

Deploying Your Spring Boot Applications



Federico Mestrone
Software Engineer and Training Consultant

@fedmest www.federicomestrone.com



Library Overview



In this module:

- Prepare for production
 - Configure deployment artifacts
 - Ready for publishing to Maven repo
 - And share with the world
- Deploy to production
 - Package for running on servers
 - Run your application to scale
 - On premises
 - In the cloud



Getting Ready for Production



Production Checklist



Testing



Documentation



Instrumentation



Appeal



Production Checklist



Testing

Should never make it so far down the line...



Documentation

Make sure your tests cover all the fundamental aspects of your library



Instrumentation

- Never able to completely rule out bugs, BUT...
- If your tests pass, you should have confidence the product will not let users down



Appeal



Pointers for Testing

**Use @SpringBootTest and
@ContextConfiguration to
create right env for your tests**

**Create mock versions of
non-subject beans with
@MockBean**

**Activate specific profiles for
testing with @ActiveProfiles**

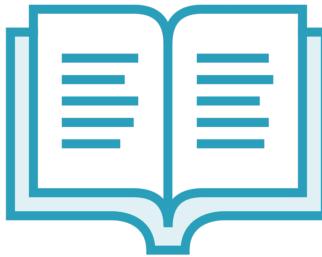
**Test web controllers with
@MockMvc**



Production Checklist



Testing



Documentation



Instrumentation



Appeal

Add JavaDoc comments

- Public
- Protected
- Default

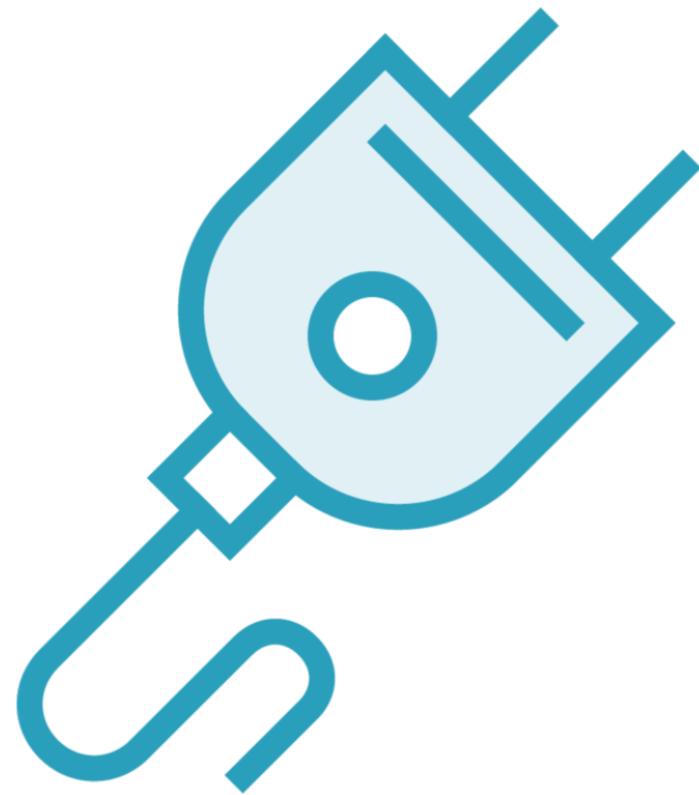
Provide installation and getting started guides

Provide the Javadoc reference documentation via a website

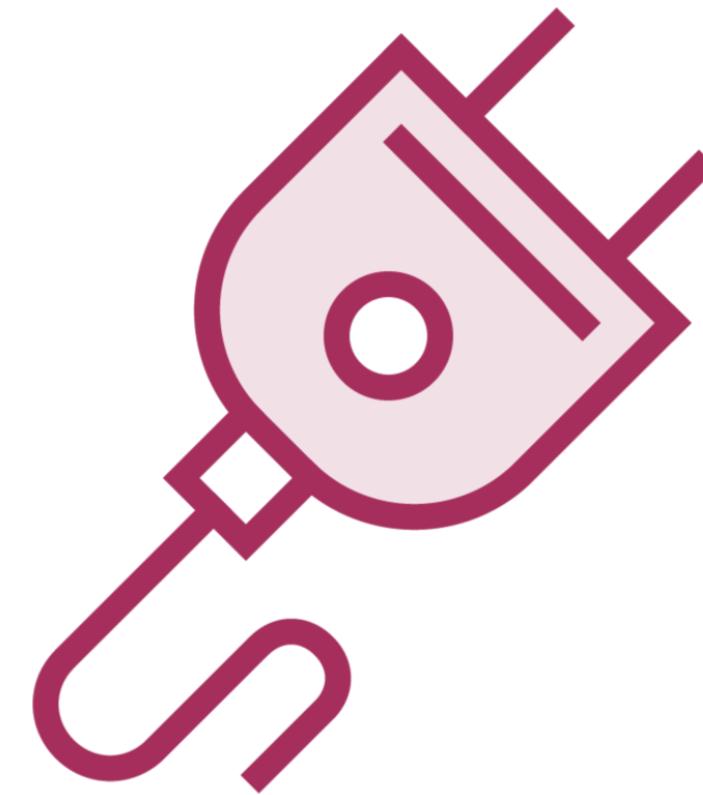
- With links to and from the guides



Generating JavaDoc Comments



**JAutoDoc Plugin
for Eclipse**



**JavaDocs Plugin
for IntelliJ**



Production Checklist



Testing

Provide adequate logging at various levels

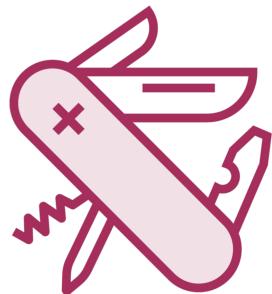
- Too much is no good either



Documentation

Allow for remote inspection and management

- Including health checks
- And “control levers”



Instrumentation



Appeal



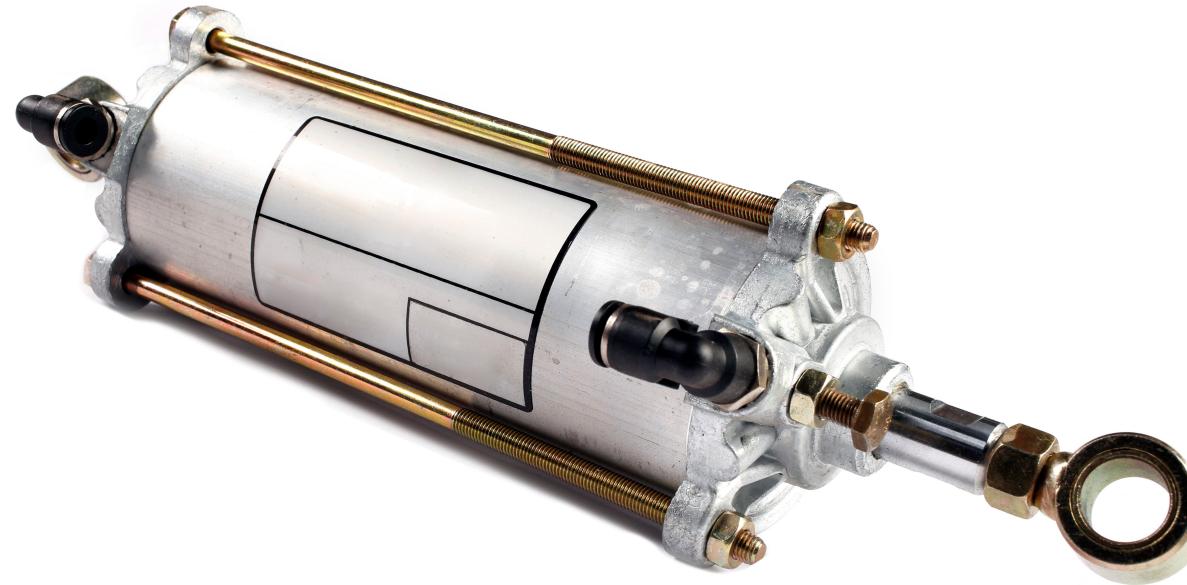
Spring Actuator



Helps you monitor and manage
your Spring Boot applications
once they reach production



Actuator Features



Technology-agnostic endpoints

- HTTP
- JMX

Individually enable/disable endpoints

Individually expose endpoints

- Technology-specific exposure

Many built-in endpoints

- But can always define your own



Built-in Endpoint Highlights

Endpoint	Result
/conditions	Conditional beans
/beans	All beans
/mappings	HTTP mappings
/loggers	Logger configuration
/env	All environment variables
/configprops	A list of all property POJOs
/health	The Health Indicator
/metrics	Various system metrics
/shutdown	Shut the application down gracefully



Adding Actuator Dependency (Maven)

```
<dependencies>

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

</dependencies>
```



Adding Actuator Dependency (Gradle)

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-actuator'  
}
```



Health Indicator

Reachable at /health

- One of the built-in endpoints

Gathers health status information from any bean that implements HealthIndicator

- Acts as a one-stop checkpoint for an application's health status
- Any library can contribute to the overall health status of the application
- If just one fails, the overall status is down too



Health Indicator Example

```
public class PdfferHealthIndicator implements HealthIndicator {  
    @Autowired  
    ApplicationContext context;  
  
    @Override  
    public Health health() {  
        String[] producer = context.getBeanNamesForType(PdfferProducerBean.class);  
        if (producer.length == 1) {  
            return Health.up().withDetail("ProducerBean", "OK").build();  
        } else if (producer.length == 0) {  
            return Health.down().withDetail("ProducerBean", "MISSING").build();  
        } else {  
            return Health.unknown().withDetail("ProducerBean", "TOO_MANY").build();  
        }  
    }  
}
```

{}



my-pdf-app

my-pdf-app > pom.xml

Project my-pdf-app ~/Downloads/my-pdf/.idea

Run: MyPdfAppApplication

Console Endpoints

Beans Health Mappings

application

diskSpace

- exists: true
- free: 597,570,523,136
- threshold: 10,485,760
- total: 1,000,345,825,280

pdffer

- ProducerBean: OK

ping

Database Maven

Structure

Favorites

Web

Run TODO Problems Terminal Profiler Build Endpoints Spring Event Log

All files are up-to-date (7 minutes ago)

Management Endpoints

Offer a door into the application

- For inspecting it and managing it at runtime

Will make debugging and fixing easier

Defined as Spring Beans with @Endpoint

- Endpoint operations can have Read, Write, or Delete semantics



Management Endpoint Example

```
@Component  
@Endpoint(id="pdfferWebControllerProps")  
public class WebControllerInfoEndpoint {  
  
    @Autowired  
    private PdfferWebControllerProps webControllerProps;  
  
    @ReadOperation  
    public PdfferWebControllerProps props() {  
        return webControllerProps;  
    }  
}
```



Enabling and Exposing Endpoints

```
management:  
  endpoint:  
    pdfferWebControllerProps:  
      enabled: true  
    pdfferMailerControllerProps:  
      enabled: true  
    pdfferMailerProps:  
      enabled: true  
    pdfferRegistryInfo:  
      enabled: true  
endpoints:  
  web:  
    exposure:  
      include: >-[  
        pdfferWebControllerProps, pdfferMailerControllerProps, pdfferMailerProps,  
        pdfferRegistryInfo, health, metrics, conditions, loggers, configprops, mappings  
  jmx:  
    exposure:  
      include: >-[  
        pdfferWebControllerProps, pdfferMailerControllerProps, pdfferMailerProps,  
        pdfferRegistryInfo, health, metrics, conditions, loggers, configprops, mappings
```





>	DefaultDomain
>	JMImplementation
>	com.sun.management
>	java.lang
>	java.nio
>	java.util.logging
>	jdk.management.jfr
>	org.springframework.boot
>	Admin
>	Endpoint
>	Beans
>	Caches
>	Conditions
>	Configprops
>	Env
>	Health
>	Info
>	Loggers
>	Mappings
>	Metrics
>	PdfferMailerControllerProps
>	PdfferMailerProps
>	PdfferRegistryInfo
>	PdfferWebControllerProps
>	Operations
>	props
>	Scheduledtasks
>	Threaddump

Operation invocation

java.util.Map **props** ()

MBeanOperationInfo

Name	Value
Operation:	
Name	props
Description	Invoke props for endpoint pdfferWebControllerProps
Impact	INFO
ReturnType	java.util.Map

Descriptor

Name	Value
Operation:	
descriptorType	operation
displayName	props
name	props
role	operation



Overview | Memory | Threads | Classes | VM Summary MBeans

Operation invocation
java.util.Map props ()

MBeanOperationInfo

Name	Value
Operation:	props
Name	props
Description	Invoke props for endpoint pdfferWebControllerProps
Impact	INFO
ReturnType	java.util.Map

Descriptor

Name	Value
Operation:	props
descriptorType	operation
displayName	props
name	props
role	operation

baseUri=nekopdf
downloadUri=download
saveUri=save

OK

Operations

props

Scheduledtasks

Threaddump

Production Checklist



Testing



Documentation



Instrumentation



Appeal

Make it pleasant, cool, and appealing!

- Nice homepage
- Attractive documentation
- Friendly and beautiful UI



Custom Banners



Spring Boot Standard Banner

```
> java -jar terminal-pdf-generator-1.0.0.jar --pdffer.web.controller.base-uri=module6d \
--pdffer.web.controller.save-uri=store4
```

Base URI for this instance of PDFer:
module6d

Full URL for Save endpoint:
`http://localhost:8080/module6d/store4`

Customizing the Spring Banner

Easiest way

- Place a file *banner.txt* at the root of your classpath

If you want a different location

- Indicate where the file is with the *spring.banner.location* property

You can also use images

- They will be converted to ASCII Art
 - With varying degrees of success!



Banner Placeholders

Placeholder	Meaning
<code> \${application.title}</code>	The title of the application as provided by the JAR Manifest file
<code> \${application.version}</code>	The version of the application as provided by the JAR Manifest file
<code> \${application.formatted-version}</code>	(v <code> \${application.version}</code>)
<code> \${spring-boot.version}</code>	The version of Spring Boot
<code> \${spring-boot.formatted-version}</code>	(v <code> \${spring-boot.version}</code>)
<code> \${AnsiColor.<NAME>}</code>	The text color
<code> \${AnsiBackground.<NAME>}</code>	The background color
<code> \${AnsiStyle.<NAME>}</code>	The text style (regular, bold, italic...)



banner.txt

```
 ${AnsiColor.BRIGHT_CYAN}      #####  #####  #####  #####  ${AnsiColor.BRIGHT_GREEN}  /\   /\  
 ${AnsiColor.BRIGHT_CYAN} #  #  #  #  #  #  #####  #####  #####  ${AnsiColor.BRIGHT_GREEN} { `---' }  
 ${AnsiColor.BRIGHT_CYAN} #  #  #  #  #  #  #  #  #  #  ${AnsiColor.BRIGHT_GREEN} { 0 0 }  
 ${AnsiColor.BRIGHT_CYAN} #  #  #####  #  #  #####  #####  #  #  ${AnsiColor.BRIGHT_GREEN} ~~> V <~~  
 ${AnsiColor.BRIGHT_CYAN} #  #  #  #  #  #  #  #  #  #####  ${AnsiColor.BRIGHT_GREEN} \ \| / /  
 ${AnsiColor.BRIGHT_CYAN} #  #  #  #  #  #  #  #  #  #  ${AnsiColor.BRIGHT_GREEN} `-----'  
 ${AnsiColor.BRIGHT_CYAN}          #  #####  #  #  #####  #  #  ${AnsiColor.BRIGHT_GREEN} / \ \\  
  
 ${AnsiColor.BRIGHT_MAGENTA}${application.title}  
 ${AnsiColor.DEFAULT}Artifact Version: ${AnsiColor.BRIGHT_YELLOW}${application.formatted-version}  
 ${AnsiColor.DEFAULT}Spring Boot Version: ${AnsiColor.BRIGHT_YELLOW}${spring-boot.formatted-version}  
 ${AnsiColor.DEFAULT}
```





```
> cd Downloads/my-pdf-app
```

```
~/Downloads/my-pdf-app is 📦 0.0.1-SNAPSHOT via ☕ v16.0.2
> java -jar target/my-pdf-app-0.0.1-SNAPSHOT.jar
```

```
#####
# #   ##### # #   #####
# #   # #   # #   # #
# #   # #   # #   # #
# #   ##### # #   ##### # #
# #   # #   # #   # #
# #   # #   # #   # #
# #   # #   # #   # #
# #   ##### # #   ##### # #
# #   # #   # #   # #
```

```
\\`---'
{ 0 0
~~> \ <~~
\ \|/
/ \\\-----'
```

my-pdf-app

Artifact Version: (v0.0.1-SNAPSHOT)

Spring Boot Version: (v2.5.3)

```
2021-09-27 18:28:44.313 INFO 51353 --- [           main] o.n.my.mypdfapp.MyPdfAppApplication : Starting MyPdfAppApplication v0.0.1-SNAPSHOT using Java 16.0.2 on FedericonoiMac-Pro.local with PID 51353 (/Users/federico/Downloads/my-pdf-app/target/my-pdf-app-0.0.1-SNAPSHOT.jar started by federico in /Users/federico/Downloads/my-pdf-app)
2021-09-27 18:28:44.316 INFO 51353 --- [           main] o.n.my.mypdfapp.MyPdfAppApplication : No active profile set, falling back to default profiles: default
2021-09-27 18:28:45.660 INFO 51353 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2021-09-27 18:28:45.673 INFO 51353 --- [           main] o.apache.catalina.core.StandardService : Start
```

Online Ascii Art Creator

HOME PRIVACY CONTACT DONATE EN DE ES

ascii-art-generator.org

WELCOME

This free online Ascii Art creator allows you to convert images to color or monochrome Ascii Art. You can also convert text to Ascii Banners.

For colored Ascii Art we provide a wide range as output formats, including SVG.

Convert: Image to Monochrome Ascii Art
 Image to Color Ascii Art
 Text to Ascii Art Banner

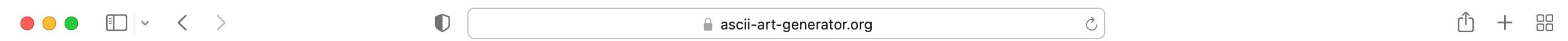
Banner text: PDFfer

Font: banner

Max. line width: 100

Line breaks are inserted to ensure text will not be longer than given number of characters. Leave empty for no line breaks.

Start



Online Ascii Art Creator

HOME

PRIVACY

CONTACT

DONATE

EN DE ES

ascii-art-generator.org

Result

Download result: [banner.txt](#) (368 B)

Remember to use a monospaced font to show this banner.

Preview

```
##### ###### #######  
# # # # # # # # #  
# # # # # # # # #  
##### # # ##### ##### # #  
# # # # # # # # #  
# # # # # # # # #  
# ##### # # # ##### #
```

[Back to start](#)



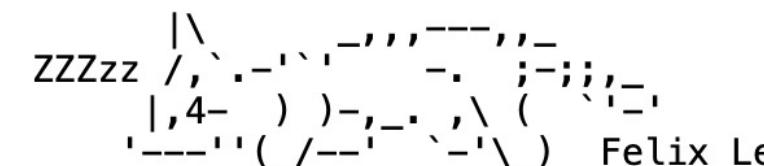
≡ ASCII Art Archive

Color Themes ▾

ASCII Art Menu

[ASCII Art FAQ](#)[ASCII Artists](#)[ASCII One Line](#)[ASCII Table ↗](#)

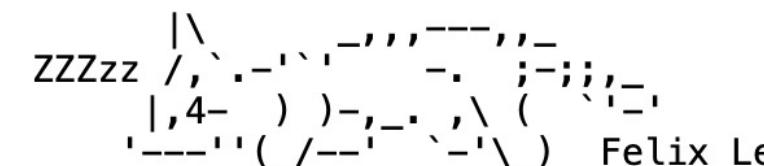
Categories

[Animals](#)[Art and design](#)[Books](#)[Buildings & places](#)[Cartoons](#)[Clothing & accessories](#)[Comics](#)[Computers](#)[Electronics](#)[Food and drinks](#)[Holiday & events](#)[Logos](#)[Miscellaneous](#)

A large collection of ASCII art drawings of cats and other related animal ASCII art pictures.



cat - domestic cats - kitten - kitty - pussycat - felines - lynx - fliis catus - pets

[Home](#) / [Animals](#) / [Cats](#)

Disabling the Spring Banner

SpringApplication class

```
new SpringApplicationBuilder(  
    MyPdfAppApplication.class  
)  
    .bannerMode(Banner.Mode.OFF)  
    .run(args);
```

application.yaml

```
spring:  
  main:  
    banner-mode: off
```

Banner Display Modes

Mode	Meaning
Banner.Mode.OFF	The banner is not displayed at all
Banner.Mode.LOG	The banner is displayed as part of the Spring standard logs
Banner.Mode.CONSOLE	The banner is displayed in the application's standard output stream



Publishing Your Artifacts



The Maven Central Repo

Offers access to a huge range of libraries

Used for dependency management by

- Maven, of course
- And practically every other Java build tool

Libraries are uniquely identified by

- A group ID
- An artifact ID
- A version number

Once published, an artifact cannot be modified

- But new versions can be added

Anyone can contribute libraries to it

- The process is known as “publishing”



Before Publication

Create an account at Sonatype

- Create a new JIRA ticket to get permission to publish under your Group ID

Configure your build tool for publication

- Where to publish
- What credentials to use



New Maven Central security capabilities. [Click for more info](#)

All Systems Operational

**PUBLISH****Producers**

Individuals

Large Organizations

> Introduction

Publishing my artifact[Getting started](#)

Requirements

> Sonatype OSSRH (OSS Repository Hosting) uses [Sonatype Nexus Repository Manager](#) to provide repository hosting service for open source project binaries - be sure to review the full [terms of service](#). OSSRH uses the Maven repository format and allows you to:

- deploy development version binaries (snapshots)
- stage release binaries
- promote release binaries and sync them to the Central Repository

Maven

Gradle

Ant

SBT

Manual

Releasing my artifact

Releasing

Producer Terms

Once released/published, you will not be able to remove/update/modify that artifact

We provide the option to publish artifacts using the `-SNAPSHOT` suffix in case that you need to do any test on your publishing process, but once it is released there is no possibility to change it. Please check the [Can I change \(modify, delete, update\) a component on Central? FAQ](#) for more details.

The initial setup for your OSSRH repository requires some manual steps and human review ([see why](#)), after which your deployment process is typically modified to get components into OSSRH. **These are all one time steps.**

Table of contents

Introduction

Initial Setup

Create a ticket with Sonatype

Review Requirements

Deployment

Releasing to Central

OSSRH Usage Notes

Accessing Repositories

POM File Config for Publishing

```
<distributionManagement>
  <repository>
    <id>ossrh</id>
    <name>Releases</name>
    <url>https://s01.oss.sonatype.org/service/local/staging/deploy/maven2</url>
  </repository>
  <snapshotRepository>
    <id>ossrh</id>
    <name>Snapshot</name>
    <url>https://s01.oss.sonatype.org/content/repositories/snapshots</url>
  </snapshotRepository>
</distributionManagement>
```



Publishing Credentials

```
<!-- settings.xml (this file should usually sit under ~/.m2) -->
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    https://maven.apache.org/xsd/settings-1.0.0.xsd">

  <servers>
    <!-- ... -->
    <server>
      <id>ossrh</id>
      <username>yourusername</username>
      <password>yourpassword</password>
    </server>
  </servers>

</settings>
```



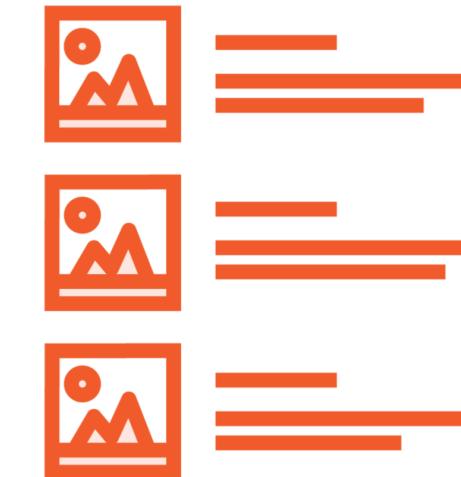
Maven Release Requirements



JavaDoc + Sources



GPG/PGP Signing



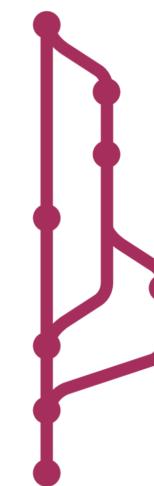
Name, Desc, URL



License



Developer Info



Source Code Repo



Attaching JavaDoc Artifacts

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>3.3.1</version>
  <executions>
    <execution>
      <id>attach-javadocs</id>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



Attaching Source Artifacts

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>3.2.1</version>
  <executions>
    <execution>
      <id>attach-sources</id>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



Signing with GPG

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-gpg-plugin</artifactId>
  <executions>
    <execution>
      <id>sign-artifacts</id>
      <phase>verify</phase>
      <goals>
        <goal>sign</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



```
~/Downloads/my-pdf-app is 📦 0.0.1-SNAPSHOT via ☕  
➤ gpg --gen-key  
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.
```

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

```
Real name: PDFfer Developer  
Email address: dev@pdffer.domain  
You selected this USER-ID:  
"PDFfer Developer <dev@pdffer.domain>"
```

Change (N)ame, (E)mail, or (O)kay/(Q)uit? 0

```
gpg: /Users/pdfferdev/.gnupg/trustdb.gpg: trustdb created  
gpg: key D6BD0629F1E19712 marked as ultimately trusted  
gpg: directory '/Users/pdfferdev/.gnupg/openpgp-revocs.d' created  
public and secret key created and signed.
```

```
pub ed25519 2021-09-27 [SC] [expires: 2023-09-27]  
    77A9CAFD432C2195735269185910D6BD0629F1E19712  
uid          PDFfer Developer <dev@pdffer.domain>  
sub cv25519 2021-09-27 [E] [expires: 2023-09-27]
```

Name, Description, and URL

```
<groupId>org.nekosoft.pdffer</groupId>
<artifactId>pdfser-template</artifactId>
<version>1.0-PLURALSIGHT</version>
<name>PDFfer Template</name>
<description>The definition of a PDFfer Template interface for the core library and
custom template libraries</description>
<url>https://pdfser.nekosoft.org</url>
```



License Information

```
<licenses>
  <license>
    <name>The Apache License, Version 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
  </license>
</licenses>
```



Developer Information

```
<developers>
  <developer>
    <name>Some Developer</name>
    <email>some@development.company</email>
    <organization>Some Development Co.</organization>
    <organizationUrl>http://some.development.company</organizationUrl>
  </developer>
</developers>
```



Source Code Repository

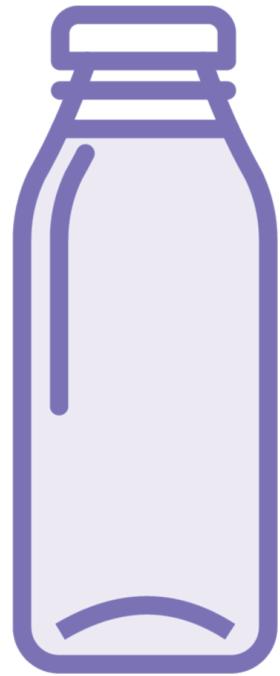
```
<scm>
  <connection>scm:git:git@github.com:YourApp/app.git</connection>
  <url>https://github.com/YourApp/app</url>
</scm>
```



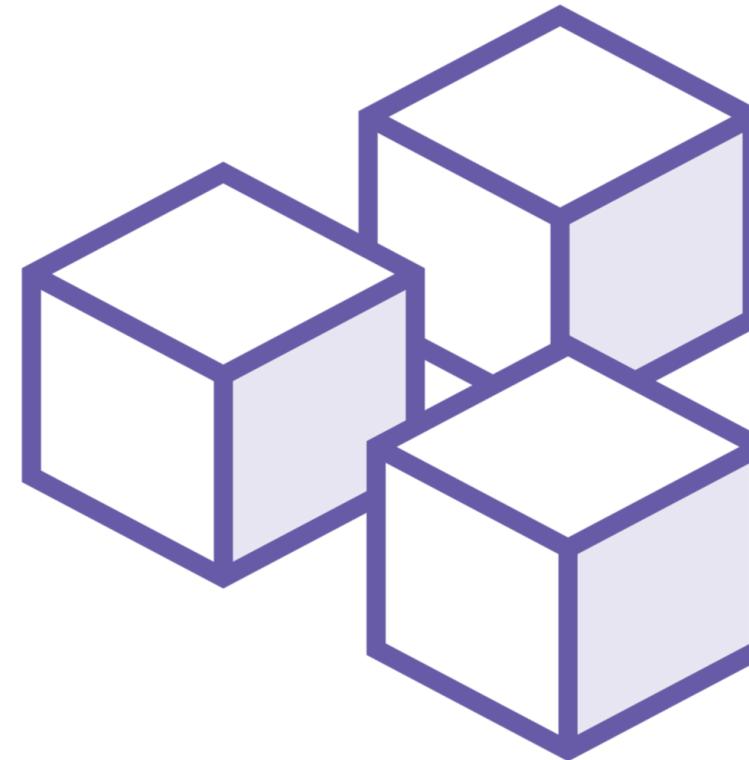
Deploying Your Spring Applications



Deployment Options



Fat JAR
The standard approach



Docker/K8s On Prem
The DIY approach to high scalability and portability



In the Cloud
The LTDI^(*) approach to high scalability and peace of mind

(*) Let Them Do It



Fat JAR

A JAR file that contains everything the application needs

- Classes and resources
- Dependencies
- Start-up code provided by Spring Boot
- Application metadata

Prepared by the Spring Maven plugin

- Application ready in seconds
- But not easily scalable
- Portability might be an issue



Docker & K8s

Containers capture an entire environment

- Like virtual machines but lightweight

Docker: most popular containerization engine

Kubernetes (K8s): an orchestrator for containers

- Uses Docker for container operations
- Offers abstraction for containers to collaborate or cooperate with each other



In the Cloud

Cloud providers offer

- Solid large-scale infrastructure
 - At reasonable costs
- Ease of deployment
- Effortless scalability

Kubernetes is easily scalable and portable

- Easily moved to the cloud



Google AppEngine Standard

Offers all advantages you expect

- Deployment is a breeze
- Scales quickly from zero to ‘000s and back
- Only pay for resources you use

Is aimed at a few specific platforms

- Spring Boot is one of them
- Need to use Java 11



What Next?





Create your own Spring Boot libraries

Learn more!

- There is so much still to discover

Use PDFfer in your own products

Contribute to improving PDFfer

Create and share your own PDFfer templates





Thank you!

