Pivotal

Spring Boot Primer

Getting Started with Spring Boot to build Cloud Native Apps.

Agenda

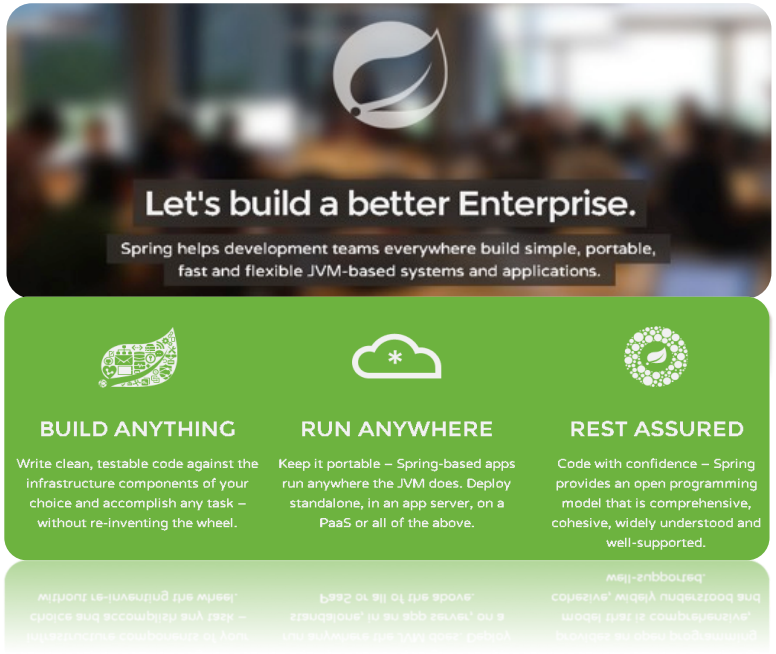
- Challenges building non-Boot applications
- What is Spring Boot?
- Capabilities
- Demos Throughout

Pivotal

What is Spring IO?

the #1 enterprise Java application development framework

- Embodiment of best industry dev practices
- The best of Java OSS, curated & improved
- OSS, Apache 2.0 licensed
- Proven, mature, since 2004
- Well understood and documented
- Millions of downloads
- Used heavily by Netflix and F2000




IO EXECUTION
Spring XD
Boot
IO COORDINATION
Cloud
IO FOUNDATION
Integration
Batch
Big Data
Web
Relational
Data Access
DATA
Non-Relational
Data Access
CORE
Framework
Security
Groovy
Reactor

Challenges writing non-Boot applications

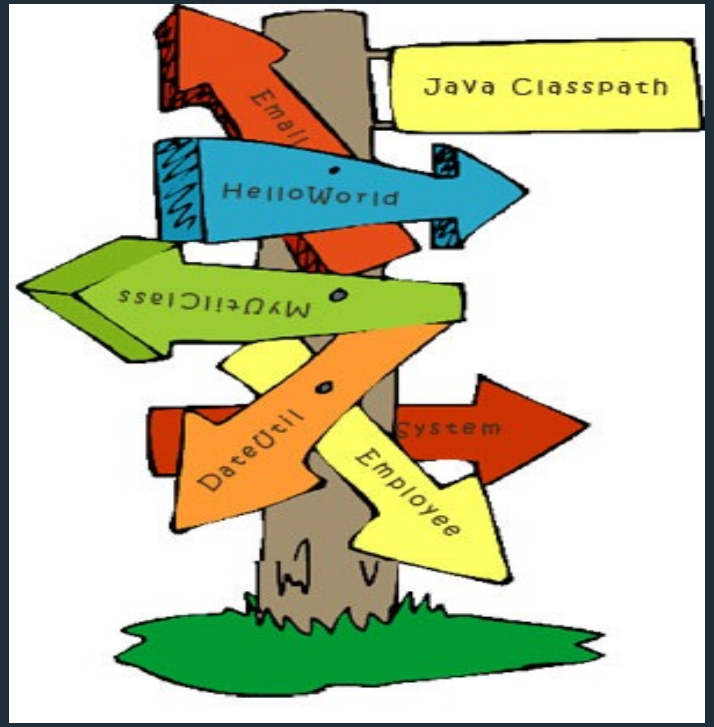
Version incompatibility of libraries

```
Exception in thread "main" java.lang.NoClassDefFoundError: javax/transaction/SystemException
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:344)
        at org.jboss.logging.Logger$1.run(Logger.java:2554)
        at java.security.AccessController.doPrivileged(Native Method)
        at org.jboss.logging.Logger.getMessageLogger(Logger.java:2529)
        at org.jboss.logging.Logger.getMessageLogger(Logger.java:2516)
        at org.hibernate.internal.CoreLogging.messageLogger(CoreLogging.java:28)
        at org.hibernate.internal.CoreLogging.messageLogger(CoreLogging.java:24)
        at org.hibernate.cfg.Configuration.<clinit>(Configuration.java:86)
        at com.mycompany.ContactManager.main(ContactManager.java:20)
Caused by: java.lang.ClassNotFoundException: javax.transaction.SystemException
        at java.net.URLClassLoader$1.run(URLClassLoader.java:372)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:360)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
        ... 10 more
```

Pivotal

Challenges building non-Boot applications

- `Classpath` errors
  - A Jar is missing
  - There is one Jar too many
  - A class is not visible where it should be



Pivotal

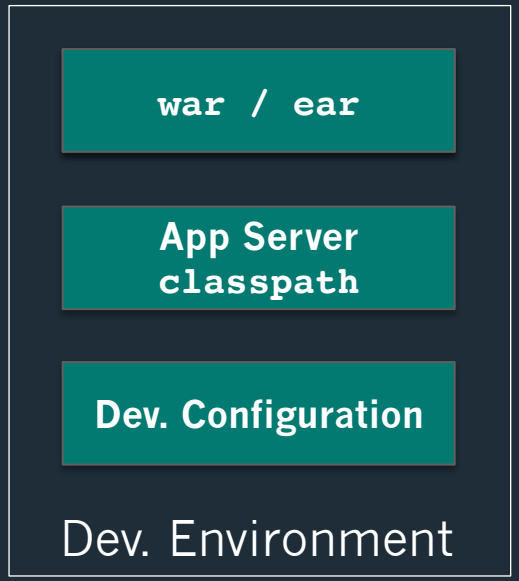
Configuration

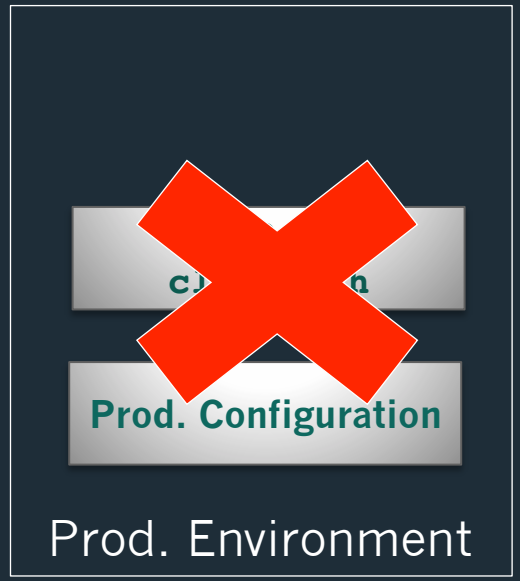
- Maven and Spring

Pivotal

Challenges building non-Boot applications

Standard `war/ear` packaging causes drift across environments



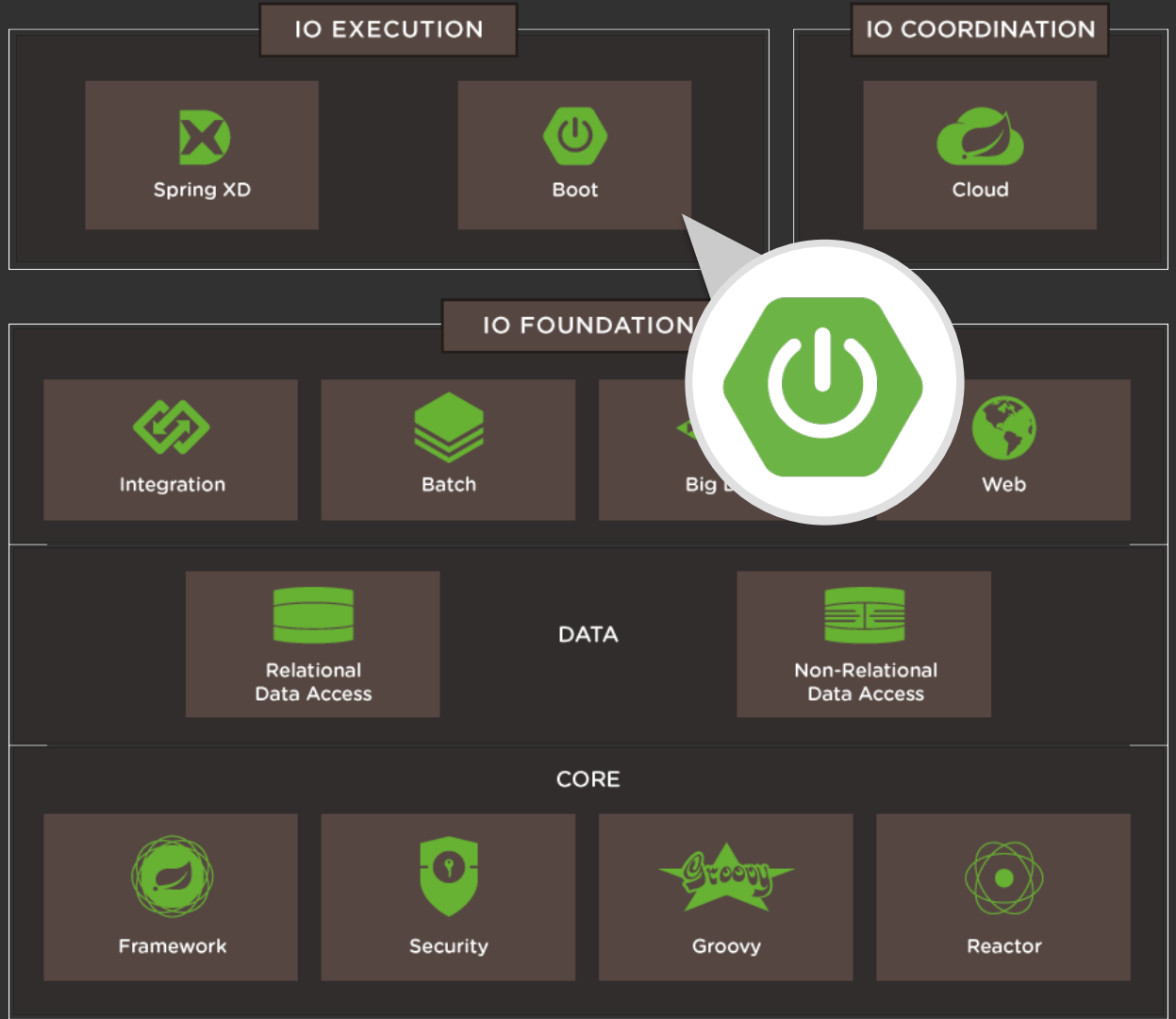


Pivotal

What is Spring Boot?

- Spring applications typically require a lot of setup
  - Consider working with JPA. You need:
    - Datasource, TransactionManager, EntityManagerFactory ...
  - Consider a web MVC app. You need:
    - WebApplicationInitializer / web.xml, ContextLoaderListener, DispatcherServlet, ...
  - An MVC app using JPA would need all of this

- *BUT: Much of this is predictable*
  - Spring Boot can do most of this setup for you


IO EXECUTION
Spring XD
Boot
IO COORDINATION
Cloud
IO FOUNDATION
Integration
Batch
Web
DATA
Relational
Data Access
Non-Relational
Data Access
CORE
Framework
Security
Groovy
Reactor

What is Spring Boot?

An opinionated framework to simplify bootstrapping and development of new Spring Applications

Pivotal

Spring Boot

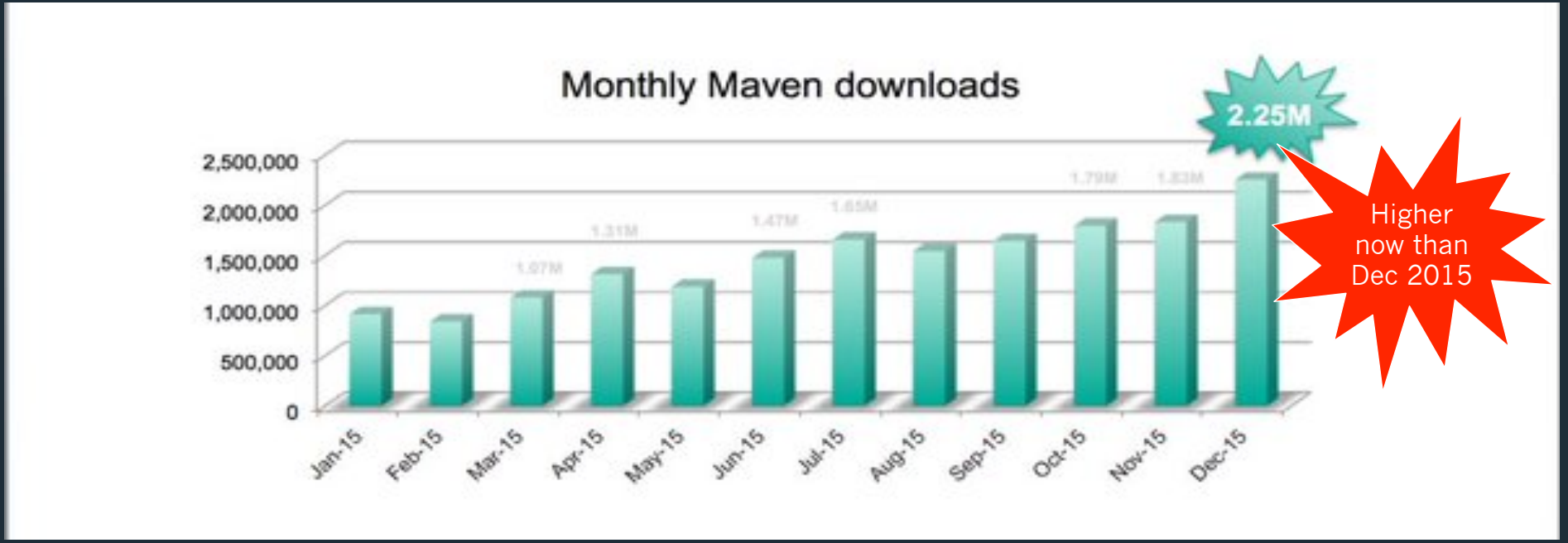
Dynamic language productivity with maturity of enterprise Java

Cloud Native: Direct support for Microservices, NetflixOSS ++

Fully automated app server configuration and deployment

Production ready Ops metrics out of the box, with a switch

Spring Boot Adoption



Pivotal

Capabilities

- Provides a consistent and easy way to a start project
- Manages project dependencies
- Packages deployable artifacts prevent configuration drift across environments
- Provides conditional configuration across environments
- Provides tools to improve developer productivity
- Keeps you from writing explicit configuration
- Exposes endpoints to manage and monitor applications in production

Pivotal


DEMO
CF

Spring Initializr

A configurable service to consistently and easily generate a quickstart project

Pivotal

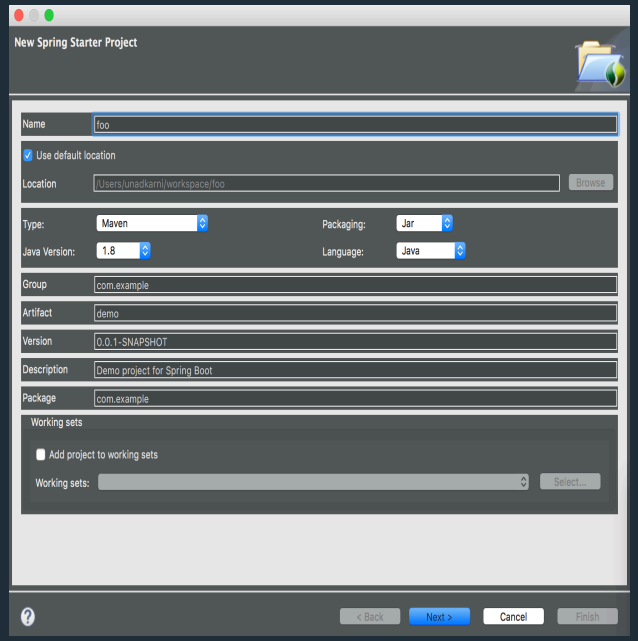
Spring Initializr

- Generates a Spring Boot project structure.
- Provides a Maven/Gradle build specification.
- Doesn’t generate application code.
- You can customize the Spring Initialzr
  - `https://github.com/spring-io/initializr/`

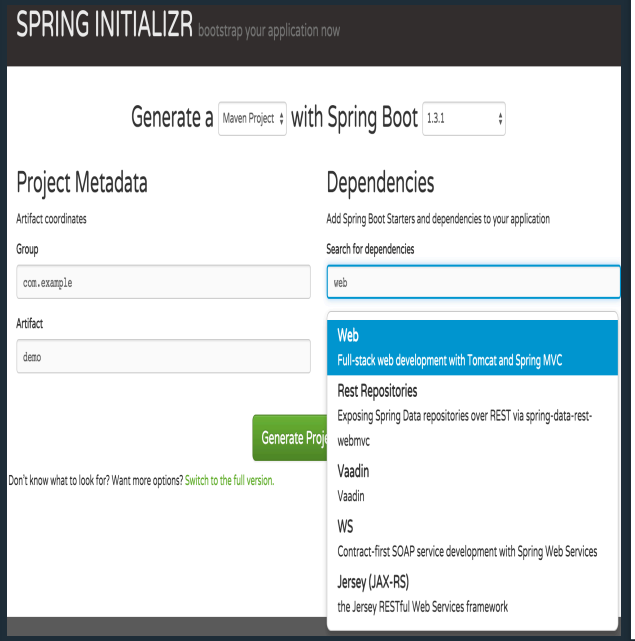
Pivotal

Supported Interfaces

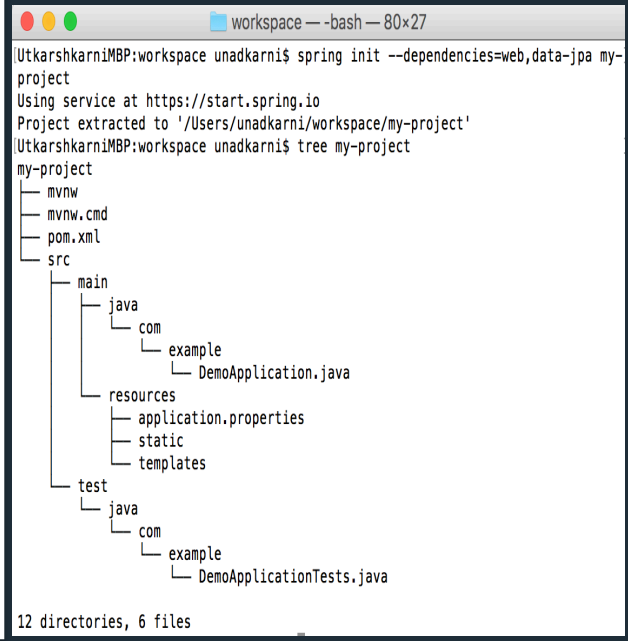
1. IDE



2. Web-based interface



3. Spring Boot CLI

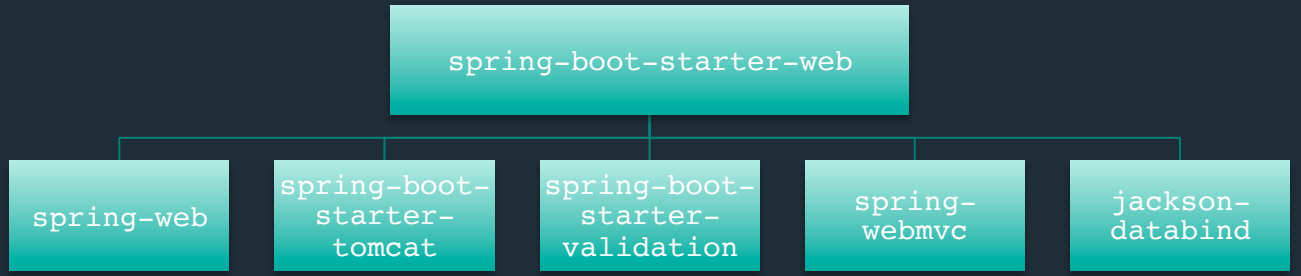


Pivotal


DEMO
CF

Spring Boot starters

- Are virtual packages deployed to Maven central
- They pull in other dependencies while containing no code of their own



Pivotal

Project `pom.xml`

- Set parent of your `pom.xml` as `spring-boot-starter-parent`
  - `Inherit preset version numbers`

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>demo</name>
    <description>Demo project for Spring Boot</description>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.3.1.RELEASE</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
```

Pivotal

Starters vs Current Practice

Starter versions determined by version of Spring Boot used.

No explicit version numbers. Guaranteed to be compatible.

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Concise. Boot pom.xml without version numbers.

Are the version numbers compatible with each other?

Is the list of dependencies complete?

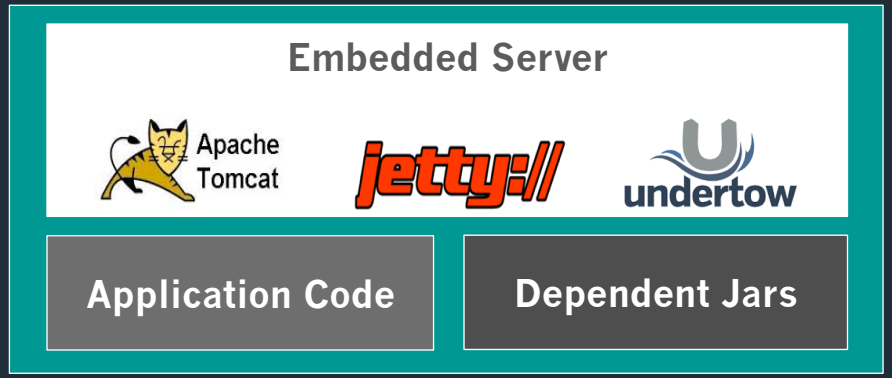
```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-web</artifactId>
        <version>4.1.6.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.thymeleaf</groupId>
        <artifactId>thymeleaf-spring4</artifactId>
        <version>2.1.4.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.data</groupId>
        <artifactId>spring-data-jpa</artifactId>
        <version>1.9.2.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-entitymanager</artifactId>
        <version>5.0.6.Final</version>
    </dependency>
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <version>1.4.190</version>
    </dependency>
</dependencies>
```

Verbose. Non-boot pom.xml with version numbers.

Pivotal

Packaging – Make Jar not War

Boot's Maven / Gradle plugins produce an executable *"fat jar"*




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

Pivotal

Benefits of executable *fat jars*

- No installation of application servers
- No setting `classpath`
- Promotes consistency across environments
- Cloud-friendly
- Can be started as Unix services using
  - `init.d`
  - `systemd`

Pivotal

Profiles

Segregate parts of the application configuration and make it available in certain environments via

- Annotations
- Properties file

Pivotal

@Profile

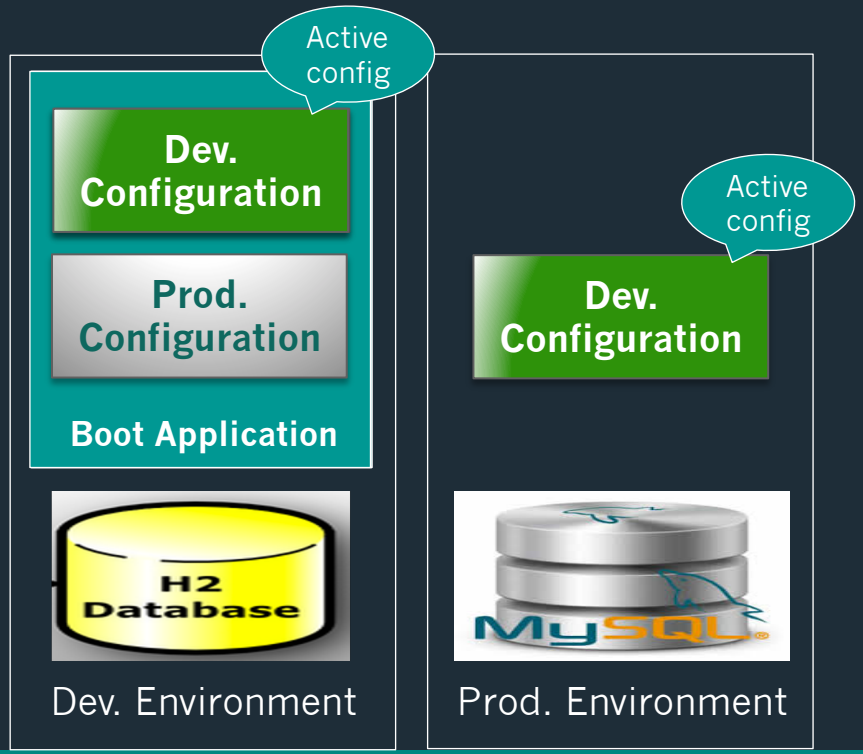
Mark `@Component` or `@Configuration` with `@Profile` to limit when it is loaded.

```
@Configuration
@Profile("development")
public class DevelopmentConfiguration {
    // The @Profile requires that the "development"
    // profile be active at runtime
    // for this configuration to be applied.
    // Activate this profile in
    // properties file with spring.profile.active=development
    // command line with --spring.profiles.active=development
}
```

```
@Configuration
@Profile("production")
public class ProductionConfiguration {
    // The @Profile requires that the "production"
    // profile be active at runtime
    // for this configuration to be applied.
    // Activate this profile in
    // properties file with spring.profile.active=production
    // command line with --spring.profiles.active=production
}
```

Pivotal

Profiles



Checkout code from source control

Build an executable jar

Configuration activated based on active profile

Deploy anywhere

No Code Changes

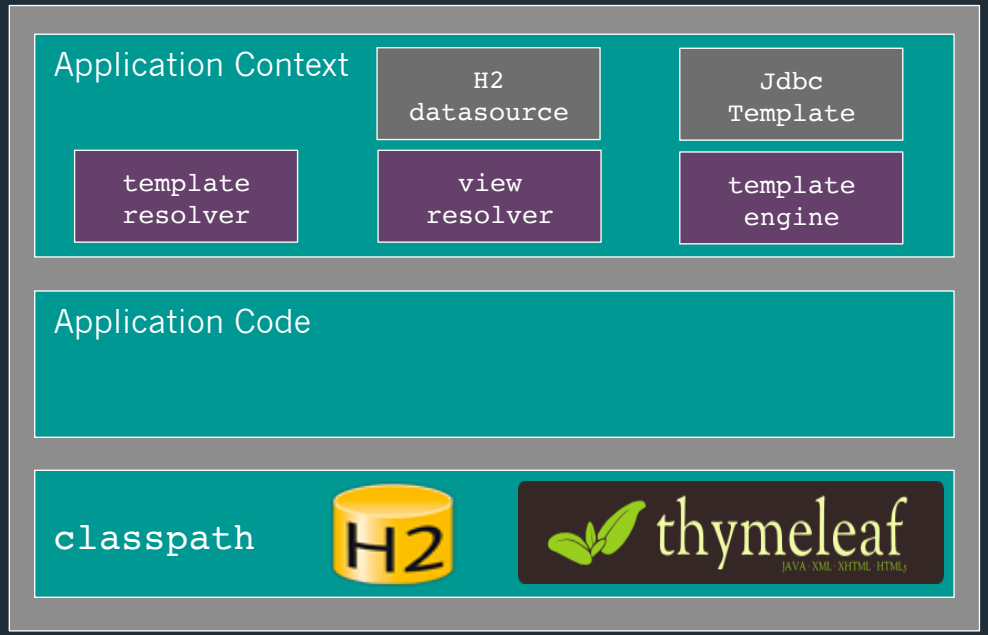
No Configuration Changes

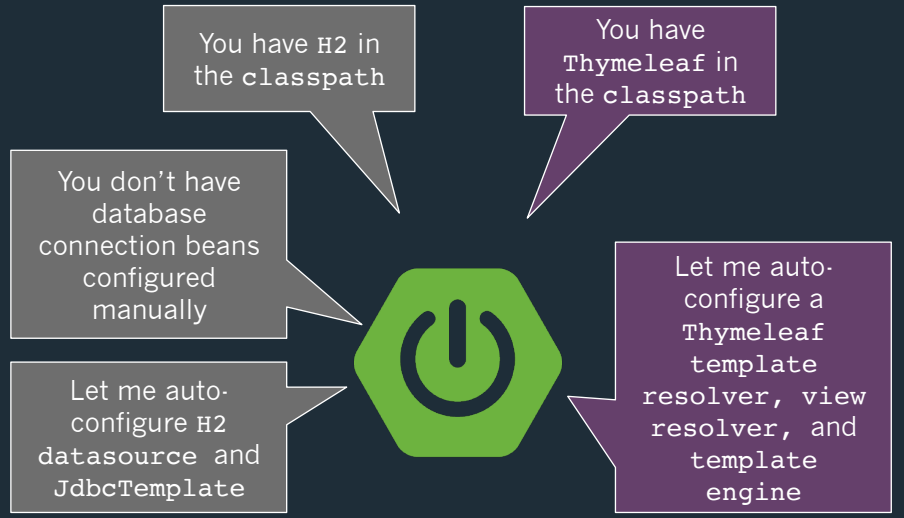
It Just Works!

Pivotal

Auto-Configuration

Automatically configures application based on the `classpath`

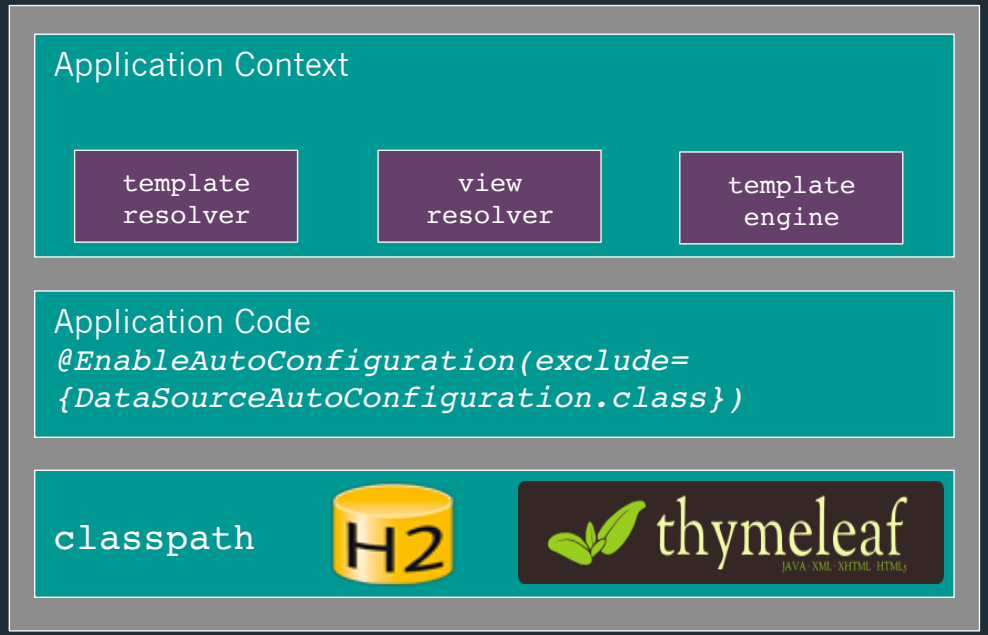


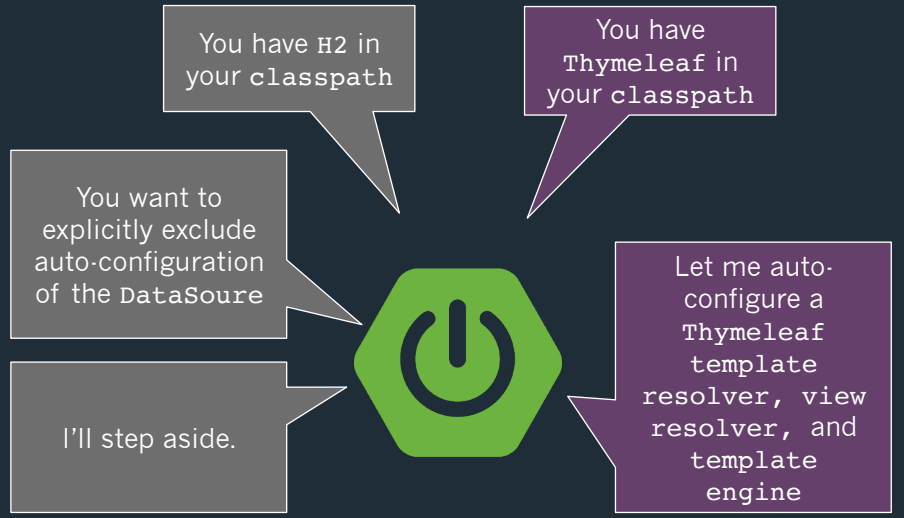


Pivotal

Override Auto-Configuration

Explicitly exclude auto-configuration





Pivotal


DEMO
CF

Actuator

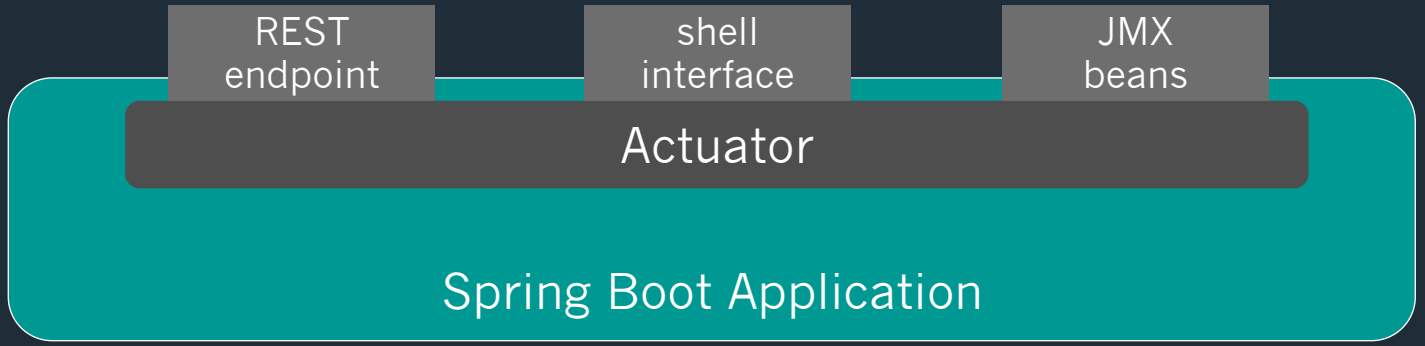
Offers production-ready features such as monitoring and metrics to improve operator efficiency

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

Pivotal

Actuator

- Services exposed through
  - HTTP endpoints for Spring MVC Applications
  - shell interface
  - JMX Beans



Pivotal

Types of Actuator Endpoints

Actuator endpoints can be organized into three categories

Configuration
- /beans
- /autoconfig
- /env
- /configprops
- /controller

Metrics
- /metrics
- /trace
- /dump

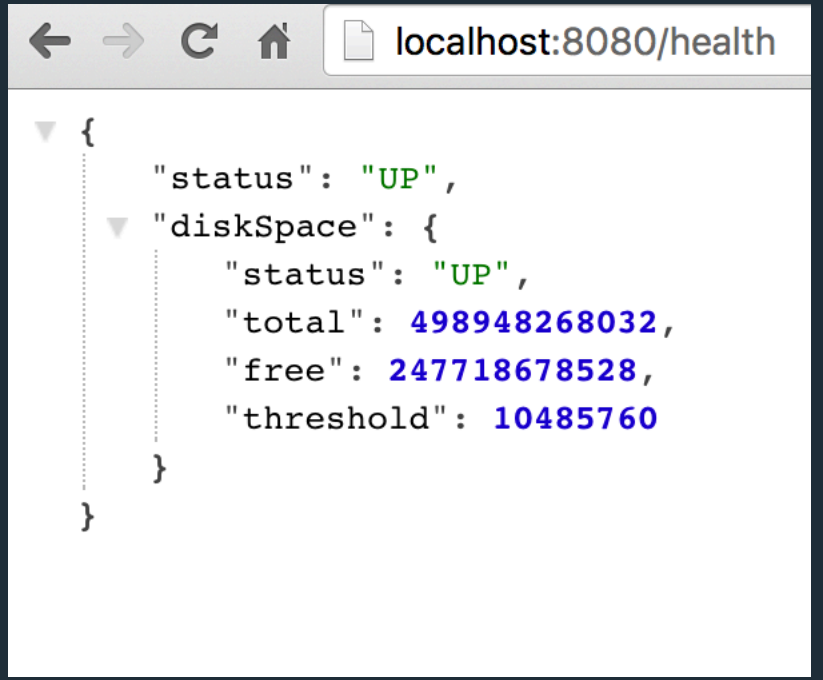
Miscellaneous
- /shutdown
- /info

Pivotal

/health

Reports health metrics for the application

- Each check returns a `Health` object
  - `status`
    - `UP`
    - `DOWN`
    - `UNKONWN`
    - `OUT_OF_SERVICE`
  - Possibly with details
- Plugin own health definitions

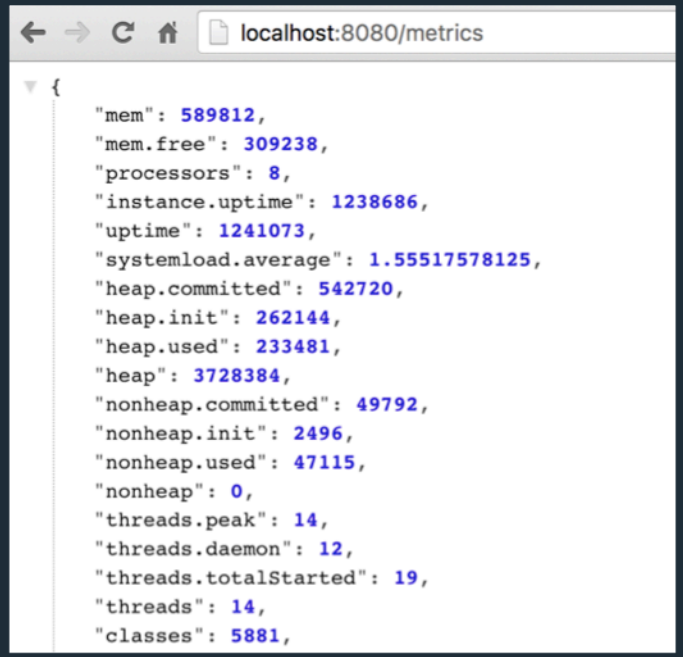


Pivotal

/metrics

Reports application metrics such as

- Memory usage and
- HTTP request counters



Pivotal


DEMO
CF

Spring Boot & Cloud Foundry

Pivotal


DEMO
CF

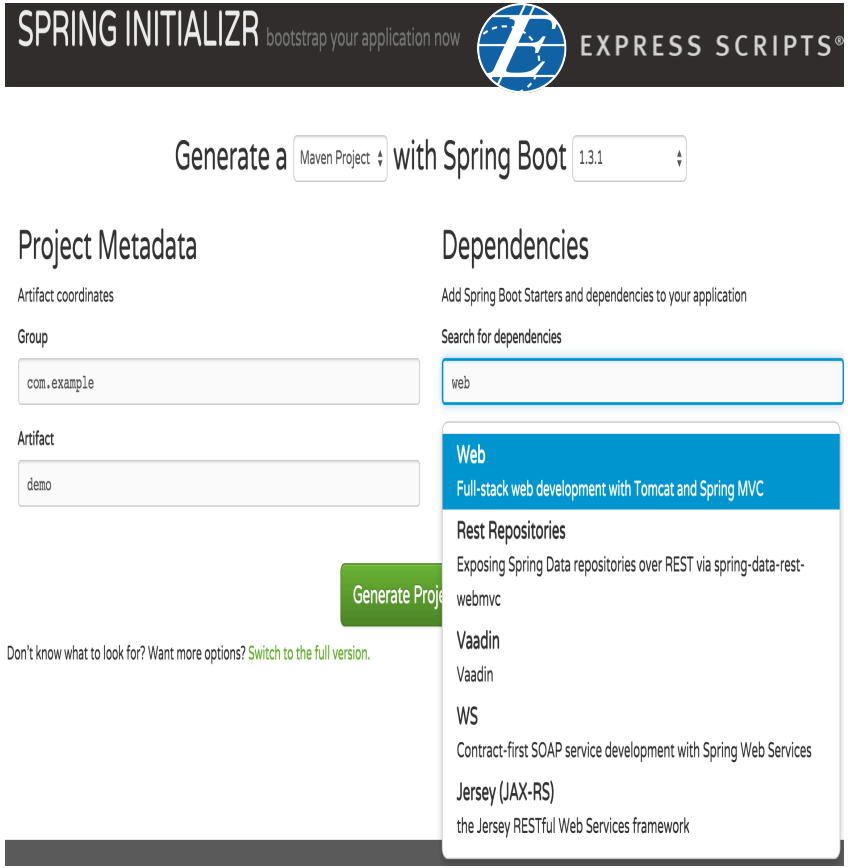
HATEOS

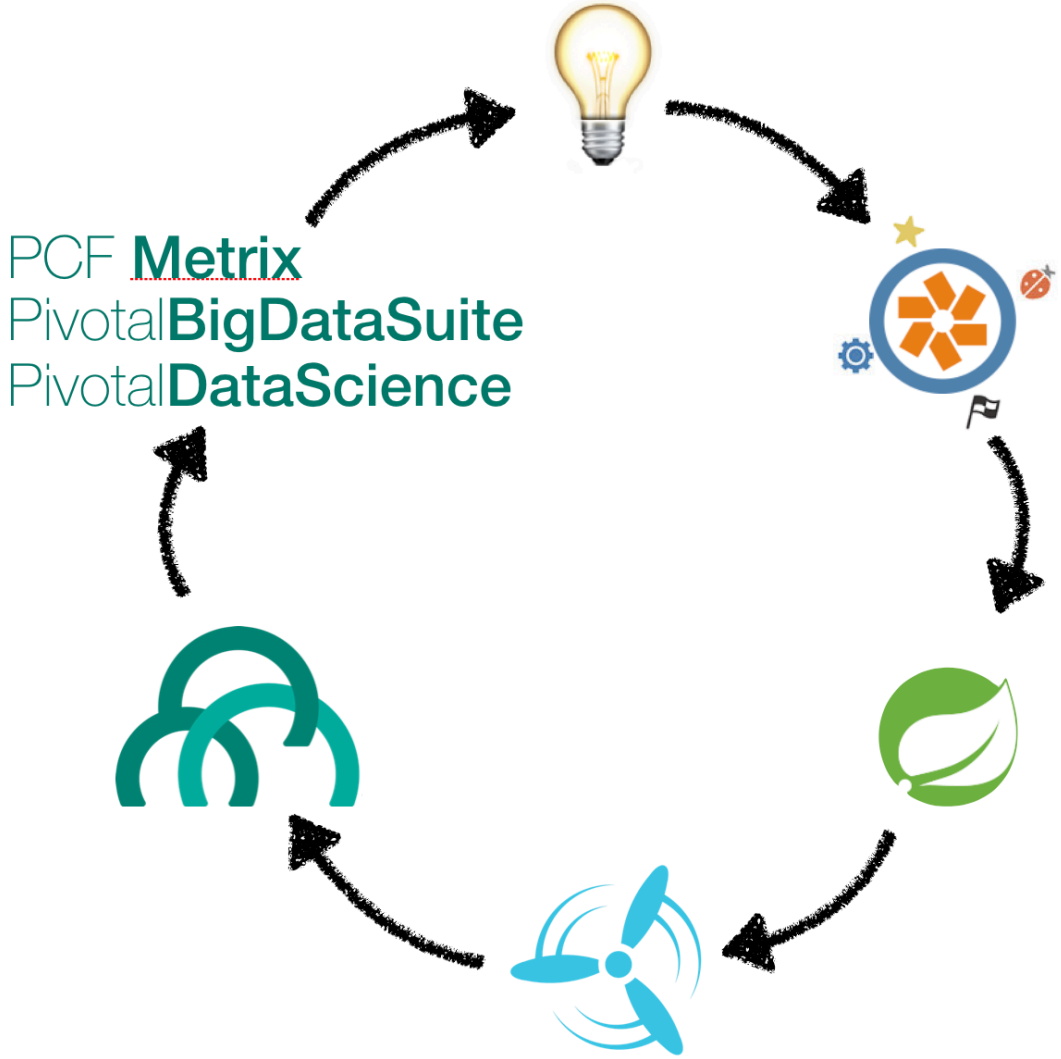
Pivotal

Spring Data Rest

Pivotal


DEMO
CF


SPRING INITIALIZR bootstrap your application now
EXPRESS SCRIPTS®
Generate a Maven Project with Spring Boot 1.3.1
Project Metadata
Artifact coordinates
Group
com.example
Artifact
demo
Dependencies
Add Spring Boot Starters and dependencies to your application
Search for dependencies
web
Web
Full-stack web development with Tomcat and Spring MVC
Rest Repositories
Exposing Spring Data repositories over REST via spring-data-rest-webmvc
Vaadin
Vaadin
WS
Contract-first SOAP service development with Spring Web Services
Jersey (JAX-RS)
the Jersey RESTful Web Services framework
Generate Proj
Don't know what to look for? Want more options? Switch to the full version.


PCF Metrix
PivotalBigDataSuite
PivotalDataScience