

Practical Spring Boot: For the Non-Enthusiast

@GROOVYPGH



def me =

```
[  
    name : 'Billy Conner, Jr.',  
    title: 'Software Engineer, Alpha Technologies',  
    affiliation: 'Pittsburgh Groovy Meetup (GroovyPGH)',  
    email : 'billyconnerjr@groovypgh.org',  
    twitter : '@billyconnerjr'  
]  
print me.values().each{ println it }
```

Introduction



What is Spring Boot?

Java-based Framework

Designed to get you up and running quickly

Simplifies the **bootstrapping** of Spring Applications

Hold on...What is Spring?

Includes many different features for Java-based Enterprise Applications:

- Spring MVC Web applications
- Data Access
- Security (Authorization and Authentication)
- Transaction Management
- Messaging
- Testing support
- many more...



So...What is Spring Boot?

- No need for abundant XML configuration
This was a big criticism of Spring applications.
- Spring Boot takes an opinionated approach to the configuration
- Spring Boot favors **annotations** over XML
- **No war** files to configure
- Embedded Apache



In other words...

As described in his blog, Phil Webb, the Project Lead for Spring Boot..



How Can 'I' use Spring Boot?

IF:

- I don't work in a Spring Environment
- I'm strictly a Front-end Developer
- I don't have the resources to create a full-stack (such as a server or database)

Use cases

- Build prototypes applications quickly
- Build an API for your front-end to consume
- Build full blown MVC application quickly

Diving in



Getting Spring Boot

Requirement: Java Development Kit 1.6+ (1.8 recommended)

Via Software Version Manager

- *sdkman – a software development kit manager
<http://sdkman.io/>

Via Dependency Management System:

- Gradle build - included as a dependency
- Maven build – included as a dependency

*my personal recommendation

Running Spring Boot

- Using Gradle build
- Using Maven build
- Spring boot itself ships with a Command Line Interface (CLI)
- Add the spring-boot.*.jar to your Java Class path
Spring Boot is simply a java library
- Using an IDE such as IntelliJ or Eclipse



Supported Languages

Officially supported:

- Java
- **Groovy**
- Kotlin

Plugins and modules allow:

- Scala
- Jython
- And many others

Bare Minimum Application

Using Groovy:

```
@RestController  
  
class App {  
    @RequestMapping('/')  
    def home(){  
        'Hello, I am a Spring Boot Web Application!'  
    }  
}
```



★ // Using the spring boot CLI and groovy, run:
// spring run app.groovy

Demystified

- Embedded Apache Server Started
- @RestController - signifies Spring MVC to look for web routes and write the results to an HTTP response
- @RequestMapping – maps the method home () to the default route “/”
- In this example the **Groovy** syntax allows the exclusion of a return statement and semi-colons “;”

Lets get more interesting

Including another route:

```
@RequestMapping('/create/{name}')  
def nameDoesntMatter(@PathVariable def name){  
    "Hello ${name}, Your account has been created"  
}  
  
// @PathVariable binds 'name' to URI template variable  
  
//Groovy provides the string interpolation for ${name}  
// alternate: "Hello" + name + "thank you for visiting"
```



Further more

A route with this context implies a POST request.

```
@RequestMapping(value = "/account/{name}",  
method=RequestMethod.POST)  
  
def nameDoesntMatter(@PathVariable def name){  
    "Hello ${name}, Your account has been created"  
}
```

Starting to look like a basic rest service!



What is Groovy?

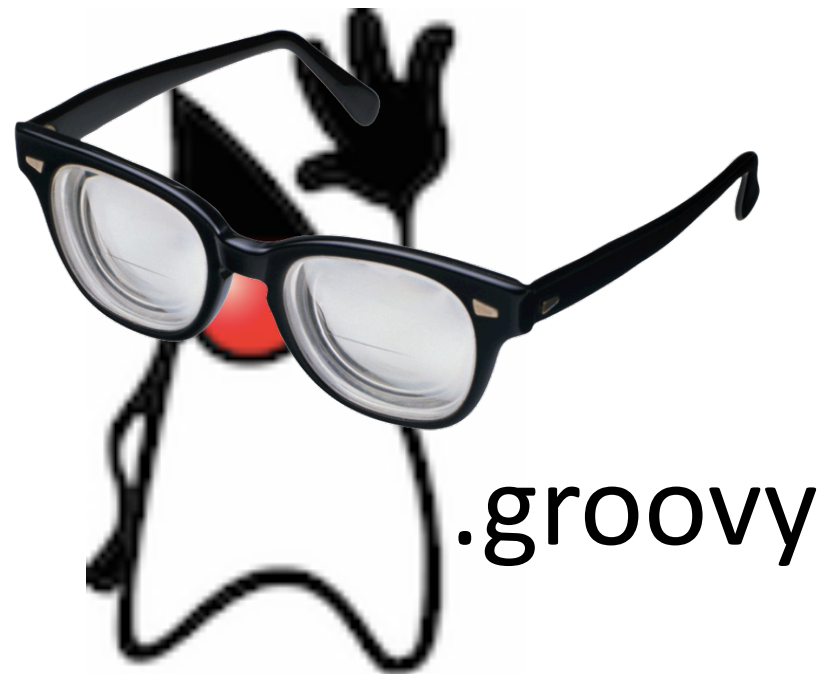
Sits on top of the Java platform

‘Java plus more’

Compiles to Java bytecode

Groovy can run Java classes!

Groovy can run all Java libraries!



Java

```
public class Person {  
    private String name;  
    private int age;  
  
    public Person() {}  
  
    public String getName() {  
        return name;  
    }  
    public int getAge() {  
        return age;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
}
```

Groovy

```
class Person {  
    String name  
    int age  
}
```



Spring_INITIALIZER

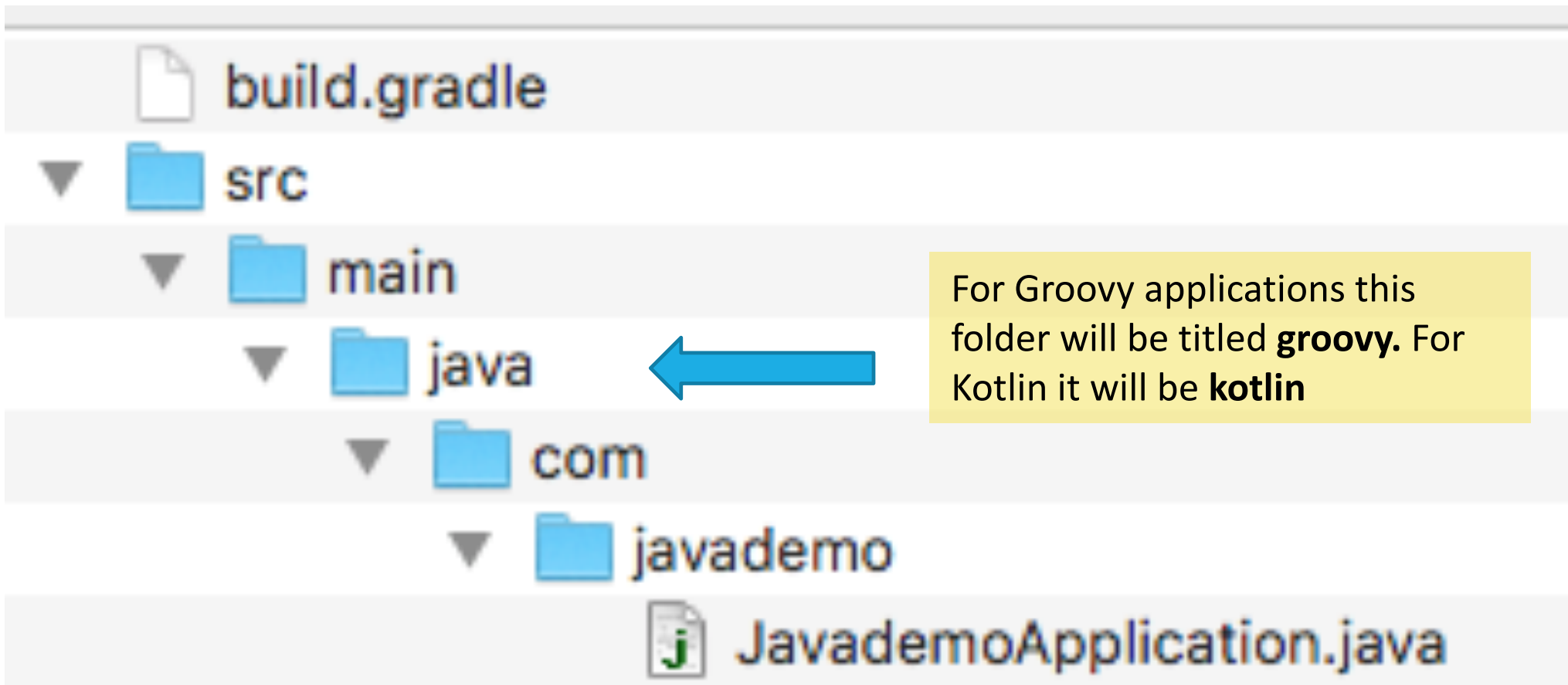
Visit <http://start.spring.io/>

Allows you to create the skeleton Spring Boot application

- Gradle build script and wrapper
- Folder structure

IMPORTANT





Formal Application folder Structure

Demo: book service 1



Demo: book service 2

Wrote three files(Main, Model, Controller)

Specifying acceptable method

Automatic Jackson marshalling and un-marshalling

http codes generated

Demo: book service 3

Spring data jpa as a dependency

Defining entities and repository

Free methods

- findOne(ID id)
- findAll()
- save(Iterable<S> entities)
- save(S entity)
- exists(ID id)
- count()
- multiple delete methods

Demo: book service 4

auto loading data

endpoints for books and authors

query methods

CORS annotation

Query Method Keywords

Example: `findByAgeGreaterThan(21)`

And

Or

Is,Equals

Between

LessThan

LessThanEqual

GreaterThan

GreaterThanEqual

After

Before

IsNull

IsNotNull,NotNull

Like

NotLike

StartingWith

EndingWith

Containing

OrderBy

Not

In

NotIn

True

False

IgnoreCase

Persisted Data: Using H2

Currently using in memory db

View in web console <http://localhost:8080/h2-console>

use correct url reference from application.properties

- jdbc:h2:mem:mydb

Persisted Data: Using MySQL

Include the MySQL driver as a dependency (Gradle example below):

```
dependencies {  
    compile("org.springframework.boot:spring-boot-starter-data-jpa")  
    compile("com.h2database:h2") // remove the in memory database  
  
    compile ("mysql:mysql-connector-java")  
    ...  
}
```

Update the applications.properties file with database information

```
spring.datasource.url=jdbc:h2:mysql://hostname/dbname  
spring.datasource.username: yourDbUsername  
spring.datasource.password: yourDbPassword  
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect  
spring.jpa.hibernate.ddl-auto=none
```

Where to go now

Use this service for development simulations

You can begin build this into production level application

provide executable jar for distribution

GroovyPGH



@groovypgh



Pittsburgh Groovy Programming Meetup

<http://www.meetup.com/Pittsburgh-Groovy-Programming/>



Spock Framework



Welcome other JVM Languages



and more...



Resources

Code for this demo:

<https://github.com/billyconnerjr/springboot4nonenthusiast>

Spring Boot Initializer

<http://start.spring.io>

Spring Boot Documentation

<https://spring.io/docs>

Spring Boot Guides

<https://spring.io/guides>

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

Thank you!