

Running Hadoop Apps with Spring Boot

Thomas Risberg
Pivotal Software
trisberg@pivotal.io

Code:

<https://github.com/trisberg/bostonhadoop>

What we need to have installed

- **Gradle 2.1**
<http://www.gradle.org/>
- **Spring Boot 1.1.8**
<http://projects.spring.io/spring-boot/>

What else do we use?

- **Apache Hadoop 2.4.1**
<http://hadoop.apache.org>
- **Spring for Apache Hadoop 2.0.2-hadoop24**
<http://projects.spring.io/spring-hadoop/>
- **Apache Spark 1.1.0**
<http://spark.apache.org>

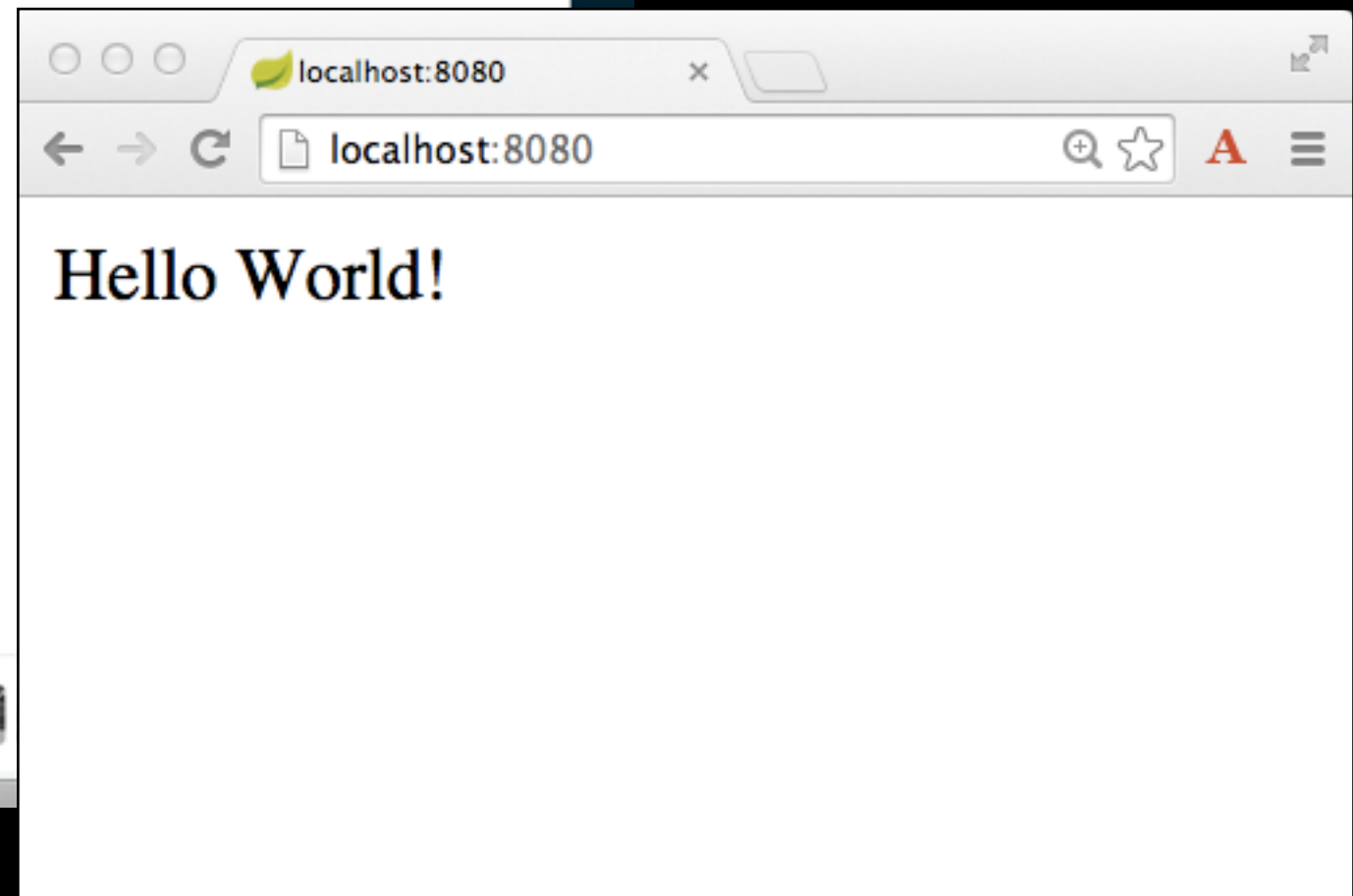
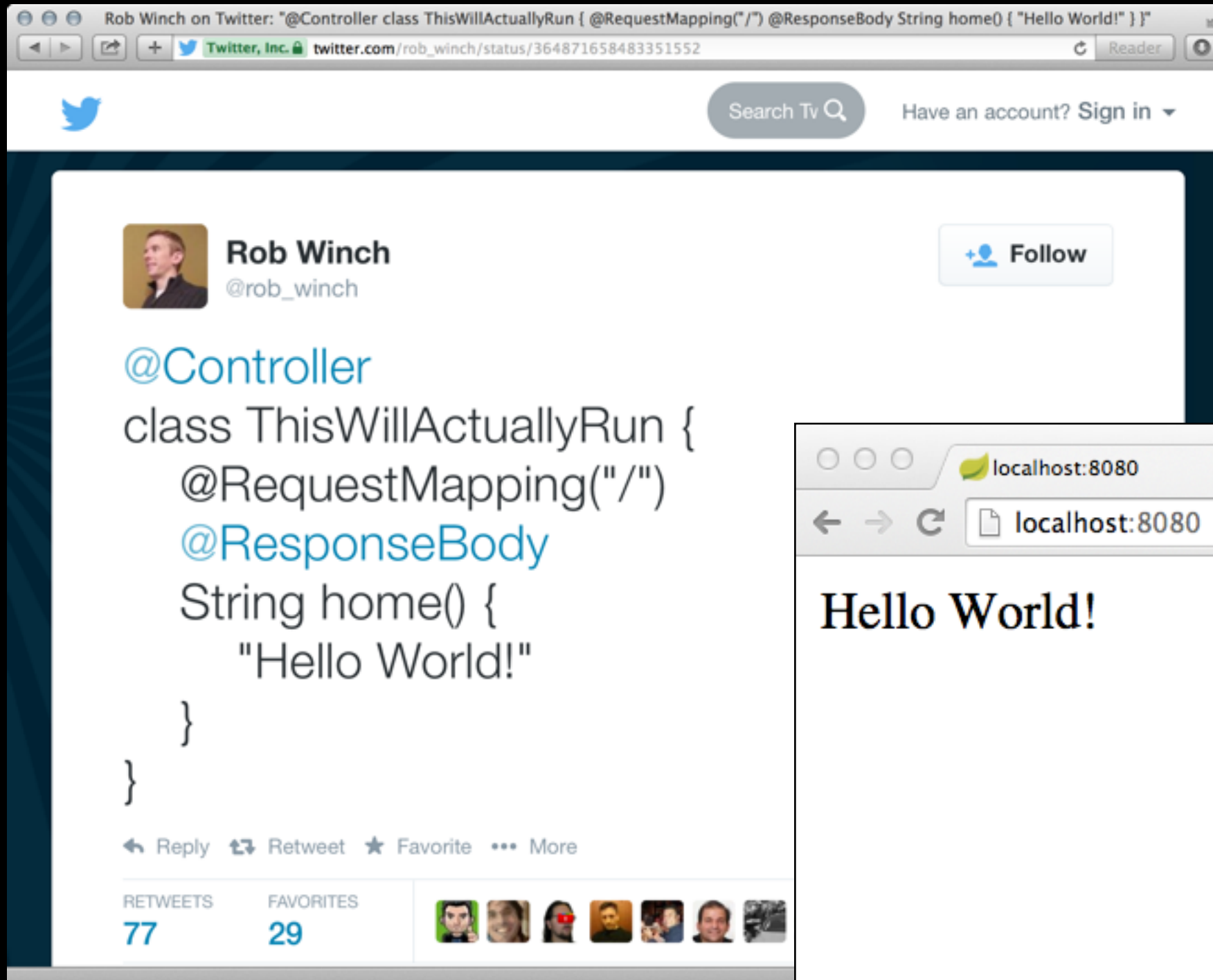
Spring Framework

- **Since 2003** the goal has been to simplify enterprise Java development
- **POJO** based programming model providing abstractions over many enterprise APIs
- **Dependency Injection** helps to improve configuration and testability
- **Declarative transaction management** makes your code simpler and easier to test
- **MVC framework** originally an alternative to Apache Struts, has survived and evolved
- **Spring Batch** addresses common scenarios for batch processing
- **Spring Integration** for implementing enterprise integration patterns using messaging

Spring Boot

- **Addresses complexity** in configuring Java/Spring applications
- **Standalone or Web apps** using embedded web server or deployed as WAR file
- **Spring Boot starters** combine common dependencies in a single dependency that you add to your Maven or Gradle builds
- **Autoconfiguration** uses conditional configuration in Spring 4 and makes opinionated decisions, based on classpath content, configuring beans your application should need
- **Command-line interface (CLI)** let's you use the Groovy programming language together with autoconfiguration to make Spring application development super simple
- **Actuator** adds management features to a Spring Boot app

Tweet Driven Development



Demos

- **Hello Groovy**
- **YARN**
- **Batch**

Hello Groovy

```
@Grab('org.springframework.data:spring-data-hadoop:2.1.0.M1-hadoop24')

import org.apache.hadoop.fs.FileStatus
import org.springframework.data.hadoop.fs.FsShell

public class App implements CommandLineRunner {

    @Autowired FsShell fsShell

    void run(String... args) {
        println("Hello!")
        for (FileStatus fs : fsShell.ls("/")) {
            println("> ${fs.path.name}")
        }
    }

    @Bean FsShell fsShell() {
        org.apache.hadoop.conf.Configuration conf =
            new org.apache.hadoop.conf.Configuration()
        conf.set("fs.defaultFS", "hdfs://borneo:8020")
        return new FsShell(conf)
    }
}
```


YARN Demo

```
@Configuration
@ComponentScan
@EnableAutoConfiguration
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @YarnComponent
    @Profile("container")
    public static class HelloPojo {
        private static final Log log = LoggerFactory.getLog(HelloPojo.class);

        @OnContainerStart
        public void onStart() {
            log.info("Hello from YARN!");
        }
    }
}
```

Batch Demo

```
@ComponentScan
@EnableAutoConfiguration
@ImportResource("batch-hashtags.xml")
@EnableBatchProcessing
public class Application {
```

```
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
```

```
}
```

```
<batch:job id="job">
```

```
    <batch:step id="import" next="hashtags">
        <batch:tasklet ref="scriptTasklet"/>
    </batch:step>
```

```
    <batch:step id="hashtags" next="export">
        <batch:tasklet ref="hashtagsTasklet"/>
    </batch:step>
```

```
    <batch:step id="export" parent="export-step"/>
```

```
</batch:job>
```

Spring XD

- **The project's goal** is to simplify the development of big data applications
- **Distributed and extensible system for:**
 - data ingestion
 - real time analytics
 - batch processing
 - data export

Links

- **Spring Boot** <http://projects.spring.io/spring-boot/>
Spring Initializr <http://start.spring.io/>
- **Spring for Apache Hadoop**
<http://projects.spring.io/spring-hadoop/>
- **Spring XD** <http://projects.spring.io/spring-xd/>
- **Code and slides**
<https://github.com/trisberg/bostonhadoop>
- **GVM** <http://gvmtool.net/>
- **Groovy** <http://groovy-lang.org/>