



QUOIN

WE SIMPLY BUILD GREAT SOFTWARE

704-918-1799 |
www.quoininc.com

Spring Boot Web Services

Boston

New York

Charlotte

Washington, D.C.

Nicaragua

**209 Delburg Street, Suite
207
Davidson, North Carolina
28036**

Introduction

Looking for an easy way to implement a RESTful web service in Java? Spring Boot by Pivotal Software, Inc gives you everything you need. This presentation will show you how to set up a simple RESTful web service using Spring Boot. Saying hello world and getting a response has never been easier.

Spring Boot Features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' POMs to simplify your Maven configuration
- Automatically configure Spring whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

Presentation Focus

- Simple Spring Boot Restful web service using Maven
- Out of scope this time:
 - Spring Boot CLI
 - Gradle Project
 - SOAP web services

Project Setup

- IDE: Eclipse with Spring IDE extension or Spring Tool Suite
- Create New Maven project
- Create a class with a main
- Create a controller class
- Populate pom.xml
- Application.properties

pom.xml Highlights

- Parent: spring-boot-starter-parent
- Dependencies:
 - Spring-boot-starter-web (required)
 - Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container.
 - Spring-boot-starter-actuator
 - Starter for using Spring Boot's Actuator which provides production ready features to help you monitor and manage your application (/env,-> /configprops)
 - Spring-boot-starter-security
 - Starter for using Spring Security

Main class: Application.java

```
package quoininc.helloworld;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.PropertySource;

@SpringBootApplication
@PropertySource(value = {"file:/etc/quoininc/helloworld/application.properties"})
public class Application {

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

Controller class: HelloController

```
package quoininc.helloworld.controllers;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/v1/heyhey")
public class HelloController {
    @RequestMapping(value="/hollaback", method=RequestMethod.GET)
    public String hollaBackEnthusiastically(final String helloMessage) {
        return helloMessage.toUpperCase() + "!!!";
    }
}
```


Application.properties (non-SSL)

#Spring Security

security.user.name = testing

security.user.password = moretesting

#SSL info

server.port = 8800

Application.properties (SSL)

#Spring Security

security.user.name = testing

security.user.password = moretesting

#SSL info

server.port = 8443

server.ssl.key-store = /home/jwelch/helloworld.p12

server.ssl.key-store-password = p@\$w0rD

server.ssl.keyStoreType = PKCS12

server.ssl.keyAlias = helloworld