

arnaud.nauwynck@gmail.com

Introduction to SpringBoot (Spring Framework @AutoConfigure...)

This document:
<http://arnaud-nauwynck.github.io/docs/Intro-SpringBoot.pdf>

I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions.



springboot



All

Images

Videos

News

Books

More ▾

Search tools

About 1,500,000 results (0.39 seconds)

Did you mean: **spring boot**

Spring Boot - Projects

<https://projects.spring.io/spring-boot/> ▾

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring ...

Spring Boot Reference Guide

docs.spring.io/spring-boot/docs/current/reference/htmlsingle/ ▾

This section provides a brief overview of **Spring Boot** reference documentation. Think of it as map for the rest of the document. You can read this reference guide ...

Getting Started · Building an Application with Spring Boot

<https://spring.io/guides/gs/spring-boot/> ▾

This guide provides a sampling of how **Spring Boot** helps you accelerate and facilitate application development. As you read more Spring Getting Started guides ...

GitHub - spring-projects/spring-boot: Spring Boot

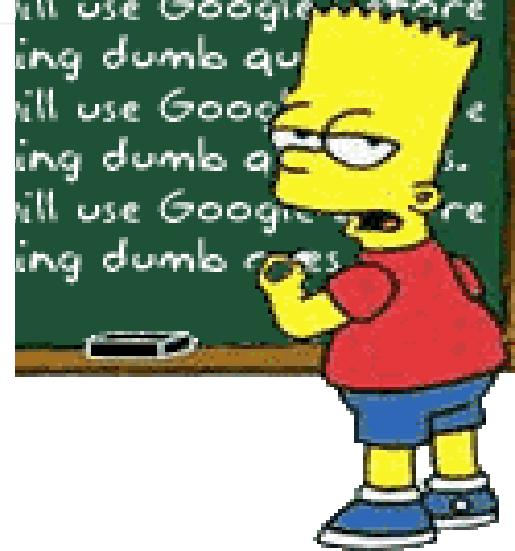
<https://github.com/spring-projects/spring-boot> ▾

Spring Boot makes it easy to create Spring-powered, production-grade applications and services with absolute minimum fuss. It takes an opinionated view of the ...

Spring Boot 1.4 Release Notes · spring-projects/spring-boot ...

<https://github.com/spring-projects/spring-boot/.../Spring-Boot-1.4-Release-Note...> ▾

Classes, methods and properties that were deprecated in **Spring Boot** 1.3 have been removed in this release. Please ensure that you aren't calling deprecated ...



Google Rank #1 “springboot”

<https://projects.spring.io/spring-boot/>

no Wikipedia?

The screenshot shows the Spring Projects website with the Spring logo at the top. The navigation bar includes links for DOCS, GUIDES, PROJECTS (which is highlighted in green), BLOG, QUESTIONS, and a search icon. Below the navigation, there's a section for 'PROJECTS' with a 'Spring Boot' card. The card contains a brief description: 'Takes an opinionated view of building production-ready Spring applications. Spring Boot favors convention over configuration and is designed to get you up and running as quickly as possible.' It features a 'QUICK START' button and a 'Spring with Convention Over Configuration' callout. A blue speech bubble on the right side of the card says 'A Sub-Project (among 100) of SpringFramework'. At the bottom of the page, there's a summary of Spring Boot's benefits: 'Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.' A 'Features' section lists several benefits, and a 'RELEASE' table shows the current versions available.

A Sub-Project (among 100) of SpringFramework

Spring with Convention Over Configuration

Easier
Smaller
Just work

Spring Boot

RELEASE	DOCUMENTATION
2.0.0 SNAPSHOT	Reference API
1.5.0 SNAPSHOT	Reference API
1.4.3 SNAPSHOT	Reference API
1.4.2 CURRENT	Reference API
1.3.9 SNAPSHOT	Reference API

SpringBoot = a Sub-Project of SpringFramework

“What is SpringBoot ? “

=

“What is SpringFramework ?”

+

“What is in SpringBoot & not in SpringFramework ?”

<http://blog.mimacom.com/introduction-to-spring-boot/>



well the main concept of Spring Boot:



I think it is an excellent analogy about what Spring and Spring Boot are. With Spring framework you have a lot of great ingredients to make a yummy cake (or Spring application), and with Spring Boot you have a cook that will look the



springframework



All Images Videos News Shopping More ▾ Search tools

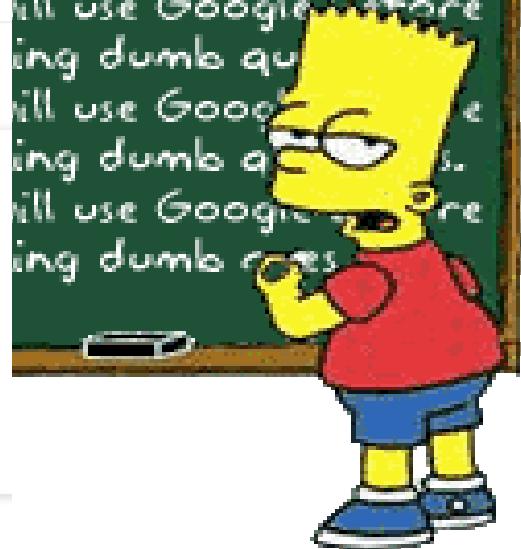
About 2,550,000 results (0.40 seconds)

The **Spring Framework** is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

Spring Framework - Wikipedia

https://en.wikipedia.org/wiki/Spring_Framework

[About this result](#) • [Feedback](#)



Spring Framework - Projects

<https://projects.spring.io/spring-framework/> ▾

A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

Spring

<https://spring.io/> ▾

Spring helps development teams everywhere build simple, portable, fast and flexible JVM-based systems and applications.

Spring Framework Reference Documentation

docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/ ▾

I. Overview of **Spring Framework**. 1. Getting Started with Spring; 2. Introduction to the **Spring Framework**. 2.1. Dependency Injection and Inversion of Control; 2.2.

“SpringFramework” Wikipedia



WIKIPEDIA
The Free Encyclopedia

Article Talk

Read Edit View history

Not logged in Talk Contributions Create account Log in

Search Wikipedia



Spring Framework

From Wikipedia, the free encyclopedia

The **Spring Framework** is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is open source.

Contents [hide]

- 1 Version history
- 2 Modules
 - 2.1 Inversion of control container (dependency injection)
 - 2.2 Aspect-oriented programming framework
 - 2.3 Data access framework
 - 2.4 Transaction management framework
 - 2.5 Model-view-controller framework
 - 2.6 Remote access framework
 - 2.7 Convention-over-configuration rapid application development
 - 2.7.1 Spring Boot
 - 2.7.2 Spring Roo
 - 2.8 Batch framework
 - 2.9 Integration framework
- 3 Criticisms
- 4 See also

Spring Framework



Pivotal Software

Developer(s)	Pivotal Software
Initial release	1 October 2002; 14 years ago
Stable release	4.3.2 [1] / June 10, 2016
Preview release	5.0.0 M2 / September 21, 2016
Repository	github.com/spring-projects/spring-framework [2]
Development status	Active
Written in	Java
Operating system	Cross-platform
Platform	Java Virtual Machine
Type	Application framework
License	Apache License 2.0
Website	spring.io [3]



WIKIPEDIA
The Free Encyclopedia

Wikipedia Extract

The Spring Framework is an **application framework** and **inversion of control container** for the Java platform.

can be used by any Java application ..
.. extensions for web applications.

... popular in the Java community .. alternative EJB.

open source

History & Authors

- framework for J2EE-* libs in <xml>...

- Developped by Rod Johnson & Juergen Hoeller
 - in ~ 2003



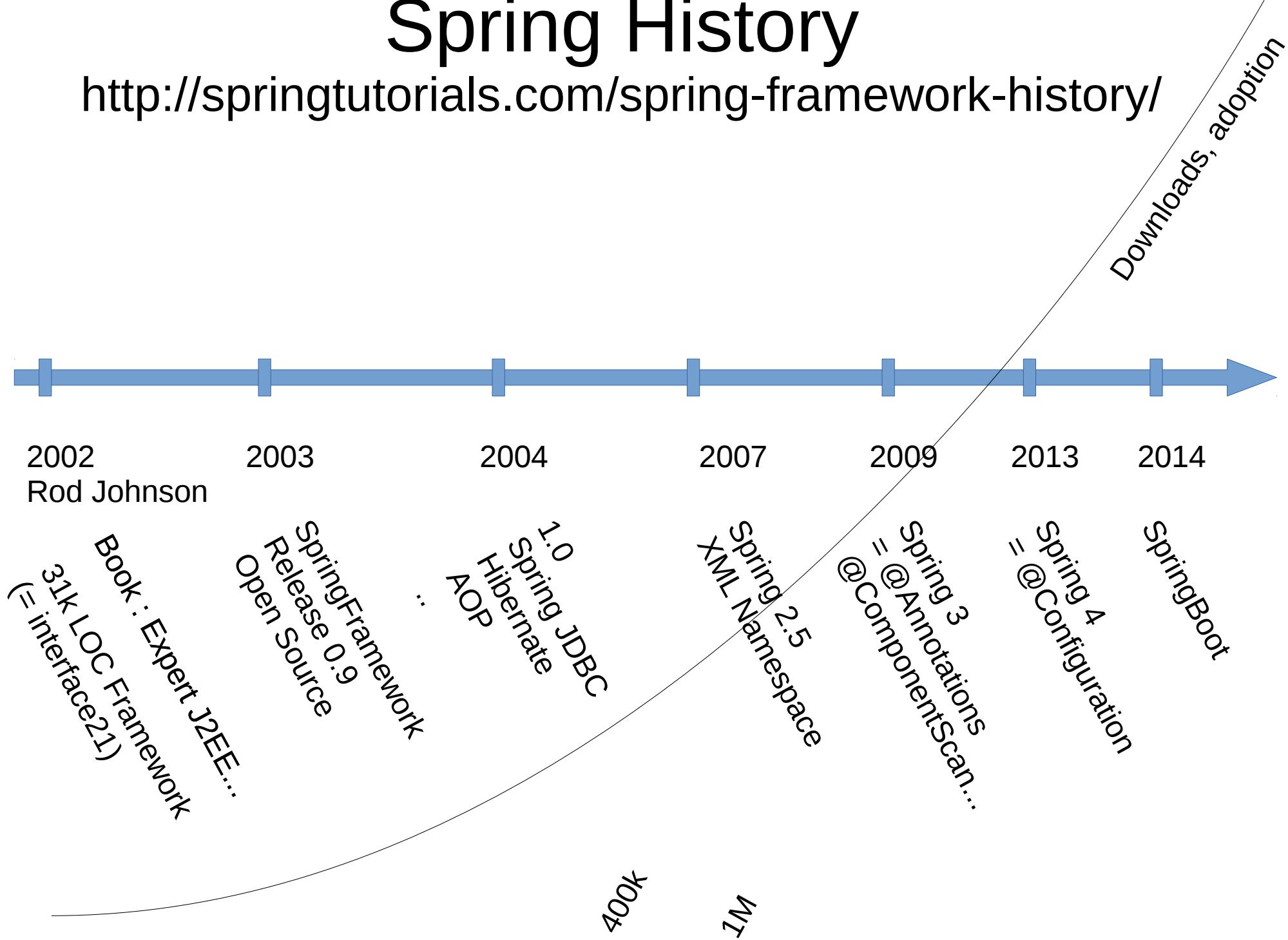
- SpringBoot is @Automagic without </xml>

- From Dave Sayer & Phil Webb
 - In ~ 2014



Spring History

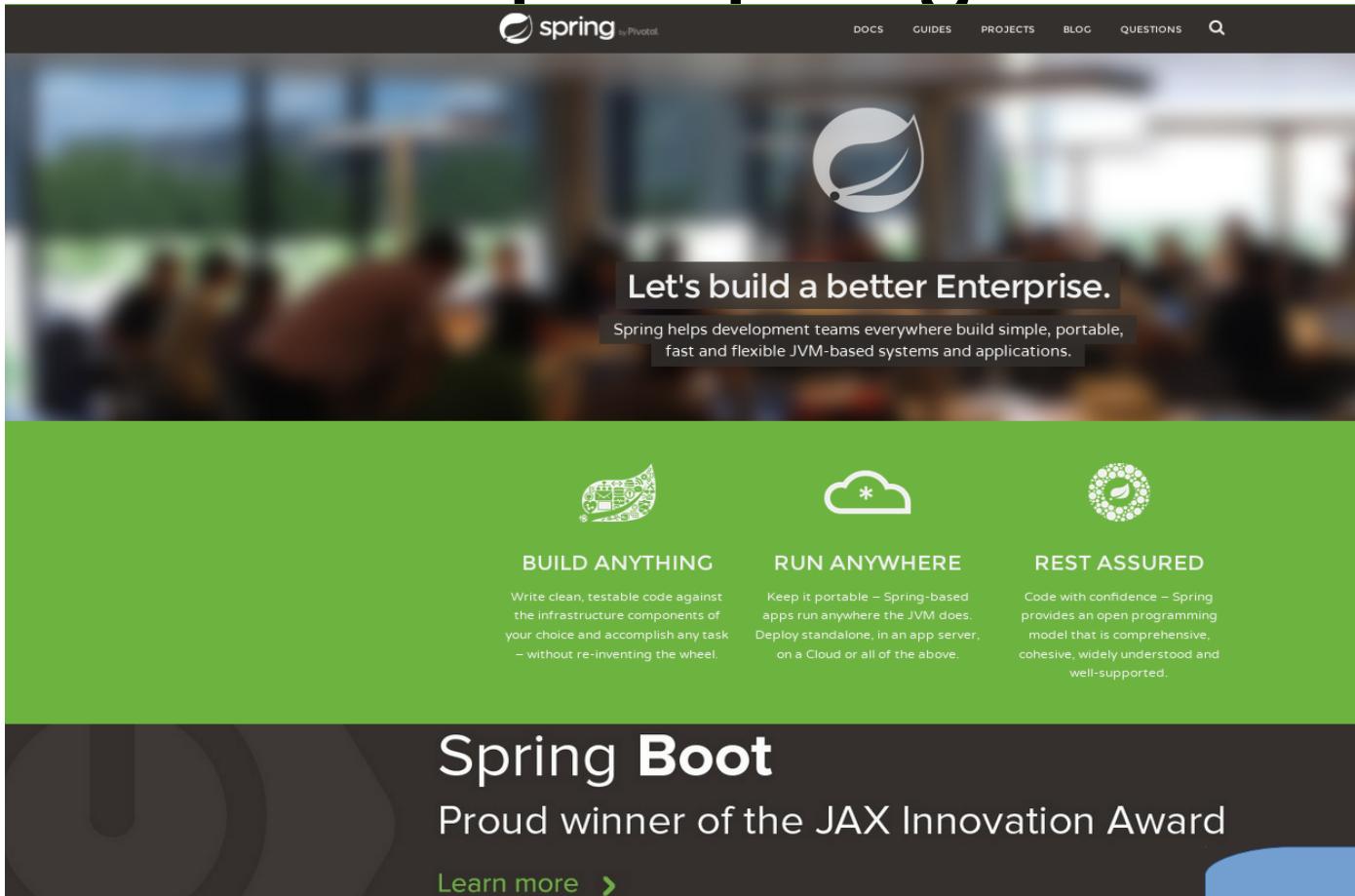
<http://springtutorials.com/spring-framework-history/>



Game of the Name (French Translation)

- Spring = “ressort”, “printemps”, “renouveau”
 - After the cold winter of ugly EJB specs 1.0, 2.0, ...
- Framework = “Cadre de Travail”
 - = Way of working, proposed / imposed by library
- Boot = “démarrage”
 - = your app is a 1 line main() with @magic

http://spring.io



The screenshot shows the official website for Spring Framework. At the top, there's a navigation bar with links for DOCS, GUIDES, PROJECTS, BLOG, and QUESTIONS, along with a search icon. Below the navigation is a large banner featuring a blurred background image of people at an event. In the center of the banner is a stylized leaf logo. Below the logo, the text "Let's build a better Enterprise." is displayed, followed by a subtitle: "Spring helps development teams everywhere build simple, portable, fast and flexible JVM-based systems and applications." The main content area has a green header with three sections: "BUILD ANYTHING" (with a gear icon), "RUN ANYWHERE" (with a cloud icon), and "REST ASSURED" (with a circular icon). Each section contains a brief description of the framework's capabilities. Below this is a dark banner for "Spring Boot", which is described as a "Proud winner of the JAX Innovation Award". A "Learn more" button is located below the banner. At the bottom, there's a "Spring Initializr" section with a "Generate now!" button, and a "Guides" section with a "Browse the Guides" button. A blue speech bubble on the left points to the "Guides" section, and another blue speech bubble on the right points to the "Initializr" section.

spring by Pivotal

DOCS GUIDES PROJECTS BLOG QUESTIONS

Let's build a better Enterprise.

Spring helps development teams everywhere build simple, portable, fast and flexible JVM-based systems and applications.

BUILD ANYTHING

Write clean, testable code against the infrastructure components of your choice and accomplish any task – without re-inventing the wheel.

RUN ANYWHERE

Keep it portable – Spring-based apps run anywhere the JVM does. Deploy standalone, in an app server, on a Cloud or all of the above.

REST ASSURED

Code with confidence – Spring provides an open programming model that is comprehensive, cohesive, widely understood and well-supported.

Spring Boot

Proud winner of the JAX Innovation Award

Learn more >

Spring Initializr

Bootstrap your Spring Boot application with [start.spring.io](#).

Generate now! >

Guides

Whether you're an expert or a newcomer our task-focused Getting Started Guides and Tutorials are designed to get you productive with Spring as quickly as possible.

Browse the Guides >

Amazing
Tutorials & Doc

Amazing Starter

<https://spring.io/guides>

[DOCS](#)[GUIDES](#)[PROJECTS](#)[BLOG](#)[QUESTIONS](#)

Guides

Whatever you're building, these guides are designed to get you productive as quickly as possible – using the latest Spring project releases and techniques as recommended by the [Spring team](#).

Have a suggestion for a new guide? Let us know at [@springcentral](#).



Getting Started Guides

Designed to be completed in 15-30 minutes, these guides provide quick, hands-on instructions for building the "Hello World" of any development task with Spring. In most cases, the only prerequisites are a JDK and a text editor.

[Building a RESTful Web Service](#)

Learn how to create a RESTful web service

[Scheduling Tasks](#)

Learn how to schedule tasks with Spring.

[Consuming a RESTful Web Service](#)

Learn how to retrieve web page data with Spring's RestTemplate.

[Building Java Projects with Gradle](#)

Learn how to build a Java project with Gradle.

[Building Java Projects with Maven](#)

Learn how to build a Java project with Maven.

[Accessing Relational Data using JDBC with Spring](#)

Learn how to access relational data with Spring.

[Uploading Files](#)

Learn how to build a Spring application that accepts multi-part file uploads.

[Authenticating a User with LDAP](#)

Learn how to secure an application with LDAP.

[Registering an Application with Facebook](#)

Learn how to register an application to integrate with Facebook.

[Messaging with Redis](#)

[Registering an Application with](#)

[Messaging with RabbitMQ](#)

http://start.spring.io/

SPRING INITIALIZR bootstrap your application now

Generate a with Spring Boot

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Generate Project alt + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

1/ Download + 2/ Mvn + 3/ Eclipse

```
$ mvn clean compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-clean-plugin:2.6.1:clean (default-clean) @ demo ---
[INFO] Deleting /mnt/a_1tera2/homeData/arnaud/perso/devPerso/my-github/test-snippet/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ demo ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /mnt/a_1tera2/homeData/arnaud/perso/devPerso/my-github/test-springboot/target/classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.380 s
[INFO] Finished at: 2016-11-15T07:57:46+01:00
[INFO] Final Memory: 18M/82M
[INFO] -----
```

\$

1/ Download + 2/ Mvn + 3/ Eclipse

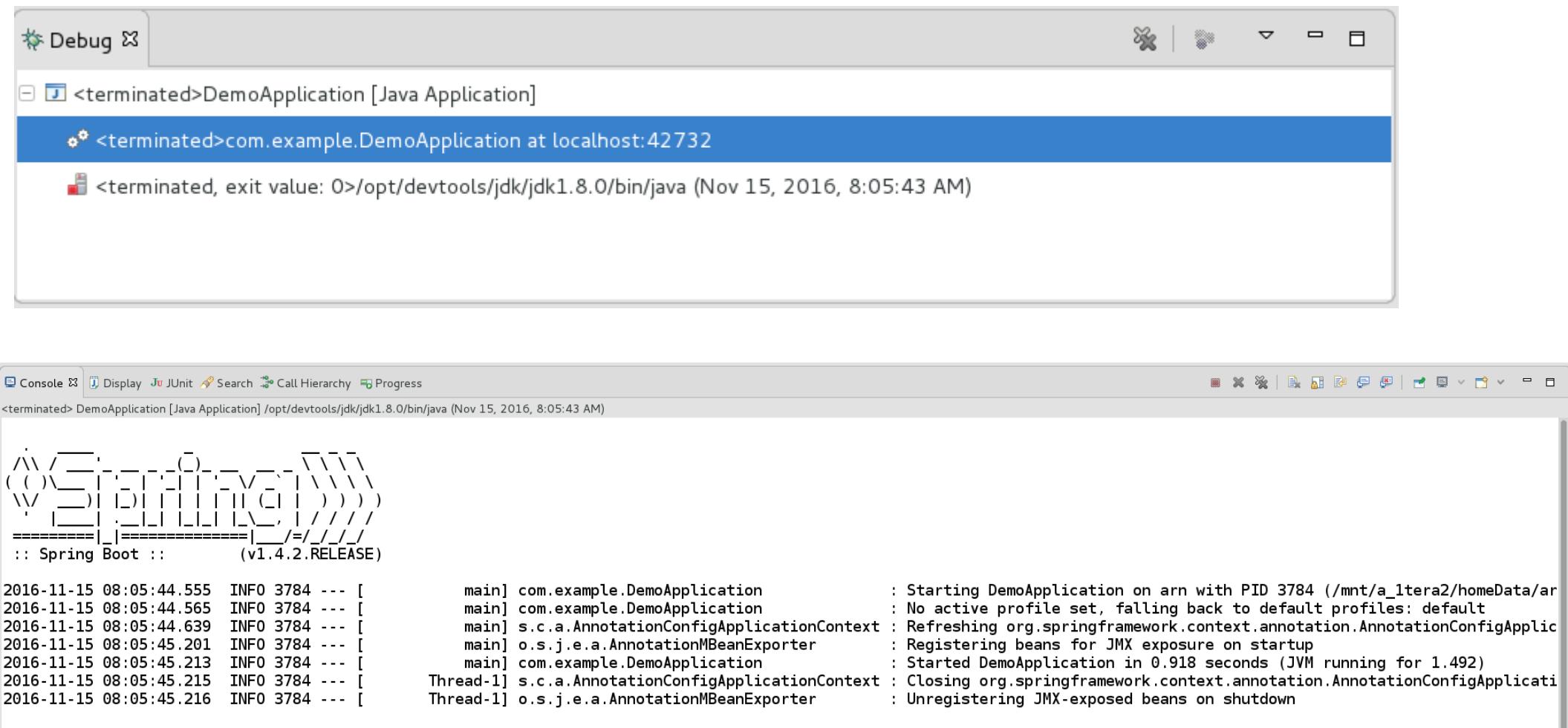
The screenshot shows the Eclipse IDE interface with the following details:

- Left Sidebar (Package Explorer):** Shows the project structure under "demo".
 - src/main/java:
 - com.example:
 - DemoApplication.java
 - src/main/resources
 - src/test/java
 - com.example
 - DemoApplicationTests.java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml

Right Side (Editor): Displays the content of `DemoApplication.java`.

```
package com.example;  
  
import org.springframework.boot.SpringApplication;  
  
@SpringBootApplication  
public class DemoApplication {  
  
    public static void main(String[] args) {  
        SpringApplication.run(DemoApplication.class, args);  
    }  
}
```

4/ Run IT ...



5/ Junit Test It !!

The screenshot shows a Java development environment with two main panes. On the left is the 'Package Explorer' pane, which displays the project structure. The 'demo' project contains several source folders: 'src/main/java' (containing 'com.example' package with 'DemoApplication.java'), 'src/main/resources', and 'src/test/java' (containing 'com.example' package with 'DemoApplicationTests.java'). Other visible files include 'JRE System Library [JavaSE-1.8]', 'Maven Dependencies', 'src', 'target', 'mvnw', 'mvnw.cmd', and 'pom.xml'. The 'DemoApplicationTests.java' file is currently selected in the package explorer and is shown in the code editor on the right.

```
package com.example;

import org.junit.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

@SpringBootTest
public class DemoApplicationTests {

    @Test
    public void contextLoads() {
    }
}
```

5/ Junit OK

The screenshot shows the Eclipse IDE interface during a JUnit test execution. The top bar displays tabs for 'Debug' (selected), 'Breakpoints', 'Expressions', and 'Registers'. The left side features a 'Debug' view with a list of terminated processes, including 'DemoApplicationTests [JUnit]' which is currently selected. Below this is the code editor with 'DemoApplication.java' and 'DemoApplicationTests.java' tabs. The code in 'DemoApplicationTests.java' is as follows:

```
package com.example;

import org.junit.Test;

@RunWith(SpringRunner.class)
@SpringBootTest
public class DemoApplicationTests {

    @Test
    public void contextLoads() {
    }
}
```

The right side of the interface shows the 'Variables' view, which is currently empty. At the bottom, the 'Console' tab is active, showing the message 'Finished after 0.823 seconds'. Below the console are summary statistics: 'Runs: 1/1', 'Errors: 0', 'Failures: 0', and a green progress bar indicating 100% completion. The bottom navigation bar includes tabs for 'Console', 'Display', 'JUnit' (selected), 'Search', 'Call Hierarchy', and 'Progress'. A status bar at the very bottom shows the test result: 'com.example.DemoApplicationTests [Runner: JUnit 4] (0.008 s)' and 'contextLoads (0.008 s)'.

6/ Easy packaging to launch main ... java -jar

Small all-in-one target/jar

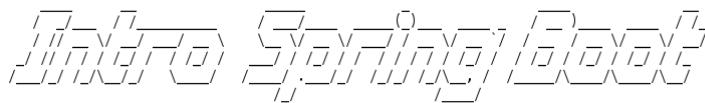
6.3 M “only”

```
$ ls -lh target
total 6.3M
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:29 classes
-rw-r--r-- 1 arnaud arnaud 6.3M Nov 16 21:29 demo-0.0.1-SNAPSHOT.jar
-rw-r--r-- 1 arnaud arnaud 2.7K Nov 16 21:29 demo-0.0.1-SNAPSHOT.jar.original
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:29 generated-sources
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:29 generated-test-sources
drwxr-xr-x 2 arnaud arnaud 4.0K Nov 16 21:29 maven-archiver
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:29 maven-status
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:29 test-classes
$ 
$ jar tf target/demo-0.0.1-SNAPSHOT.jar
META-INF/
META-INF/MANIFEST.MF
BOOT-INF/
BOOT-INF/classes/
BOOT-INF/classes/com/
BOOT-INF/classes/com/example/
BOOT-INF/classes/application.properties
BOOT-INF/classes/com/example/DemoApplication.class
META-INF/maven/
META-INF/maven/com.example/
META-INF/maven/com.example/demo/
META-INF/maven/com.example/demo/pom.xml
META-INF/maven/com.example/demo/pom.properties
BOOT-INF/lib/
BOOT-INF/lib/spring-beans-4.3.4.RELEASE.jar
BOOT-INF/lib/spring-boot-starter-logging-1.4.2.RELEASE.jar
BOOT-INF/lib/slf4j-api-1.7.21.jar
BOOT-INF/lib/jul-to-slf4j-1.7.21.jar
BOOT-INF/lib/spring-context-4.3.4.RELEASE.jar
BOOT-INF/lib/spring-core-4.3.4.RELEASE.jar
BOOT-INF/lib/spring-boot-starter-1.4.2.RELEASE.jar
BOOT-INF/lib/spring-aop-4.3.4.RELEASE.jar
BOOT-INF/lib/spring-boot-autoconfigure-1.4.2.RELEASE.jar
BOOT-INF/lib/logback-classic-1.1.7.jar
BOOT-INF/lib/spring-boot-1.4.2.RELEASE.jar
BOOT-INF/lib/spring-expression-4.3.4.RELEASE.jar
```

“Hello World” ok
But what is Really SpringBoot?

What's Next ?

An “Ascii Art Banner” Printer ?



Font Name: Slant

[Use Font](#) | [Select & Copy](#)

<http://patorjk.com/software/taag/>

src/main/resources/
banner.txt

The screenshot shows an IDE interface. On the left is a "Package Explorer" view showing a project structure with "Other Projects" containing a "demo" folder. Inside "demo" are "src/main/java" (containing "DemoApplication.java"), "src/main/resources" (containing "application.properties"), and "src/test/java". A blue callout bubble points to "src/main/resources/banner.txt". On the right is a "banner.txt" file viewer showing the ASCII art banner. Below it is a terminal window titled "terminated> DemoApplication [Java Application] /opt/devtools/jdk/jdk1.8.0/bin/java (Nov 16, 2016, 8:46:49 PM)" displaying the Java application logs. The logs show the application starting up, registering beans, and shutting down.

```
2016-11-16 20:46:50.362 INFO 3401 ... [main] com.example.DemoApplication : Starting DemoApplication on arn with PID 3401
2016-11-16 20:46:50.370 INFO 3401 ... [main] com.example.DemoApplication : No active profile set, falling back to default
2016-11-16 20:46:50.414 INFO 3401 ... [main] s.c.a.AnnotationConfigApplicationContext : Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext
2016-11-16 20:46:50.971 INFO 3401 ... [main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2016-11-16 20:46:50.981 INFO 3401 ... [main] com.example.DemoApplication : Started DemoApplication in 0.834 seconds (JVM Thread)
2016-11-16 20:46:50.985 INFO 3401 ... [Thread-1] s.c.a.AnnotationConfigApplicationContext : Closing org.springframework.context.annotation.AnnotationConfigApplicationContext
2016-11-16 20:46:50.987 INFO 3401 ... [Thread-1] o.s.j.e.a.AnnotationMBeanExporter : Unregistering JMX-exposed beans on shutdown
```

“Hello” that contains the “World” jars

The screenshot shows an IDE interface with several windows open:

- Package Explorer:** Shows the project structure under "demo".
- DemoApplication.java:** The main Java file.
- pom.xml:** The Maven configuration file.
- Dependency Hierarchy [test]:** A central window displaying the dependency tree. It shows the following structure:
 - spring-boot-starter: 1.4.2.RELEASE [compile]
 - spring-boot: 1.4.2.RELEASE [compile]
 - spring-core: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-context: 4.3.4.RELEASE [compile]
 - spring-aop: 4.3.4.RELEASE [compile]
 - spring-beans: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-core: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-beans: 4.3.4.RELEASE [compile]
 - spring-core: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-core: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-expression: 4.3.4.RELEASE [compile]
 - spring-core: 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-boot-autoconfigure: 1.4.2.RELEASE [compile]
 - spring-boot: 1.4.2.RELEASE (omitted for conflict with 1.4.2.RELEASE) [compile]
 - spring-boot-starter-logging: 1.4.2.RELEASE [compile]
 - logback-classic: 1.1.7 [compile]
 - logback-core: 1.1.7 [compile]
 - slf4j-api: 1.7.21 (managed from 1.7.20) (omitted for conflict with 1.7.21) [compile]
 - jcl-over-slf4j: 1.7.21 [compile]
 - slf4j-api: 1.7.21 (omitted for conflict with 1.7.21) [compile]
 - jul-to-slf4j: 1.7.21 [compile]
 - slf4j-api: 1.7.21 (omitted for conflict with 1.7.21) [compile]
 - log4j-over-slf4j: 1.7.21 [compile]
- Resolved Dependencies:** A list of resolved dependencies and their versions.

At the bottom, tabs indicate the current view: Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml. A status bar at the bottom left says "Infinitest is waiting for changes" and "32 items selected".

mvn dependency:tree

Same as in eclipse, using command line

```
$ mvn dependency:tree
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-dependency-plugin:2.10:tree (default-cli) @ demo ---
[INFO] com.example:demo:jar:0.0.1-SNAPSHOT
[INFO] +- org.springframework.boot:spring-boot-starter:jar:1.4.2.RELEASE:compile
[INFO] |  +- org.springframework.boot:spring-boot:jar:1.4.2.RELEASE:compile
[INFO] |  |  \- org.springframework:spring-context:jar:4.3.4.RELEASE:compile
[INFO] |  |  +- org.springframework:spring-aop:jar:4.3.4.RELEASE:compile
[INFO] |  |  +- org.springframework:spring-beans:jar:4.3.4.RELEASE:compile
[INFO] |  |  \- org.springframework:spring-expression:jar:4.3.4.RELEASE:compile
[INFO] |  +- org.springframework.boot:spring-boot-autoconfigure:jar:1.4.2.RELEASE:compile
[INFO] |  +- org.springframework.boot:spring-boot-starter-logging:jar:1.4.2.RELEASE:compile
[INFO] |  |  +- ch.qos.logback:logback-classic:jar:1.1.7:compile
[INFO] |  |  |  \- ch.qos.logback:logback-core:jar:1.1.7:compile
[INFO] |  |  +- org.slf4j:jcl-over-slf4j:jar:1.7.21:compile
[INFO] |  |  +- org.slf4j:jul-to-slf4j:jar:1.7.21:compile
[INFO] |  |  \- org.slf4j:log4j-over-slf4j:jar:1.7.21:compile
[INFO] |  +- org.springframework:spring-core:jar:4.3.4.RELEASE:compile
[INFO] |  \- org.yaml:snakeyaml:jar:1.17:runtime
[INFO] \- org.springframework.boot:spring-boot-starter-test:jar:1.4.2.RELEASE:test
[INFO]   +- org.springframework.boot:spring-boot-test:jar:1.4.2.RELEASE:test
[INFO]   +- org.springframework.boot:spring-boot-test-autoconfigure:jar:1.4.2.RELEASE:test
[INFO]   +- com.jayway.jsonpath:json-path:jar:2.2.0:test
[INFO]   |  +- net.minidev:json-smart:jar:2.2.1:test
[INFO]   |  |  \- net.minidev:accessors-smart:jar:1.1:test
[INFO]   |  |  \- org.ow2.asm:asm:jar:5.0.3:test
[INFO]   |  \- org.slf4j:slf4j-api:jar:1.7.21:compile
[INFO]   +- junit:junit:jar:4.12:test
[INFO]   +- org.assertj:assertj-core:jar:2.5.0:test
[INFO]   +- org.mockito:mockito-core:jar:1.10.19:test
[INFO]   |  \- org.objenesis:objenesis:jar:2.1:test
[INFO]   +- org.hamcrest:hamcrest-core:jar:1.3:test
[INFO]   +- org.hamcrest:hamcrest-library:jar:1.3:test
[INFO]   +- org.skyscreamer:jsonassert:jar:1.3.0:test
[INFO]   |  \- org.json:json:jar:20140107:test
[INFO]   \- org.springframework:spring-test:jar:4.3.4.RELEASE:test
[INFO] -----
[INFO] BUILD SUCCESS
```

Dependencies ...

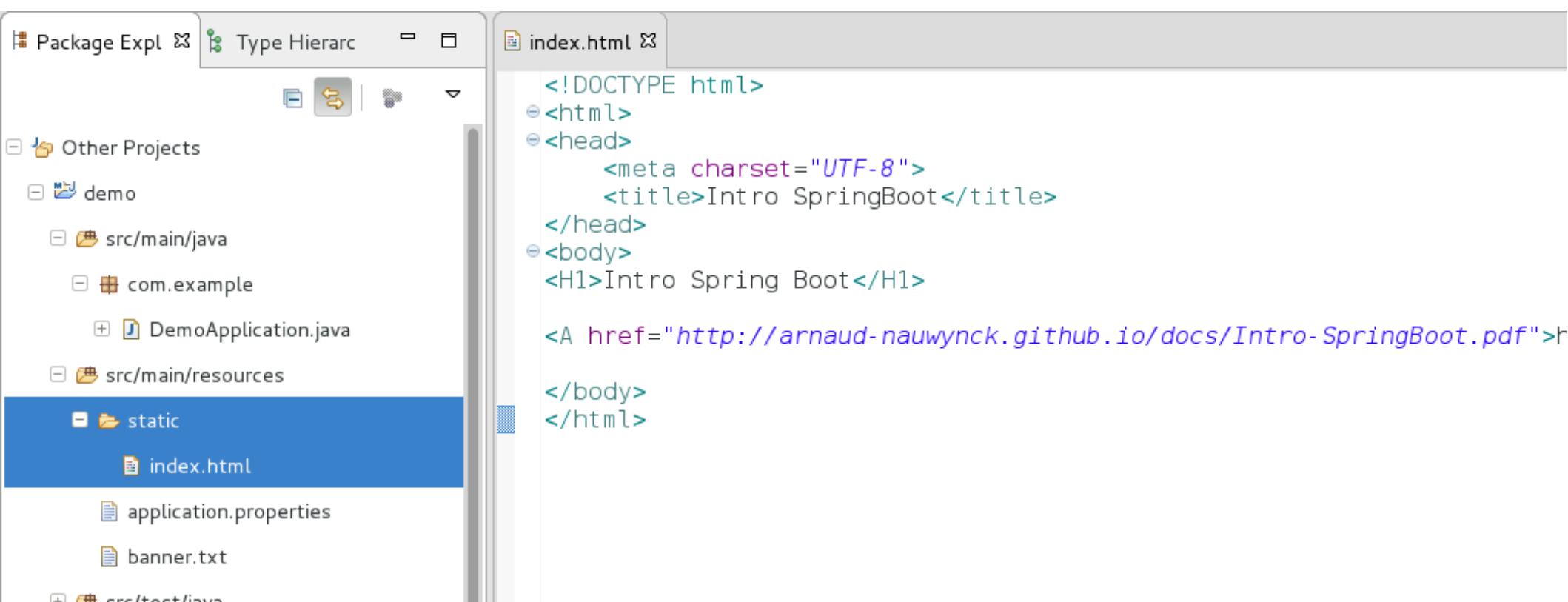
(excluding <scope>test</scope>)

```
$ mvn dependency:list | grep -v :test
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-dependency-plugin:2.10:list (default-cli) @ demo ---
[INFO]
[INFO] The following files have been resolved:
[INFO]   org.slf4j:jul-to-slf4j:jar:1.7.21:compile
[INFO]   org.springframework:spring-aop:jar:4.3.4.RELEASE:compile
[INFO]   org.yaml:snakeyaml:jar:1.17:runtime
[INFO]   org.springframework:spring-beans:jar:4.3.4.RELEASE:compile
[INFO]   org.springframework.boot:spring-boot-starter-logging:jar:1.4.2.RELEASE:compile
[INFO]   org.slf4j:slf4j-api:jar:1.7.21:compile
[INFO]   org.springframework:spring-context:jar:4.3.4.RELEASE:compile
[INFO]   org.springframework:spring-core:jar:4.3.4.RELEASE:compile
[INFO]   org.springframework.boot:spring-boot-starter:jar:1.4.2.RELEASE:compile
[INFO]   org.springframework.boot:spring-boot-autoconfigure:jar:1.4.2.RELEASE:compile
[INFO]   ch.qos.logback:logback-classic:jar:1.1.7:compile
[INFO]   org.springframework.boot:spring-boot:jar:1.4.2.RELEASE:compile
[INFO]   org.springframework:spring-expression:jar:4.3.4.RELEASE:compile
[INFO]   ch.qos.logback:logback-core:jar:1.1.7:compile
[INFO]   org.slf4j:jcl-over-slf4j:jar:1.7.21:compile
[INFO]   org.slf4j:log4j-over-slf4j:jar:1.7.21:compile
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
```

Almost a WebApp (add 4 lines)...

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Add Web page resources/static/index.html



Restart... your Web Application running in 2 seconds

The screenshot shows a Java development environment with the following interface elements:

- Debug View:** Shows the application is running under the name "DemoApplication" at port 51469. It lists several threads: "Daemon Thread [ContainerBackgroundProcessor[StandardEngine[Tomcat]]] (Running)", "Thread [container-0] (Running)", and "Daemon Thread [NioBlockingSelector.BlockPoller-1] (Running)".
- Code Editor:** Displays the source code for `DemoApplication.java`. The code defines a `main` method that runs the `SpringApplication`.
- Variables View:** Empty.
- Console View:** Shows the application's startup logs. Key messages include:
 - "Tomcat initialized with port(s): 8080 (http)"
 - "Starting service Tomcat"
 - "Starting Servlet Engine: Apache Tomcat/8.5.6"
 - "Initializing Spring embedded WebApplicationContext"
 - "Root WebApplicationContext: initialization completed in 1201 ms"
 - "Mapping servlet: 'dispatcherServlet' to [/*]"
 - "Mapping filter: 'characterEncodingFilter' to: [/*]"
 - "Mapping filter: 'hiddenHttpMethodFilter' to: [/*]"
 - "Mapping filter: 'httpPutFormContentFilter' to: [/*]"
 - "Mapping filter: 'requestContextFilter' to: [/*]"
 - "Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@51e0c93a: URL [null]"
 - "Mapped "[/*],produces=[text/html]" onto public org.springframework.web.ModelAndView"
 - "Mapped "[/*]" onto public org.springframework.http.ResponseEntity<java.util.Map>[java.lang.String]"
 - "Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]"
 - "Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]"
 - "Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]"
 - "Adding welcome page: class path resource [static/index.html]"
 - "Registering beans for JMX exposure on startup"
 - "Tomcat started on port(s): 8080 (http)"
 - "Started DemoApplication in 2.454 seconds (JVM running for 2.793)"

http://localhost:8080/



Intro Spring Boot

<http://arnaud-nauwynck.github.io/docs/Intro-SpringBoot.pdf>

A screenshot of the Chrome Developer Tools Network tab. The tab title is "Developer Tools - http://localhost:8080". The network timeline shows a single request to "localhost" taking 50 ms. The request details show the following information:

- Name: localhost
- Request URL: http://localhost:8080/
- Request Method: GET
- Status Code: 304
- Remote Address: [::1]:8080
- Response Headers:
 - Date: Wed, 16 Nov 2016 20:39:38 GMT
 - Last-Modified: Wed, 16 Nov 2016 20:34:49 GMT
- Request Headers:
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
 - Accept-Encoding: gzip, deflate, sdch
 - Accept-Language: en-US,en;q=0.8,fr;q=0.6
 - Cache-Control: max-age=0
 - Connection: keep-alive
 - Cookie: remember-me=K1FMSTBXeVpEcUtPTXRDaDZMN0RBZz090mQyMlhINlhXav
 - Host: localhost:8080
 - If-Modified-Since: Wed, 16 Nov 2016 20:34:49 GMT
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)

```
$ curl -v http://localhost:8080/index.html
* Hostname was NOT found in DNS cache
*   Trying ::1...
* Connected to localhost (::1) port 8080 (#0)
> GET /index.html HTTP/1.1
> User-Agent: curl/7.38.0
> Host: localhost:8080
> Accept: */*
>
< HTTP/1.1 200
< Last-Modified: Wed, 16 Nov 2016 20:34:49 GMT
< Accept-Ranges: bytes
< Content-Type: text/html
< Content-Length: 279
< Date: Wed, 16 Nov 2016 20:37:25 GMT
<
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Intro SpringBoot</title>
</head>
<body>
<H1>Intro Spring Boot</H1>

<A href="http://arnaud-nauwynck.github.io/docs/Intro-SpringBoot.pdf">
</body>
* Connection #0 to host localhost left intact
</html>$
```

AutoReload & LiveReload

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
</dependency>
```

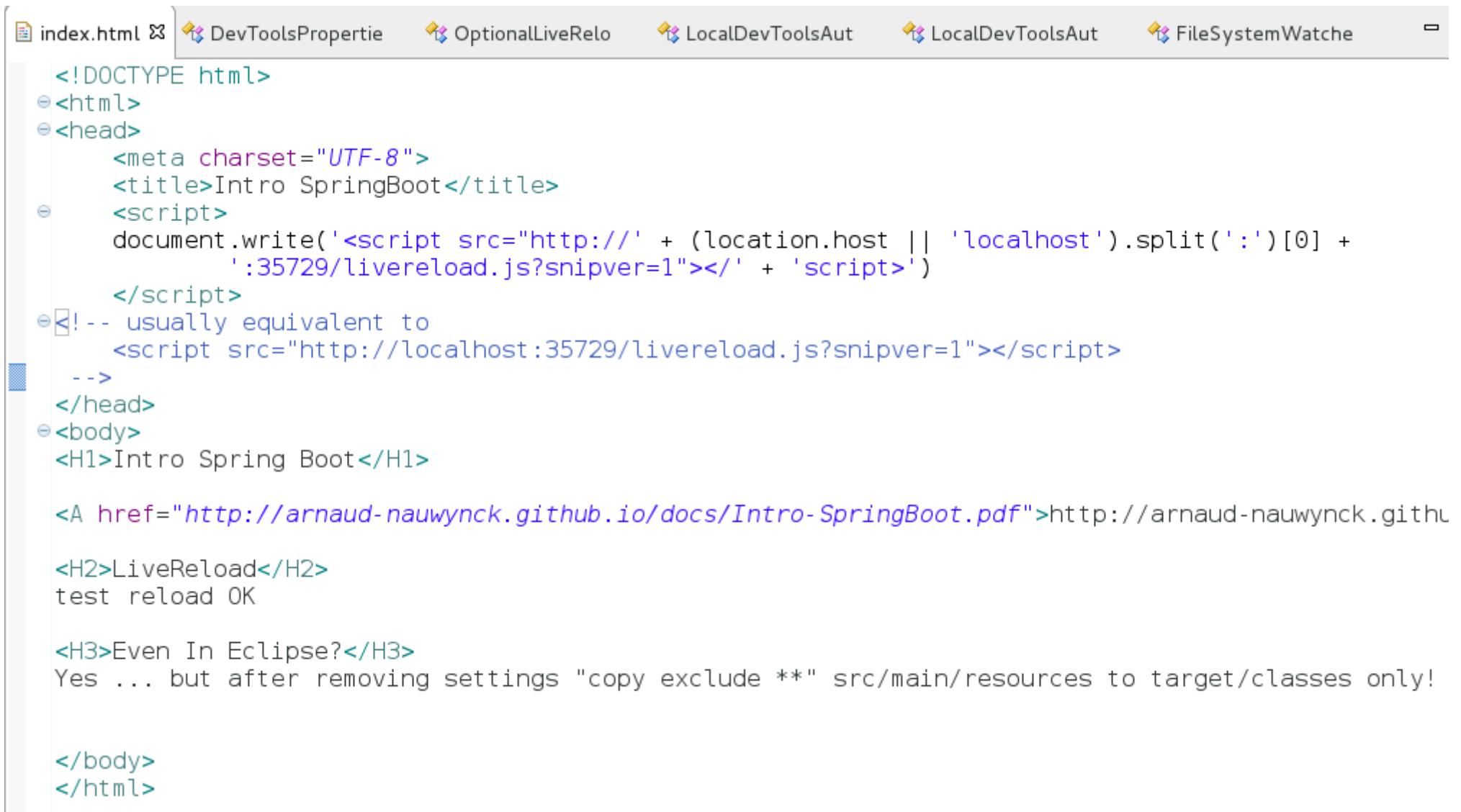
- AutoReload :
change your spring app
=> server auto restart (partial)

For fast developer experience
like “gulp serve”

- LiveReload:
change html/css/...
=> client Web Browser auto refresh!!

```
2016-11-16 23:25:59.119 INFO 7916 --- [ restartedMain] oConfiguration$WelcomePageHandlerMapping : Adding welcome page: class path resource [static/]
2016-11-16 23:25:59.222 INFO 7916 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2016-11-16 23:25:59.324 INFO 7916 --- [ restartedMain] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2016-11-16 23:25:59.383 INFO 7916 --- [ restartedMain] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
2016-11-16 23:25:59.391 INFO 7916 --- [ restartedMain] com.example.DemoApplication : Started DemoApplication in 2.665 seconds (JVM runn
```

<script>... :35729/ livereload.js</script>



The screenshot shows a browser developer tools window with the Network tab selected. A single entry is listed: "livereload.js" from "http://localhost:35729". The file size is 0 bytes and it was received in 0ms. The status is "OK".

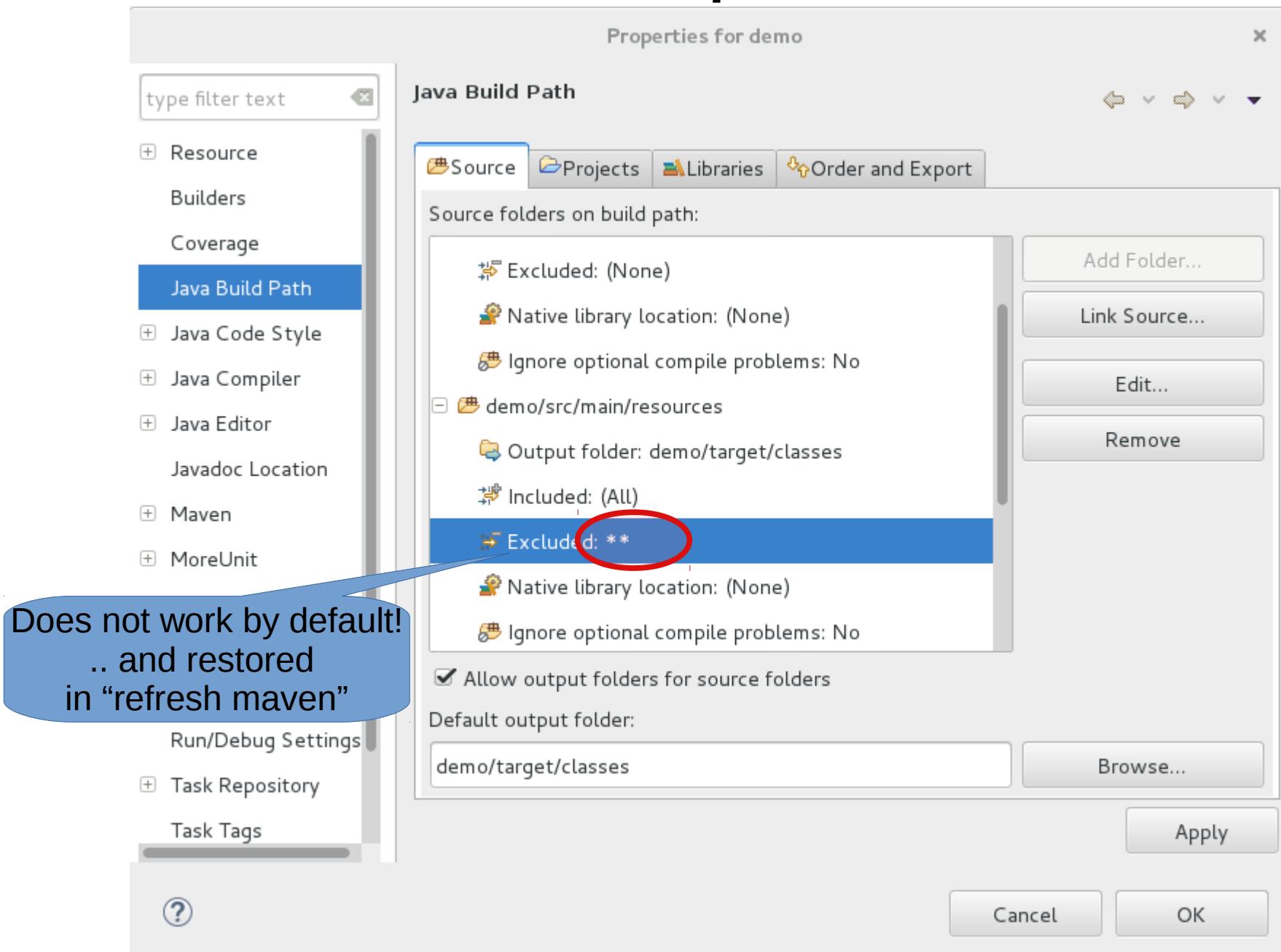
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Intro SpringBoot</title>
    <script>
        document.write('<script src="http://' + (location.host || 'localhost').split(':')[0] +
                      ':35729/livereload.js?snipver=1"></' + 'script>')
    </script>
    <!-- usually equivalent to
        <script src="http://localhost:35729/livereload.js?snipver=1"></script>
    -->
</head>
<body>
    <H1>Intro Spring Boot</H1>

    <A href="http://arnaud-nauwynck.github.io/docs/Intro-SpringBoot.pdf">http://arnaud-nauwynck.githu...
    <H2>LiveReload</H2>
    test reload OK

    <H3>Even In Eclipse?</H3>
    Yes ... but after removing settings "copy exclude **" src/main/resources to target/classes only!

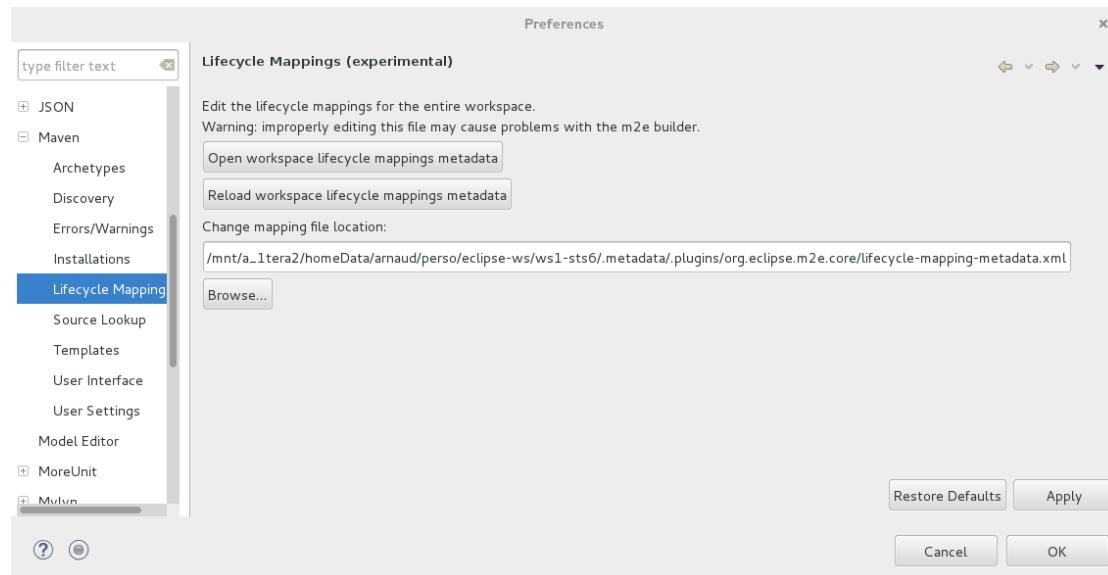
</body>
</html>
```

LiveReload in Eclipse ClassPath ...



Eclipse settings ?!@Gr#&!....

[http://stackoverflow.com/questions/7754255/
Maven-m2eclipse-excludes-my-resources-all-the-time](http://stackoverflow.com/questions/7754255/Maven-m2eclipse-excludes-my-resources-all-the-time)



```
<pluginExecution>
  <pluginExecutionFilter>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-resources-plugin</artifactId>
    <versionRange>2.6</versionRange>
    <goals>
      <goal>testResources</goal>
      <goal>resources</goal>
    </goals>
  </pluginExecutionFilter>
  <action>
    <execute />
  </action>
</pluginExecution>
```

Change default ???
(was <ignore />)

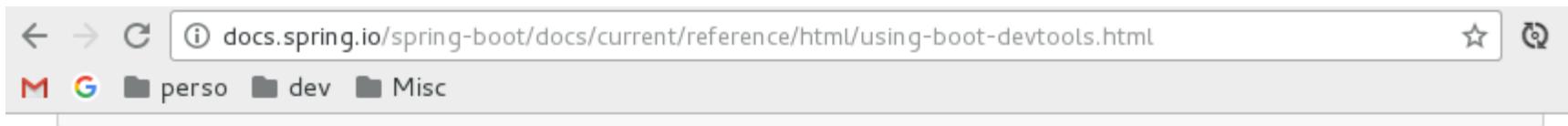
Does not work either..

Real Web Development ?

use GULP
to check + transpile “TypeScript” to Javascript
To transpile SASS to CSS
To run Unit tests

For fast developer experience
use **gulp** (npm,bower,sass,ts) ...

... then you “could” livereload sprinboot ...using a “trigger touch file” ??



20.2.4 Using a trigger file

If you work with an IDE that continuously compiles changed files, you might prefer to trigger restarts only at specific times. To do this you can use a “trigger file”, which is a special file that must be modified when you want to actually trigger a restart check. Changing the file only triggers the check and the restart will only occur if Devtools has detected it has to do something. The trigger file could be updated manually, or via an IDE plugin.

To use a trigger file use the `spring.devtools.restart.trigger-file` prop

Springboot listen to “gulp compile”??
... USELESS !!!
redundant with “gulp serve” 3000



You might want to set `spring.devtools.restart.trigger-file` as a global setting so that all your projects behave in the same way.

Still Deploy with “java -jar”

```
$ ls -lh target/
total 14M
drwxr-xr-x 4 arnaud arnaud 4.0K Nov 16 21:45 classes
-rw-r--r-- 1 arnaud arnaud 14M Nov 16 21:45 demo-0.0.1-SNAPSHOT.jar
-rw-r--r-- 1 arnaud arnaud 3.3K Nov 16 21:45 demo-0.0.1-SNAPSHOT.jar.original
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:45 generated-sources
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:45 generated-test-sources
drwxr-xr-x 2 arnaud arnaud 4.0K Nov 16 21:45 maven-archiver
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:45 maven-status
drwxr-xr-x 3 arnaud arnaud 4.0K Nov 16 21:45 test-classes
$  
$ java -jar target/demo-0.0.1-SNAPSHOT.jar
```

now 14M jar ...

java -jar ... = Web App !



```
2016-11-16 21:46:02.461  INFO 5415 --- [           main] com.example.DemoApplication          : Starting DemoApplication v0.0.1
n with PID 5415 (/mnt/a_1tera2/homeData/arnaud/perso/devPerso/my-github/test-snippets.github/test-springboot/target/demo-0.0.1-SNAP
d by arnaud in /mnt/a_1tera2/homeData/arnaud/perso/devPerso/my-github/test-snippets.github/test-springboot)
```

Dependencies Added for spring-boot-start-web

A screenshot of the IntelliJ IDEA interface showing the dependency tree for a project named "demo". A blue callout bubble on the left points to the "Tomcat embedded .jar" section of the tree.

The dependency tree is organized into several sections:

- Tomcat embedded .jar**: A blue-highlighted section containing:
 - spring-boot-starter-web : 1.4.2.RELEASE [compile]
 - spring-boot-starter : 1.4.2.RELEASE (omitted for conflict with 1.4.2.RELEASE) [compile]
 - spring-boot-starter-tomcat : 1.4.2.RELEASE [compile]
 - tomcat-embed-core : 8.5.6 [compile]
 - tomcat-embed-el : 8.5.6 [compile]
 - tomcat-embed-websocket : 8.5.6 [compile]
 - tomcat-embed-core : 8.5.6 (omitted for conflict with 8.5.6) [compile]
- hibernate-validator**: A section containing:
 - validation-api : 1.1.0.Final [compile]
 - jboss-logging : 3.3.0.Final (managed from 3.2.1.Final) [compile]
 - classmate : 1.3.3 (managed from 1.1.0) [compile]
- jackson-databind**: A section containing:
 - jackson-annotations : 2.8.4 (managed from 2.8.0) [compile]
 - jackson-core : 2.8.4 [compile]
- spring-web**: A section containing:
 - spring-aop : 4.3.4.RELEASE [compile]
 - spring-beans : 4.3.4.RELEASE [compile]
 - spring-context : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-core : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
- spring-webmvc**: A section containing:
 - spring-aop : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-beans : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-context : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
 - spring-core : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
- spring-expression**: A section containing:
 - spring-expression : 4.3.4.RELEASE [compile]
- spring-web**: A section containing:
 - spring-web : 4.3.4.RELEASE (omitted for conflict with 4.3.4.RELEASE) [compile]
- objenesis**: A section containing:
 - objenesis : 2.1 [test]
- slf4j-api**: A section containing:
 - slf4j-api : 1.7.21 [compile]
- snakeyaml**: A section containing:
 - snakeyaml : 1.17 [runtime]
- spring-aop**: A section containing:
 - spring-aop : 4.3.4.RELEASE [compile]
- spring-beans**: A section containing:
 - spring-beans : 4.3.4.RELEASE [compile]
- spring-boot**: A section containing:
 - spring-boot : 1.4.2.RELEASE [compile]
- spring-boot-autoconfigure**: A section containing:
 - spring-boot-autoconfigure : 1.4.2.RELEASE [compile]
- spring-boot-starter**: A section containing:
 - spring-boot-starter : 1.4.2.RELEASE [compile]
- spring-boot-starter-logging**: A section containing:
 - spring-boot-starter-logging : 1.4.2.RELEASE [compile]
- spring-boot-starter-test**: A section containing:
 - spring-boot-starter-test : 1.4.2.RELEASE [test]
- spring-boot-starter-tomcat**: A section containing:
 - spring-boot-starter-tomcat : 1.4.2.RELEASE [compile]
- spring-boot-starter-web**: A section containing:
 - spring-boot-starter-web : 1.4.2.RELEASE [compile]
- spring-boot-test**: A section containing:
 - spring-boot-test : 1.4.2.RELEASE [test]
- spring-boot-test-autoconfigure**: A section containing:
 - spring-boot-test-autoconfigure : 1.4.2.RELEASE [test]
- spring-context**: A section containing:
 - spring-context : 4.3.4.RELEASE [compile]
- spring-core**: A section containing:
 - spring-core : 4.3.4.RELEASE [compile]
- spring-expression**: A section containing:
 - spring-expression : 4.3.4.RELEASE [compile]
- spring-test**: A section containing:
 - spring-test : 4.3.4.RELEASE [test]
- spring-web**: A section containing:
 - spring-web : 4.3.4.RELEASE [compile]
- spring-webmvc**: A section containing:
 - spring-webmvc : 4.3.4.RELEASE [compile]
- tomcat-embed-core**: A section containing:
 - tomcat-embed-core : 8.5.6 [compile]
- tomcat-embed-el**: A section containing:
 - tomcat-embed-el : 8.5.6 [compile]
- tomcat-embed-websocket**: A section containing:
 - tomcat-embed-websocket : 8.5.6 [compile]
- validation-api**: A section containing:
 - validation-api : 1.1.0.Final [compile]

At the bottom of the window, there are tabs for Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml.

JAR > WAR

ClassLoader in springboot can zip nested jar

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
```

```
$ jar tf target/demo-0.0.1-SNAPSHOT.jar | grep '\.jar'
BOOT-INF/lib/spring-web-4.3.4.RELEASE.jar
BOOT-INF/lib/jul-to-slf4j-1.7.21.jar
BOOT-INF/lib/spring-boot-starter-tomcat-1.4.2.RELEASE.jar
BOOT-INF/lib/hibernate-validator-5.2.4.Final.jar
BOOT-INF/lib/tomcat-embed-el-8.5.6.jar
BOOT-INF/lib/spring-aop-4.3.4.RELEASE.jar
BOOT-INF/lib/tomcat-embed-websocket-8.5.6.jar
  NF/lib/snakeyaml-1.17.jar
  NF/lib/tomcat-embed-core-8.5.6.jar
  NF/lib/spring-beans-4.3.4.RELEASE.jar
  .... _NF/lib/spring-boot-starter-logging-1.4.2.RELEASE.jar
  BOOT-INF/lib/slf4j-api-1.7.21.jar
  BOOT-INF/lib/spring-webmvc-4.3.4.RELEASE.jar
  BOOT-INF/lib/spring-boot-starter-web-1.4.2.RELEASE.jar
  BOOT-INF/lib/spring-context-4.3.4.RELEASE.jar
  BOOT-INF/lib/spring-core-4.3.4.RELEASE.jar
  BOOT-INF/lib/jackson-databind-2.8.4.jar
  BOOT-INF/lib/spring-boot-starter-1.4.2.RELEASE.jar
  BOOT-INF/lib/spring-boot-autoconfigure-1.4.2.RELEASE.jar
  BOOT-INF/lib/logback-classic-1.1.7.jar
  BOOT-INF/lib/jackson-core-2.8.4.jar
  BOOT-INF/lib/spring-boot-1.4.2.RELEASE.jar
  BOOT-INF/lib/spring-expression-4.3.4.RELEASE.jar
  BOOT-INF/lib/logback-core-1.1.7.jar
  BOOT-INF/lib/validation-api-1.1.0.Final.jar
  BOOT-INF/lib/classmate-1.3.3.jar
  BOOT-INF/lib/jcl-over-slf4j-1.7.21.jar
  BOOT-INF/lib/log4j-over-slf4j-1.7.21.jar
  BOOT-INF/lib/jboss-logging-3.3.0.Final.jar
  BOOT-INF/lib/jackson-annotations-2.8.4.jar
$ □
```

RIP WebLogic / WebSphere / GlassFish / WildFly / ...

Comparatively ...

1 day to install the server

1 week to read JEE doc

1 month to read specific doc

1 month to struggle in /console & wlst

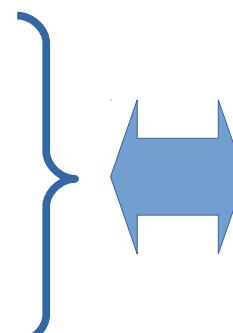
5 minutes to stop-restart a server

1 minute to rebuild

3 minutes to redeploy a war

(+ **5mn** : often restart anyway ..)

+ **1 minute** remote debugging



2 Seconds
to build / start / hot-restart

0 Value added

-10000 \$ on your bank account

either in Eclipse
Or java -jar

as Josh Long said in one of his talks
in the Spring IO
“it is better to make Jar, not War”.



Josh Long

the Spring Developer Advocate at Pivotal @starbuxman

Amazing Conferences
example at Devoxx



Home About Search Twitter Have an account? Log in

TWEETS 48.5K FOLLOWING 2,172 FOLLOWERS 13.2K LIKES 17.8K LISTS 19 Follow

Josh Long (龙之春, जोश)
@starbuxman

Spring Advocate @Pivotal
spring.io/team/jlong; @david_syer's my spirit animal; @pearson bit.ly/1HUCRWt; @OReillyMedia oreil.ly/1CiSz85

Mission, San Francisco, CA
joshlong.com
Joined April 2007

Pinned Tweet
Josh Long (龙之春, जोश) @starbuxman · 27 Oct 2015
"Getting Started w/ **@SpringCloudOSS**"
@SpringOne2GX 2015 w Dr. **@David_Syer** and me (& appearance by Dr. **@SpringRod!**)

Pivotal Getting Started with Spring Cloud
Recorded at SpringOne2GX 2015 Speakers: Dr. Dave Syer, Josh Long

New to Twitter?
Sign up now to get your own personalized timeline!

Sign up

You may also like · Refresh

Spring Boot @springboot
Phil Webb @philip_webb

#MakeJarNotWar @starbuxman

<https://twitter.com/hashtag/makejarnotwar>

← → C Twitter, Inc. [US] | <https://twitter.com/hashtag/makejarnotwar>

M G perso dev Misc

Home About #makejarnotwar

#makejarnotwar

Top | Latest | Accounts | Photos | Videos | More options ▾

New to Twitter?
Sign up now to get your own personalized timeline!

Sign up

Dimitri Hautot @D_H_ · 31 Dec 2015
Hey @starbuxman, in 2016, more than ever, #MakeJarNotWar !!! Happy New Year !!!

1 1 2 ...

Callum Watson @thecallumwatson · Mar 10
@starbuxman great presentation at @jpmorgan tech symposium today!#makejarnotwar

2 7 5 ...

Pulkit Kumar @pulkitkumar90 · Jul 20
Great talk by @starbuxman : bit.ly/2a8t3xd . Java is awesome.
#MakeJarNotWar

1 1 1 ...

vaadin @vaadin · May 19

About 2,040 results (0.35 seconds)

2000 "make jar not war"

Introduction to Spring Boot » blog

blog.mimacom.com/introduction-to-spring-boot/ ▾

Sep 30, 2015 - ... WAR file and deploy it in any server of your choice, but as Josh Long says talks in the Spring IO "it is better to **make Jar, not War**".

Make Jar, Not War - GitHub

<https://github.com/making/make-jar-not-war>

Make Jar, Not War. Contribute to make-jar-not-war · GitHub

(0 Articles : \$ 0.00)

SSL secure Help?

Images for "make jar not war"



More images for "make jar not war"

Make JAR not WAR ! - YouTube

<https://www.youtube.com/watch?v=JyfjwzXWQHg>

May 22, 2016 - Upload

Présentation en 30 n

... MVP Java 10,303

▶ 27:55

Make .jar, not .war
geeky t-shirt



Men's T-Shirt

Size

\$17.99

\$ 13.99

Make .jar, not .war
geeky t-shirt



Men's T-Shirt

Size

\$19.49

\$ 13.99

Make .jar, not .war
geeky t-shirt



Men's T-Shirt by American Apparel

Size

\$26.99

\$ 21.99

Make .jar, not .war
geeky t-shirt



Men's Ringer T-Shirt

Size

Make .jar, not .war
geeky t-shirt



Men's Hoodie

Size

Make .jar, not .war
geeky t-shirt



Women's T-Shirt

Size

App Server Embedded Server > Deployed War



Your JAR packaging also contains embedded tomcat



Your WAR is deployed into a WebServer

Historical Reasons 1 Server = 1 JVM – Several Apps

1 Server = 1 JVM process
... serves several Apps

JSP Web Portal calling other JSP ?

Web \$\$\$ Expensive => use 1 server for many apps

Limited RAM (1-8 G) ...
jvm overhead = 100M
=> share RAM on workstations

Problems:
bad isolation ...
OutOfMemory Error App1 => also crash App2,App3...
Redeploy / Restart App1 => stop all App2,App3...

