello World in a few minutes!

WARNING: hidden complexity!

AUTO CONFIGURATION

- Including a library dependency will introduce intelligent auto configuration
- Favours Java-based configuration over XML
- Convenience Annotations
- `@SpringBootApplication equivalent to:`
  - `@EnableAutoConfiguration`
  - `@ComponentScan`
  - `@Configuration`

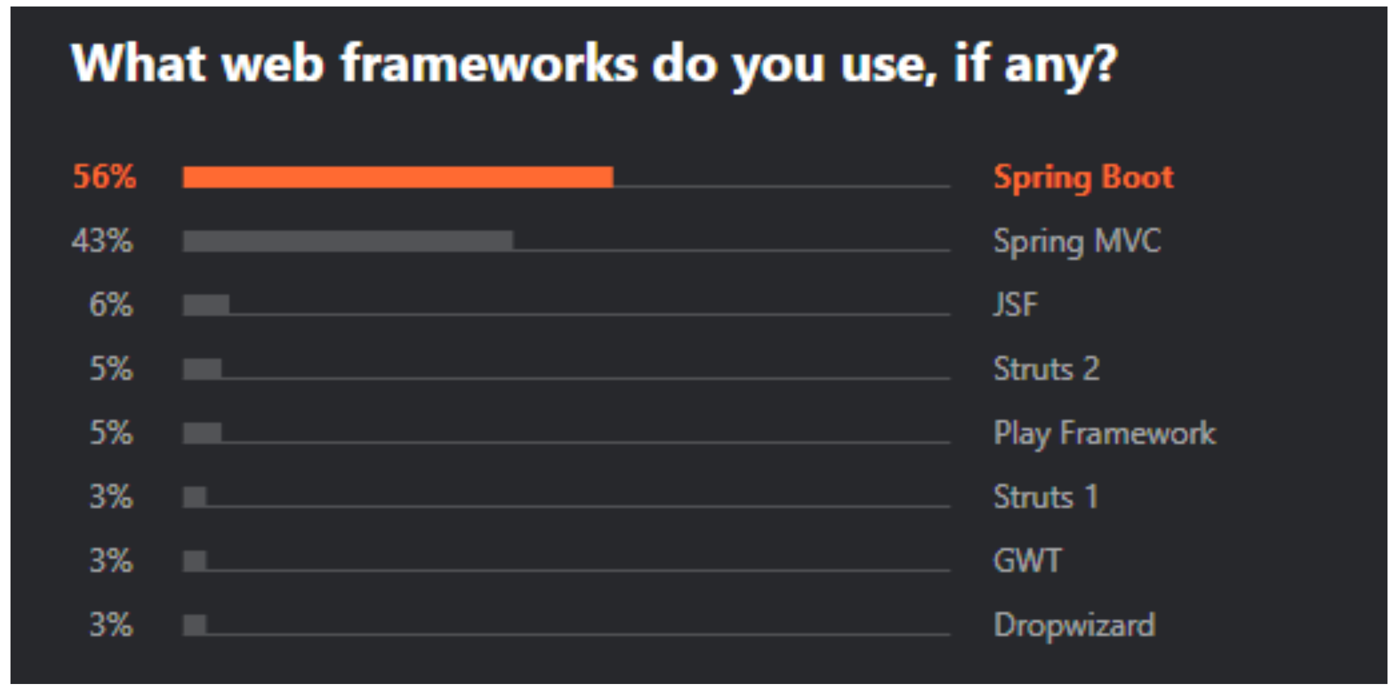


STANDALONE

- **Flexible Packaging**
  - Self-contained executable jars with embedded web server (vs)
  - Traditional WAR files
- **Multiple deployment options**
  - Cloud-Native platforms
  - Container Images (Docker)
  - Virtual / Real machines

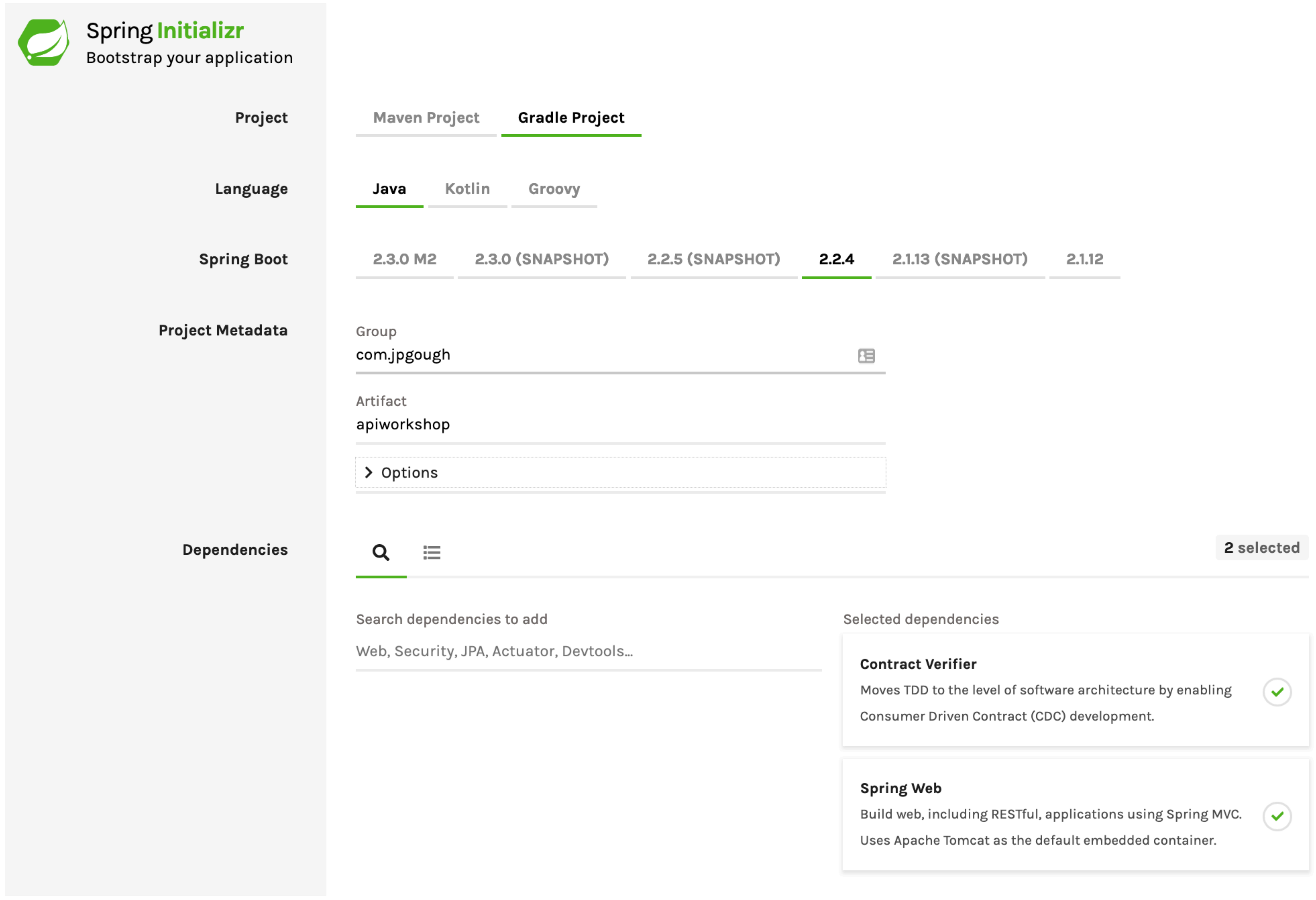
WHY USE SPRING BOOT?

- Based on mature Spring Framework (known for stability, backwards compatibility)
- Open Source, active community support
- ThoughtWorks Technology Radar - **ADOPT** in 2016 - "*If you live in a Spring ecosystem and are moving to microservices, Spring Boot is now the obvious choice*"
- JetBrains Developer Ecosystem survey (2019)
  - 56% use Spring Boot for Web Application development
  - 61% use Spring Boot as an alternative to application servers



https://www.thoughtworks.com/radar/languages-and-frameworks/spring-boot
https://www.jetbrains.com/lp/devecosystem-2019/java/

Spring Initializr - http://start.spring.io



Spring Initializr
Bootstrap your application

Project: Maven Project [✓ Gradle Project]

Language: [✓ Java] Kotlin Groovy

Spring Boot: 2.3.0 M2 2.3.0 (SNAPSHOT) 2.2.5 (SNAPSHOT) [✓ 2.2.4] 2.1.13 (SNAPSHOT) 2.1.12

Project Metadata
Group: com.jpgough
Artifact: apiworkshop
Options

Dependencies
2 selected

Search dependencies to add
Web, Security, JPA, Actuator, Devtools...

Selected dependencies

Contract Verifier ✓
Moves TDD to the level of software architecture by enabling Consumer Driven Contract (CDC) development.

Spring Web ✓
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

WEB SERVICES, APIS & REST

- SpringBoot makes it straightforward to build and deploy HTTP based web services
- Web services expose business data and capabilities via Application Programming Interfaces (APIs)
- REST can be used over HTTP to provide structure to those APIs

Spring Web spring-boot-starter-web	Starter for building web applications using Spring MVC. Uses Tomcat as the default embedded servlet container
Spring Reactive Web spring-boot-starter-webflux	Starter for building event-driven reactive web applications with Spring WebFlux and Netty

WHAT IS REST?

- **RE**presentational **S**tate **T**ransfer - Software Architecture style with guiding constraints
  - Client-server Architecture - Separation of concerns, allows components to evolve independently
  - Statelessness - No client context saved on server
  - Cacheability
  - Layered System
  - Code on Demand (optional)
  - **Uniform Interface** -
    - Resources identified using URIs
    - Resource manipulation through representation
    - Self-descriptive Messages
    - HATEOAS (hyperlinks based access of resources)

RESTFUL WEB SERVICES

- Web Service APIs adhere to REST constraints
- HTTP - commonly used as the transport layer
- URLs identify resources
  - http://www.api-workshop.com/todos
  - http://www.api-workshop.com/todos/1
- Operations
  - Create, Read, Update, Delete supported by HTTP Methods

REST OVER HTTP

- HTTP verbs for operations
  - POST, GET, PUT, DELETE
- HTTP status codes represent the result of a request
  - HTTP 2xx - Success
  - HTTP 3xx - Redirection
  - HTTP 4xx - Client errors
  - HTTP 5xx - Server errors

Lab 1 - Spring Boot

Build a ToDo list application using Spring Boot

https://github.com/jpgough/api-workshop

SPRING BOOT ANNOTATIONS

- `@RestController - HTTP Requests handled by this controller`
- `@RestController equivalent to:`
    - `@Controller`
    - `@ResponseBody`
- `@GetMapping - Maps HTTP GET Requests`
- `@PostMapping - Maps HTTP POST Requests`

EXAMPLE: GET ALL TODOS

Request: **GET http://www.api-workshop.com/todos**

HTTP Headers:
- **Accept**: application/json

Response: **200 OK**

```
{
  "todos": [
    {
      "id": 1,
      "description": "Attend API workshop",
      "done": false
    }
  ]
}
```

EXAMPLE: GET WITH QUERY PARAMETERS

Request: **GET http://www.api-workshop.com/todos?done=false**

HTTP Headers:
- **Accept**: application/json

Response: **200 OK**

```
{
  "todos": [
    {
      "id": 2,
      "description": "Learn about SpringBoot",
      "done": false
    }
  ]
}
```

EXAMPLE: CREATE A TODO

Request: **POST http://www.api-workshop.com/todos**

HTTP Headers:
- **Content-Type**: application/json

```
{
  "description": "Learn about REST APIs"
}
```

Response: **201 CREATED**

```
{
  "id": 3,
  "description": "Learn about REST APIs",
  "done": false
}
```