

Spring Boot

Another Perspective



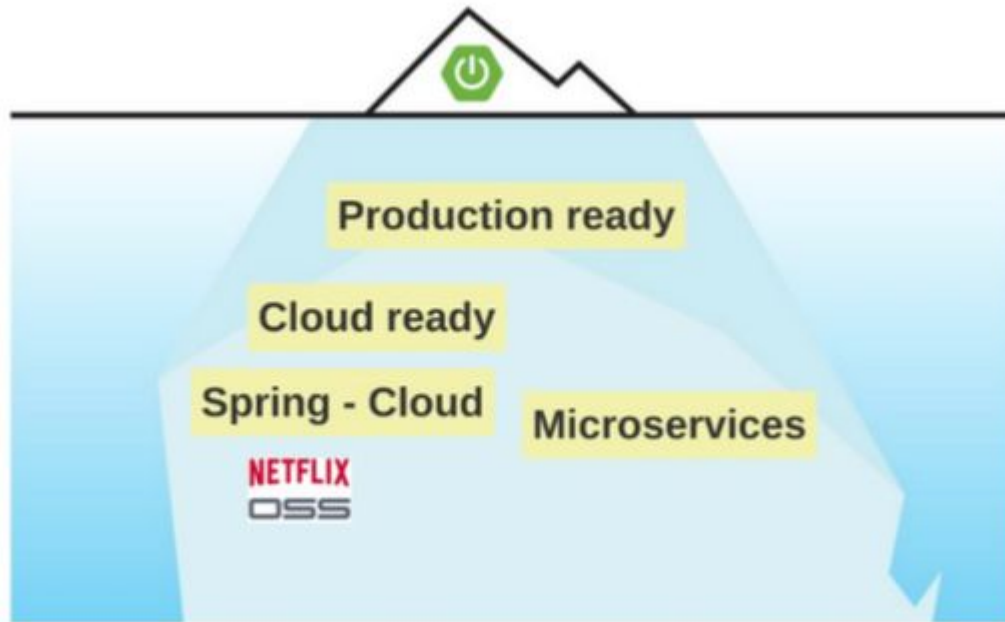
“You can’t make a delicious cake without the correct ingredients!”

https://twitter.com/phillip_webb/status/641444531867680768

Hello Spring Boot 2.x

Why ?

Application skeleton generator
Reduce effort to add new technologies



What ?

Embedded application server

Integration with tools/technologies (starter)

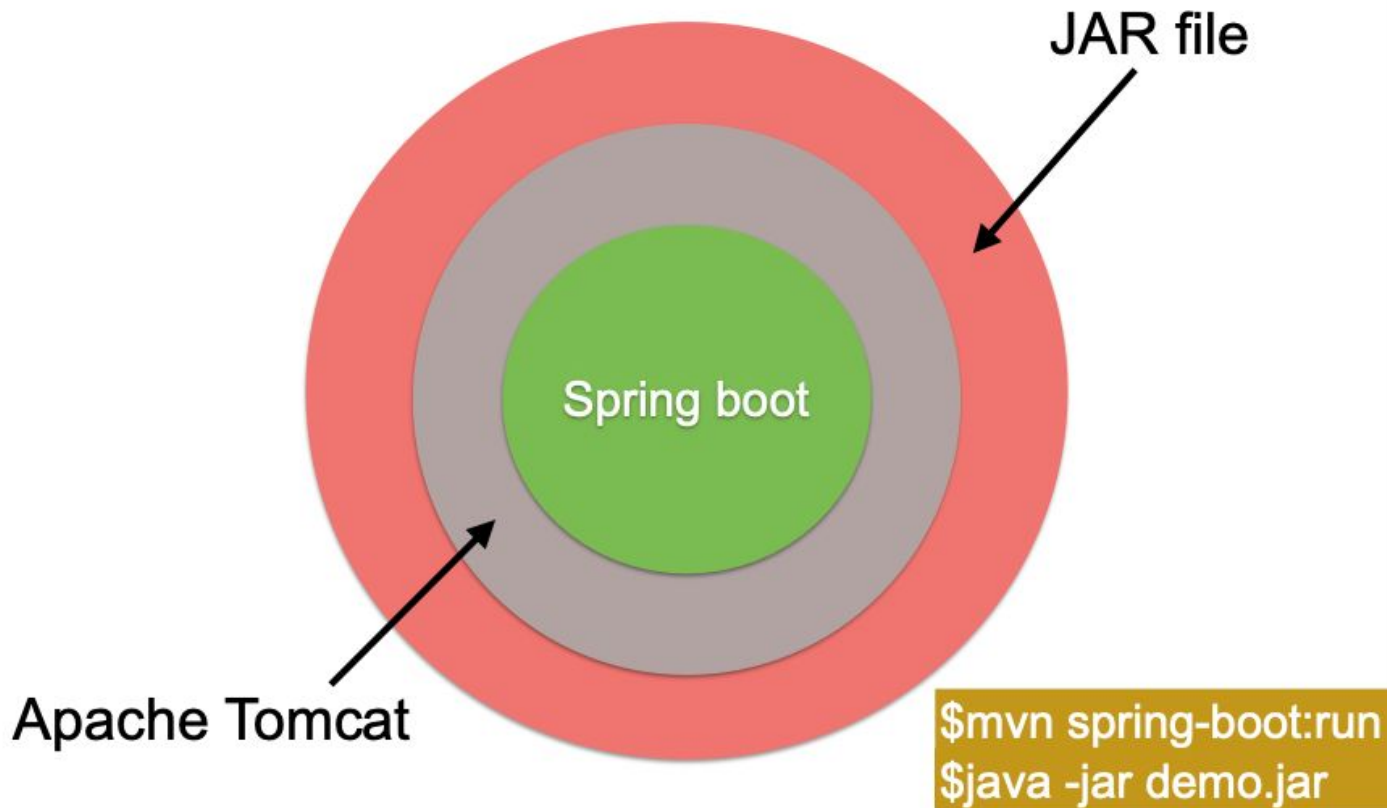
Production tools (monitoring, health check)

Configuration management

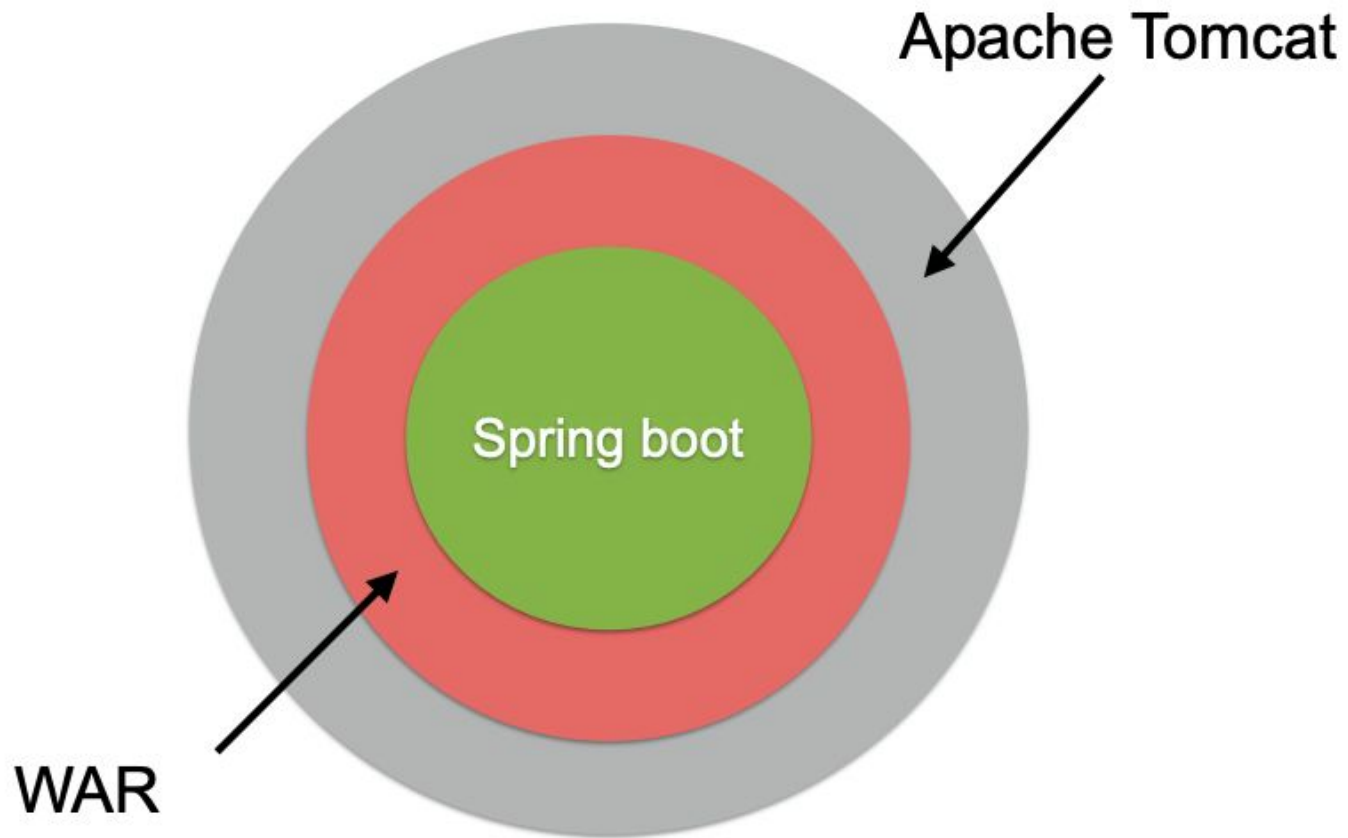
Dev tools

No source code generation, no XML

How ?

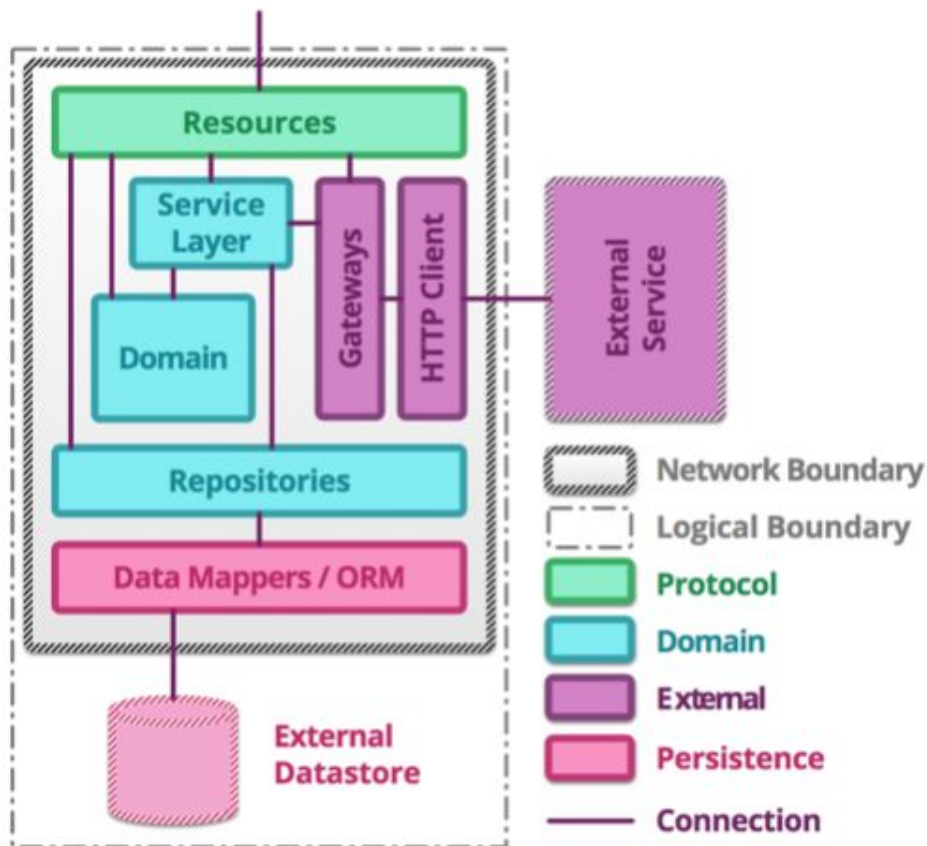


How ?

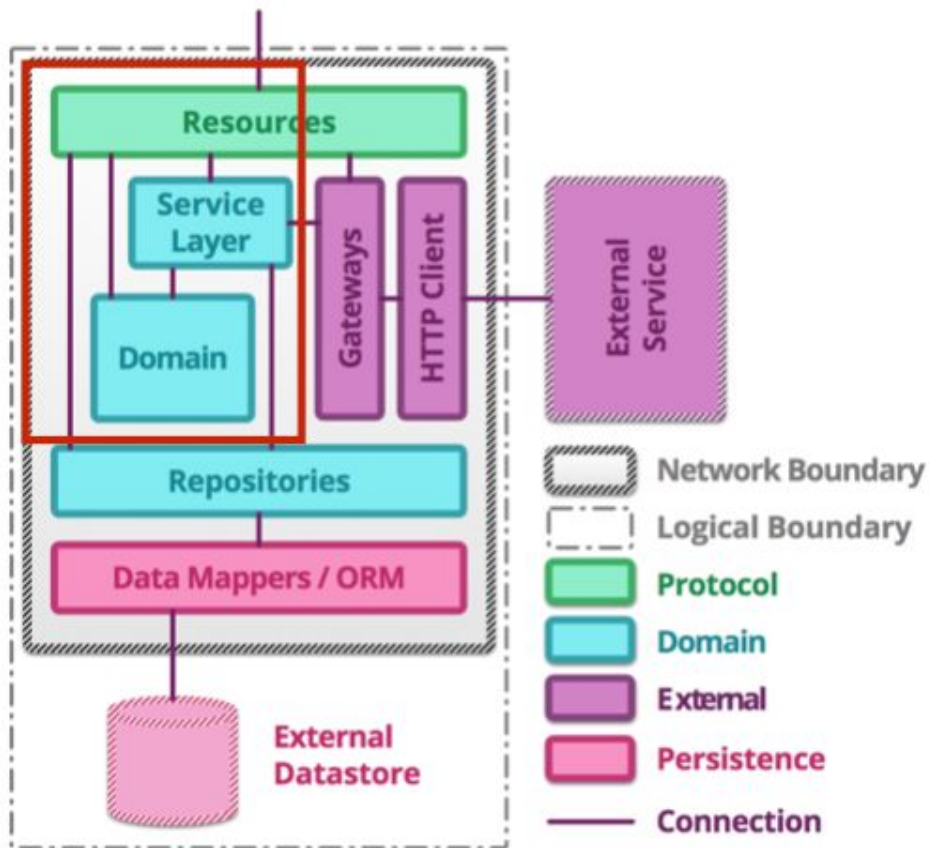


Building RESTful API with Spring Boot

Service Structure



Service Structure



Create project with Spring Initializr

Spring Initializr

<https://start.spring.io/>

SPRING INITIALIZR bootstrap your application now

Generate a with and Spring Boot

Project Metadata

Artifact coordinates

Group

Artifact

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Selected Dependencies

[Generate Project](#)

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

start.spring.io is powered by [Spring Initializr](#) and [Pivotal Web Services](#)

Structure of Spring Boot



Spring Boot main class

```
package com.example.demo;


import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

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

Run project (Dev mode)

`$/mvnw spring-boot:run`

The logo for Spring Boot, featuring a stylized horse head made of ASCII characters. Below the logo, the text "Spring Boot" is displayed in green, followed by "(v2.0.2.RELEASE)" in black.

`:: Spring Boot ::` (v2.0.2.RELEASE)

```
2018-06-07 13:03:30.412 INFO 12828 --- [
oApplication      : Starting DemoApplication on
D 12828 (started by somkiat in /Users/somkiat/Down
2018-06-07 13:03:30.418 INFO 12828 --- [
oApplication      : No active profile set, fall
```


Run project (production mode)

\$.mvnw package

```
$java -jar target/<file name>.jar
```

[illegible]

```
2018-06-07 13:03:30.412 INFO 12828 --- [
oApplication          : Starting DemoApplication on
D 12828 (started by somkiat in /Users/somkiat/Down
2018-06-07 13:03:30.418 INFO 12828 --- [
oApplication          : No active profile set, fall
```

Display all beans !!

```
@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        ApplicationContext context
            = SpringApplication.run(DemoApplication.class,
                args);

        for (String name: context.getBeanDefinitionNames()) {
            System.out.println(name);
        }
    }
}
```

Custom Banner

Custom banner

```

      .
    /\  /____'  _ _ _ _ _ ( _ ) _ _ _ _ _ \ \ \ \ \
  ( ( ) \____|  _ _ _ _ _ \ \ \ \ \ \ \ \ \ \ \ \ \ \
    \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
      '  |____|  _ _ _ _ _ \ \ \ \ \ \ \ \ \ \ \ \ \ \
=====|_|=====|_|/=/=/=/=/=/
:: Spring Boot ::                (v2.0.2.RELEASE)

```

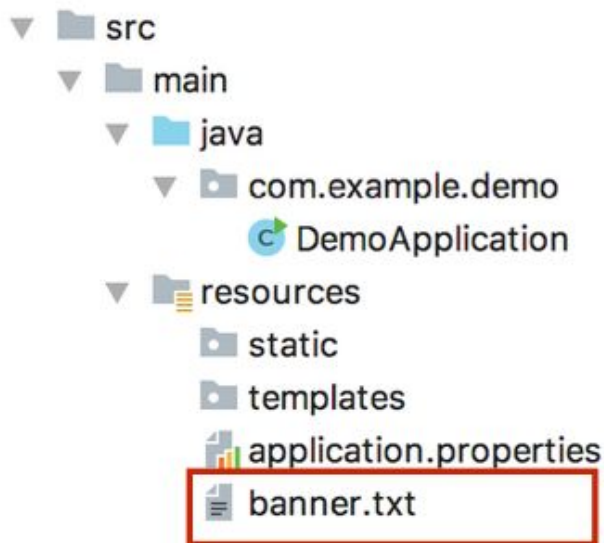
```

2018-06-07 13:03:30.412 INFO 12828 --- [
oApplication           : Starting DemoApplication on
D 12828 (started by somkiat in /Users/somkiat/Down
2018-06-07 13:03:30.418 INFO 12828 --- [
oApplication           : No active profile set, fall

```

Custom banner (1)

Create file banner.txt or banner.png in resources folder



Custom banner (2)

```
@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        ImageBanner banner = new ImageBanner(
            new ClassPathResource("try.png"));

        new SpringApplicationBuilder()
            .sources(DemoApplication.class)
            .banner(banner)
            .run();
    }
}
```

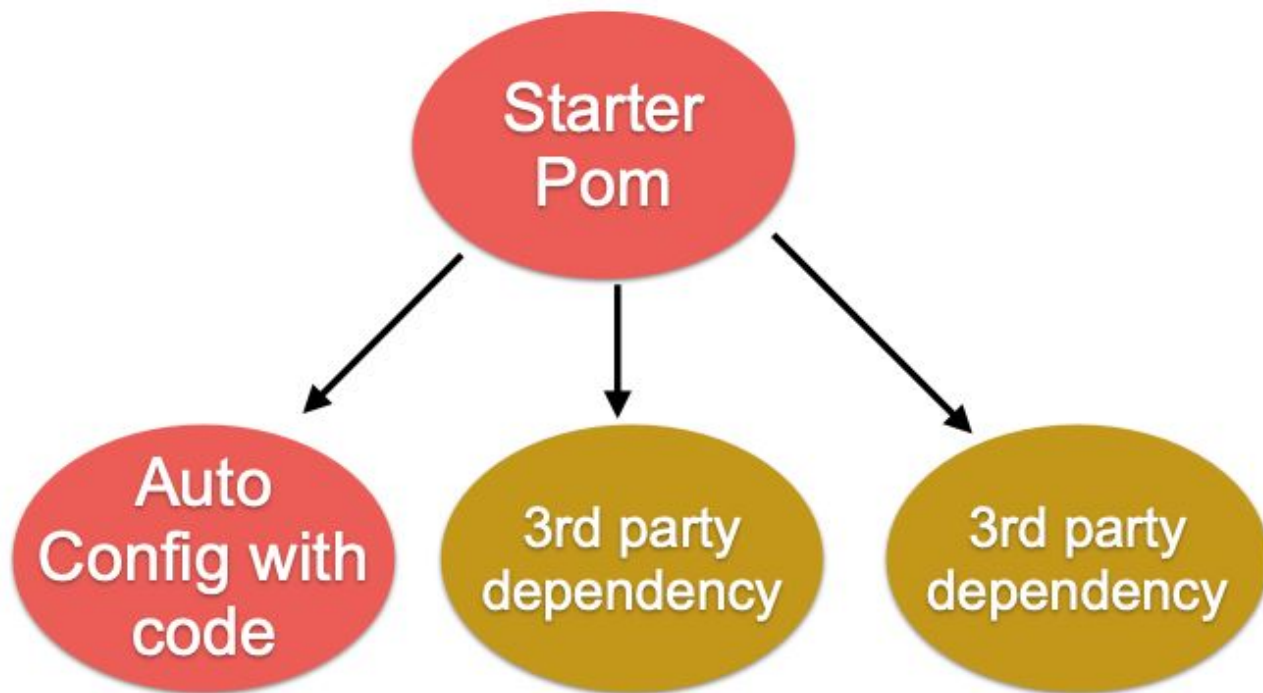
Disable banner

```
@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        new SpringApplicationBuilder()
            .sources(DemoApplication.class)
            .bannerMode(Banner.Mode.OFF)
            .run(args);
    }
}
```

Anatomy of Starter

Anatomy of Starter



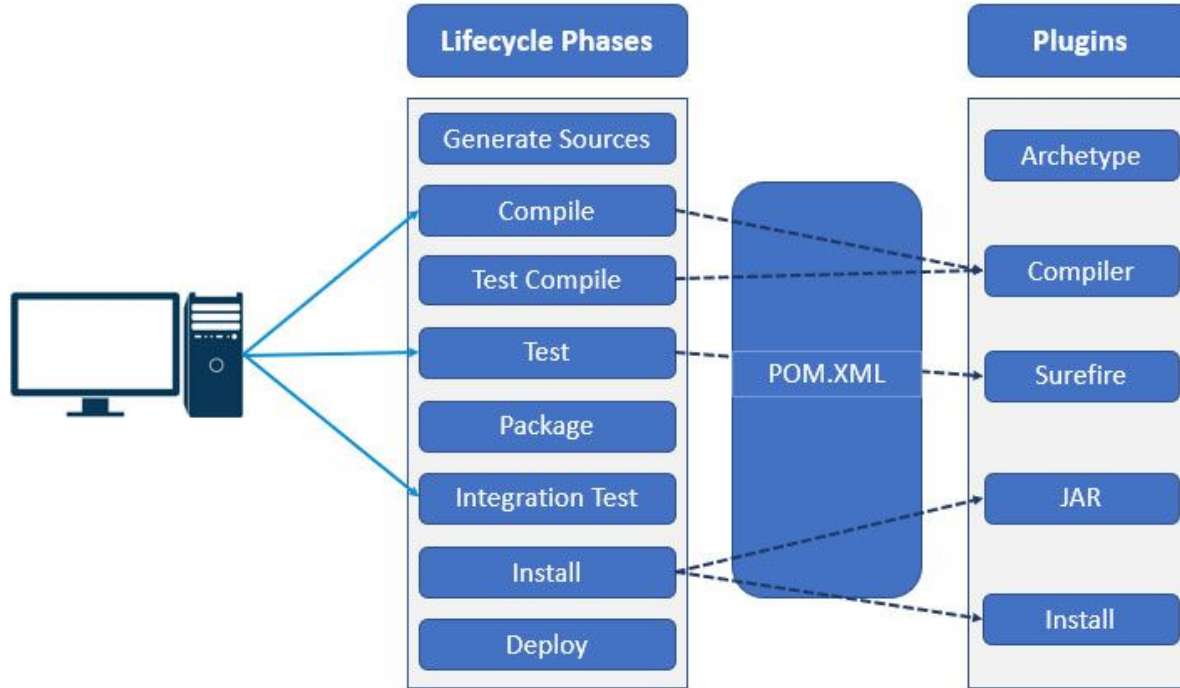
Configuration management

Properties/XML/YAML

Config server (App, Spring cloud config, git)

17 ways !!!

Maven



Look at POM.xml

POM.xml (1)

```
<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.c
  <modelVersion>4.0.0</modelVersion>

  <groupId>hello</groupId>
  <artifactId>hello</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.0.0.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>
```

POM.xml (2)

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

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

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

(D)ependency (I)njection

exercise

Break 5 minutes

10:05

REST

REST

REpresentation **S**tate **T**ransfer

The style of software architecture behind
RESTful services

Defined in 2000 by Roy Fielding

Goals

Scalability

Generality of interfaces

Independent deployment of components

RESTful service

REST Request Messages

RESTful request is typically in form of
Uniform Resource Identifiers (URI)

REST Request Messages

RESTful request is typically in form of
Uniform Resource Identifiers (URI)

Structure of URI depend on specific service

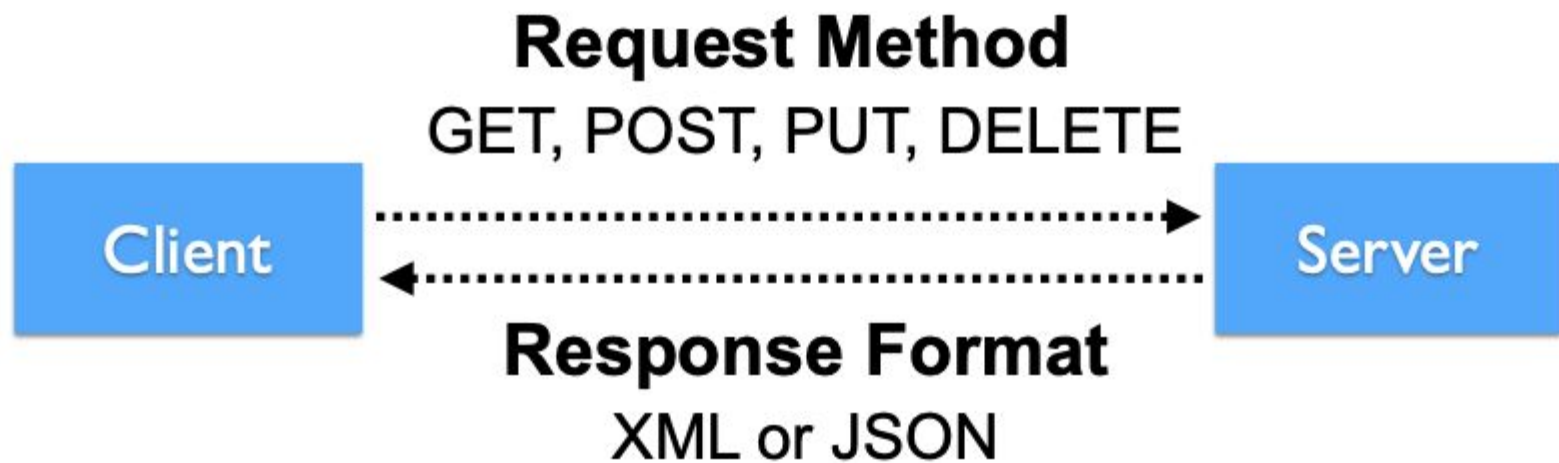
REST Request Messages

RESTful request is typically in form of
Uniform Resource Identifiers (URI)

Structure of URI depend on specific service

Request can include parameter and data in
body of request as XML, JSON etc.

REST Request & Response

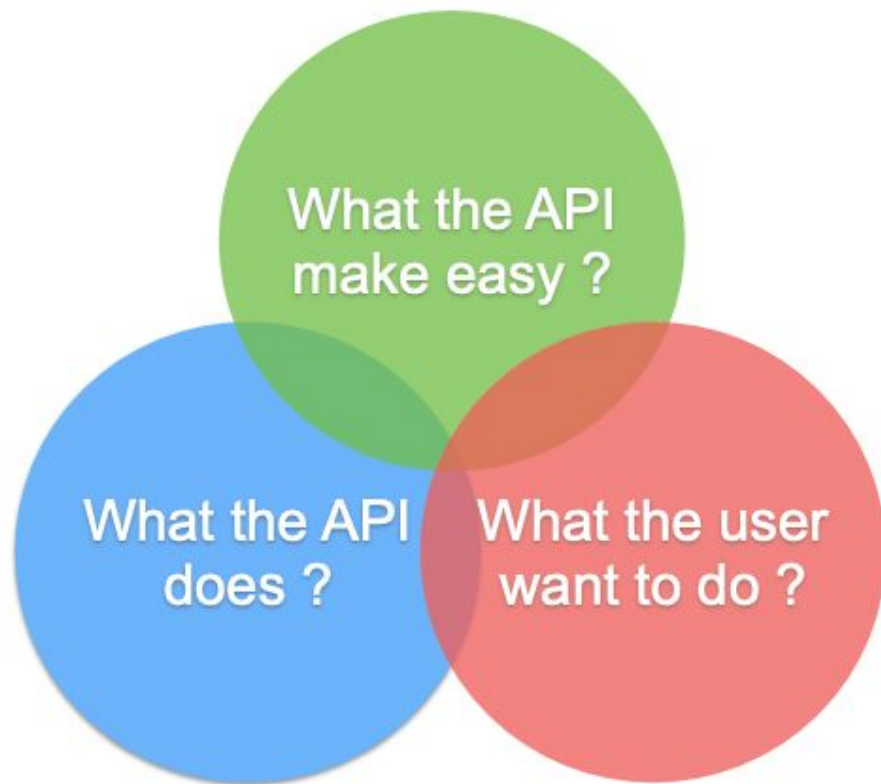


HTTP Methods meaning

Method	Meaning
GET	Read data
POST	Create/Insert new data
PUT/PATCH	Update data or insert if a new id
DELETE	Delete data

Response format ?

Good APIs ?



Java Framework for RESTful


undertow

 JAX-WS

 spring
by Pivotal.


Apache CXF

unirest
Lightweight HTTP Request Client Libraries

 Jersey
RESTful Web Services in Java.

 Ratpack
A toolkit for JVM web applications



Spark

 STRUTS

 play



Dropwizard

WildFly  SWARM

 REST
Easy

Restlet

 akka

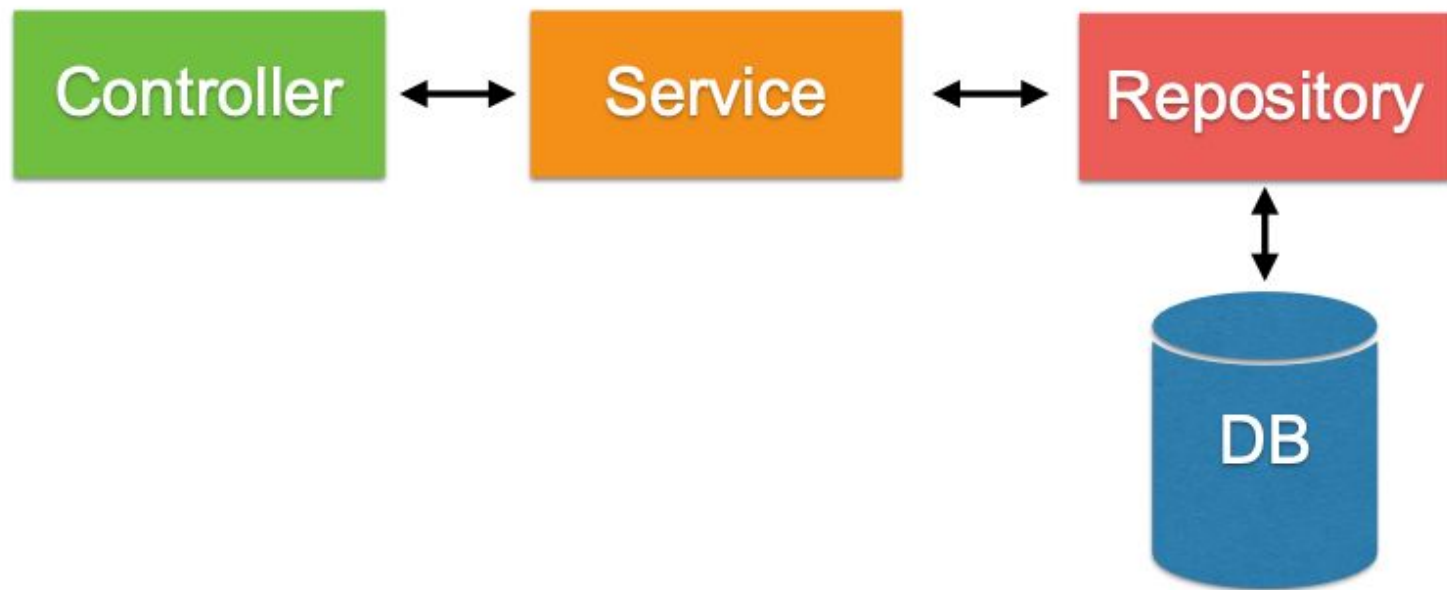
 MIAJA
web framework



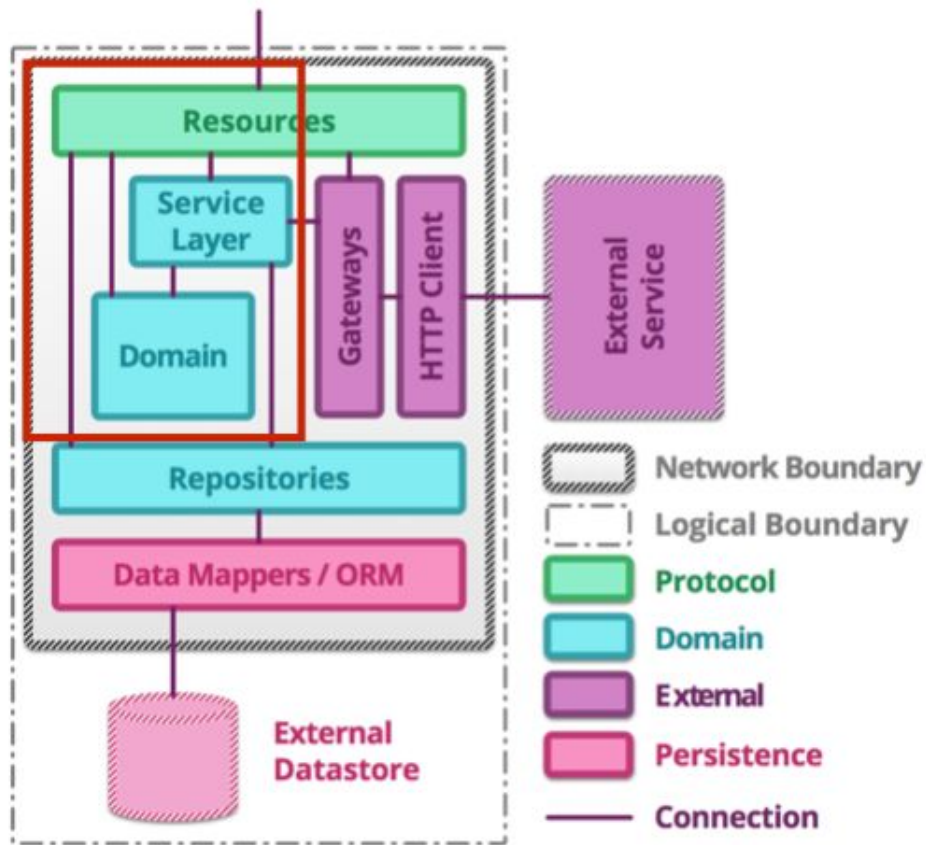
ACT.framework

Basic structure of Spring Boot

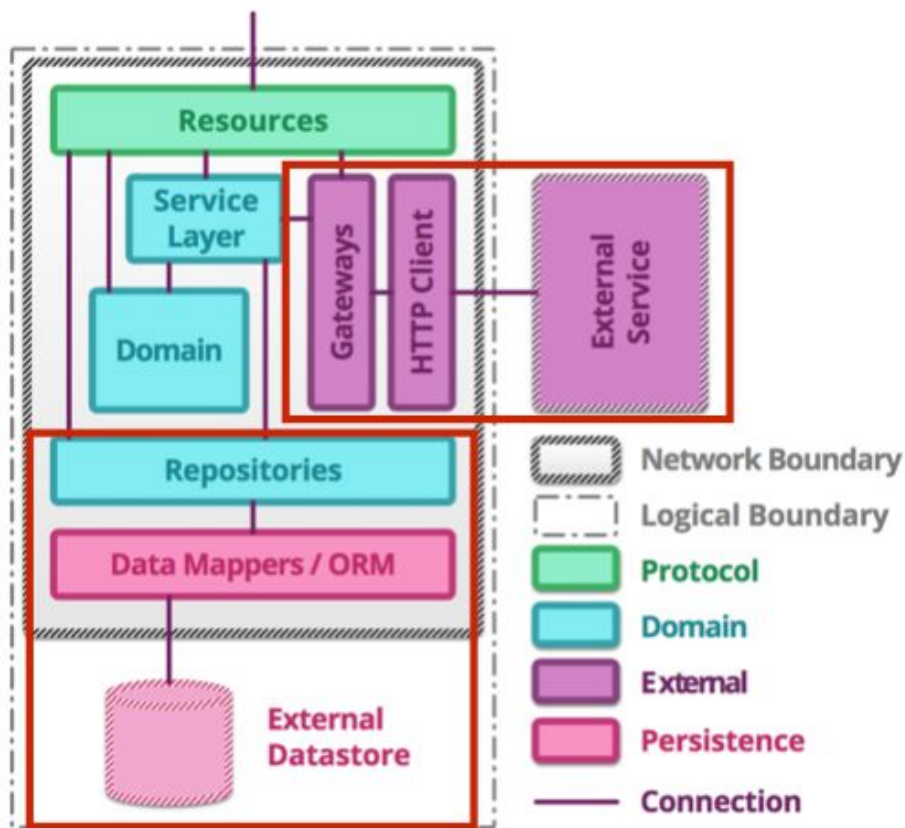
Basic structure of Spring Boot



Service Structure



Service Structure



Controller

Request and Response

Validation input

Delegate to others classes such as service and repository

Service

Control the flow of main business logics

Manage database transaction

Don't reuse service

Repository

Manage data in data store such as RDBMS and
NoSQL

Gateway

Call external service via network such as
WebServices and RESTful APIs

Spring Boot Structure (1)

Separate by function/domain/feature

```
feature1
```

- controller
- service
- repository

```
feature2
```

- controller
- service
- repositor

Spring Boot Structure (2)

Separate by function/domain/feature

