

SpringOne Platform by Pivotal.

It's a Kind of Magic Under the Covers of Spring Boot

Stéphane Nicoll
@snicoll

Brian Clozel
@bclozel



Basic Spring Boot application

```
package com.example.hello;

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

@SpringBootApplication
public class HelloApp {

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

}
```

@SpringBootApplication

@SpringBootApplication

@SpringBootConfiguration
@ComponentScan
@EnableAutoConfiguration

@ComponentScan

@ComponentScan



@ComponentScan

com.example.hello



com.example.hello.alpha



com.example.hello.bravo



com.example.charlie





@ComponentScan

com.example.hello



com.example.hello.alpha



com.example.hello.bravo



com.example.hello.charlie







@Component



@Repository

@Configuration

@Component

@Controller

@Service

@RestController



@Configuration and @ComponentScan in action

```
@SpringBootApplication
public class HelloApp {

    @Bean
    public HelloService helloService() {
        return new ConsoleHelloService("Howdy", "!");
    }

    //...

}
```

```
@Component
public class HelloCommandLineRunner implements CommandLineRunner {

    // ...

}
```

@Component

@Bean

@Component

@Bean

@Service

@Configuration

@Controller

@Repository

@RestController





User configuration

@ComponentScan



User configuration

@ComponentScan



@EnableAutoConfiguration



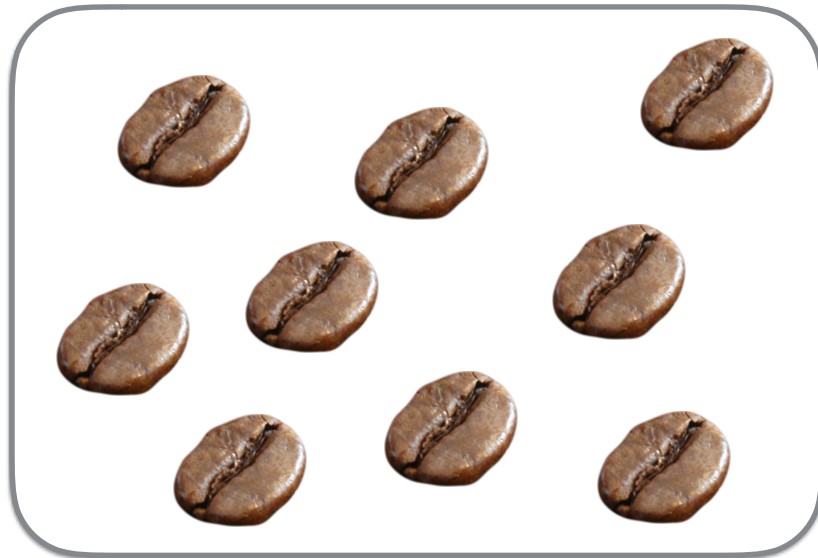
User configuration

@ComponentScan



Auto configuration

@EnableAutoConfiguration



@EnableAutoConfiguration



Basic auto-configuration



META-INF/spring.factories

```
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\nhello.autoconfigure.HelloAutoConfiguration
```

```
package hello.autoconfigure;

import hello.ConsoleHelloService;
import hello.HelloService;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class HelloAutoConfiguration {

    @Bean
    public HelloService helloService() {
        return new ConsoleHelloService();
    }

}
```



Better auto-configuration

```
package hello.autoconfigure;

import hello.ConsoleHelloService;
import hello.HelloService;

import org.springframework.boot.autoconfigure.condition.ConditionalOnClass;
import org.springframework.boot.autoconfigure.condition.ConditionalOnMissingBean;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
@ConditionalOnClass(HelloService.class)
public class HelloAutoConfiguration {

    @Bean
    @ConditionalOnMissingBean
    public HelloService helloService() {
        return new ConsoleHelloService();
    }
}
```

@ConditionalOnClass

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass
@ConditionalOnMissingClass

@ConditionalOnBean
@ConditionalOnMissingBean

@ConditionalOnClass
@ConditionalOnMissingClass

@ConditionalOnBean
@ConditionalOnMissingBean

@ConditionalOnProperty

@ConditionalOnWebApplication

@ConditionalOnNotWebApplication

@ConditionalOnEnvironment

@ConditionalOnProperty

@ConditionalOnJndi

@ConditionalOnClass

@ConditionalOnMissingClass

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnSingletonBean

@ConditionalOnNotSingletonBean

@ConditionalOnExpression

@ConditionalOnJavaVersion

@ConditionalOnOS

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnMissingBeanClass

@ConditionalOnMissingBeanType

@ConditionalOnMissingBeanName

@ConditionalOnMissingBeanMethod

@ConditionalOnMissingBeanField

@ConditionalOnMissingBeanAnnotation

@ConditionalOnMissingBeanInterface

@ConditionalOnMissingBeanPackage

@ConditionalOnProperty

@ConditionalOnWebApplication

@ConditionalOnNotWebApplication

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnWebApplication

@ConditionalOnNotWebApplication

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnWebApplication

@ConditionalOnJava

@ConditionalOnNotWebApplication

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnWebApplication
@ConditionalOnNotWebApplication

@ConditionalOnJava

@ConditionalOnJndi

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean
@ConditionalOnMissingBean

@ConditionalOnClass
@ConditionalOnMissingClass

@ConditionalOnProperty

@ConditionalOnWebApplication
@ConditionalOnNotWebApplication

@ConditionalOnJava

@ConditionalOnJndi

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

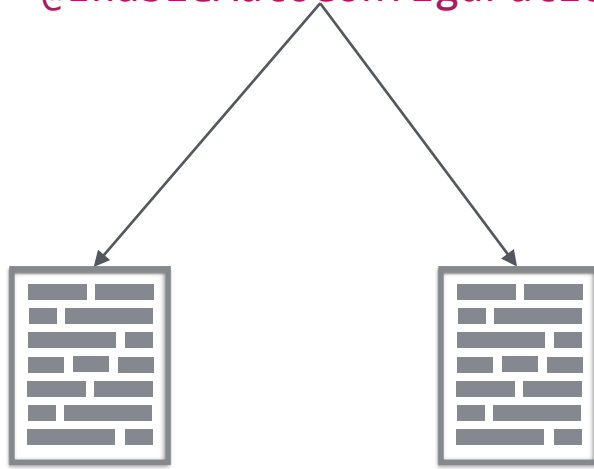
@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnExpression

@EnableAutoConfiguration

@EnableAutoConfiguration



@EnableAutoConfiguration

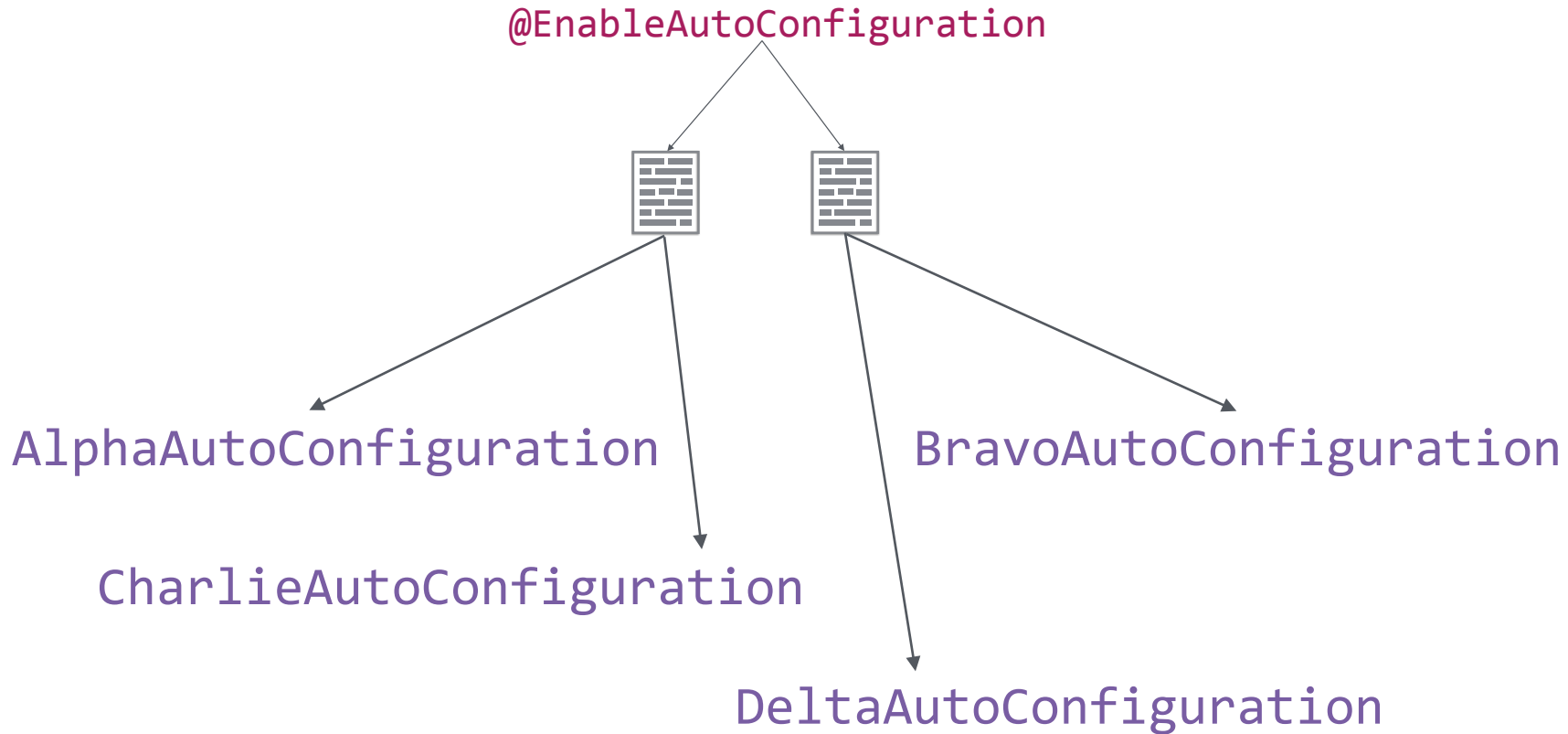


AlphaAutoConfiguration

BravoAutoConfiguration

CharlieAutoConfiguration

DeltaAutoConfiguration



@Configuration
AlphaAutoConfiguration

@ConditionalOn... ✓
@ConditionalOn... ✓
@Configuration
CharlieAutoConfiguration

@ConditionalOn... ✗
@Configuration
BravoAutoConfiguration

@ConditionalOn... ✗
@ConditionalOn...
@Configuration
DeltaAutoConfiguration

@Configuration
AlphaAutoConfiguration

@ConditionalOn...
@ConditionalOn...
@Configuration
CharlieAutoConfiguration

@ConditionalOn...

@ConditionalOn...

@Configuration

CharlieAutoConfiguration

@ConditionalOn...

@ConditionalOn...

@Configuration

class CharlieAutoConfiguration {

@ConditionalOn... ✓

@Bean

public Foo foo() {
 return new Foo();
 }

}

@ConditionalOnWebApplication

@ConditionalOnJava

@ConditionalOnNotWebApplication

@ConditionalOnJndi

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnExpression

@ConditionalOnWebApplication
@ConditionalOnNotWebApplication

@ConditionalOnJava

@ConditionalOnJndi

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnExpression

@ConditionalOnWebApplication

@ConditionalOnJava

@ConditionalOnNotWebApplication

@ConditionalOnJndi

@ConditionalOnSingleCandidate

@ConditionalOnResource

@ConditionalOnBean

@ConditionalOnMissingBean

@ConditionalOnClass

@ConditionalOnProperty

@ConditionalOnMissingClass

@ConditionalOnExpression

@ConditionalOnClass

@ConditionalOnMissingClass

@ConditionalOnResource

@ConditionalOnJava

@ConditionalOnWebApplication

@ConditionalOnNotWebApplication

@ConditionalOnJndi

@ConditionalOnProperty

@ConditionalOnExpression

@ConditionalOnSingleCandidate

@ConditionalOnBean

@ConditionalOnMissingBean



Custom Conditions





Conditions:
traps to avoid

```
@Configuration
```

```
class DataAccessAutoConfiguration {
```

```
    @Bean
```

```
    @ConditionalOnSingleCandidate(JdbcTemplate.class)
```

```
    public DataAccess dataAccess(JdbcTemplate jdbc) {
```

```
        return new DataAccess(jdbc);
```

```
    }
```

```
}
```

@Configuration

class DataSourceAutoConfiguration {

 @Bean

 public DataSource dataSource() {

 return new DataSource();

 }

}

```
@Configuration
class JdbcAutoConfiguration {

    @Bean
    @ConditionalOnSingleCandidate(dataSource.class)
    public JdbcTemplate jdbcTemplate(dataSource ds) {
        return new JdbcTemplate(ds);
    }
}
```

DataAccessAutoConfiguration

DataSourceAutoConfiguration

JdbcAutoConfiguration


```
@AutoConfigureAfter(  
    JdbcAutoConfiguration.class)
```

```
    DataAccessAutoConfiguration
```

```
    DataSourceAutoConfiguration
```

```
    JdbcAutoConfiguration
```

```
@AutoConfigureAfter(  
    JdbcAutoConfiguration.class)
```

```
    DataAccessAutoConfiguration
```

```
@AutoConfigureBefore(  
    JdbcAutoConfiguration.class)
```

```
    DataSourceAutoConfiguration
```

```
    JdbcAutoConfiguration
```

DataSourceAutoConfiguration

JdbcAutoConfiguration

DataSourceAutoConfiguration

DataSourceAutoConfiguration



DataSource

JdbcAutoConfiguration

DataSourceAutoConfiguration

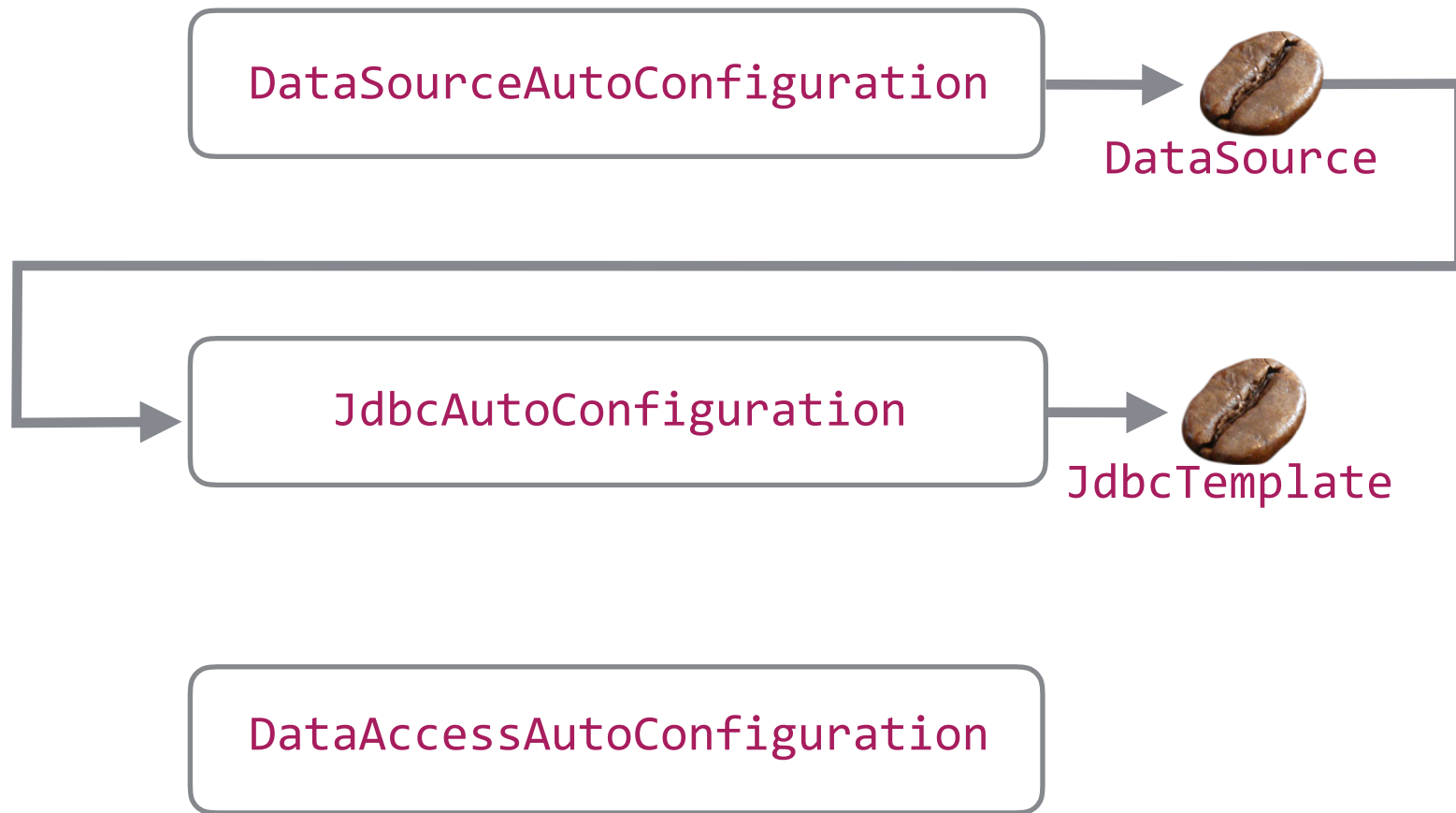
DataSourceAutoConfiguration

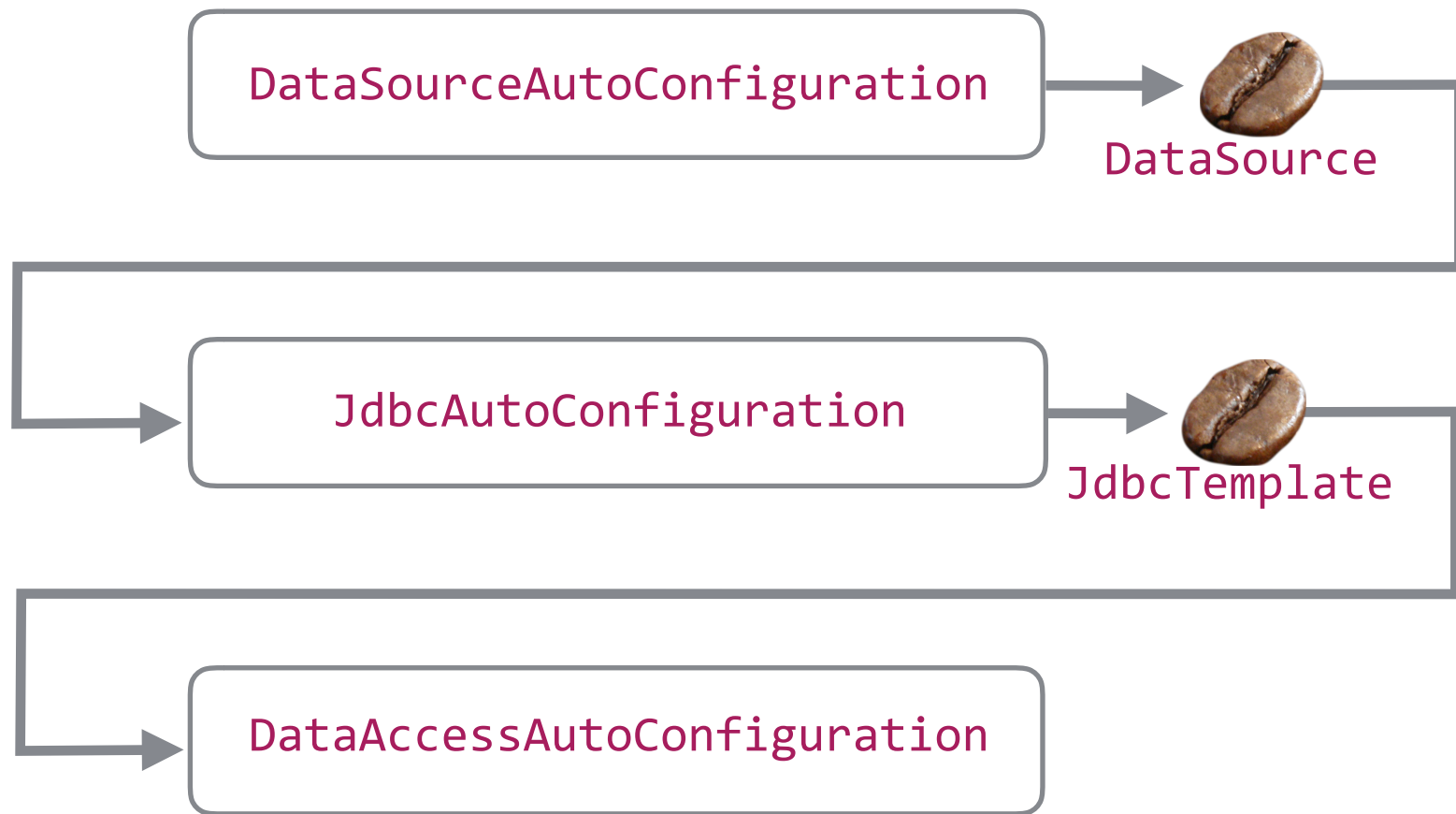


DataSource

JdbcAutoConfiguration

DataSourceAutoConfiguration





DataSourceAutoConfiguration



DataSource

JdbcAutoConfiguration



JdbcTemplate

DataSourceAutoConfiguration



DataSource


```
@Configuration
class GsonAutoConfiguration {

    @ConditionalOnClass(Gson.class)
    @Bean
    public Gson gson() {
        return new Gson();
    }
}
```

```
@Configuration
class GsonAutoConfiguration {

    @ConditionalOnClass(Gson.class)
    @Bean
    public Gson gson() {
        return new Gson();
    }

}
```

```
@ConditionalOnClass(Gson.class)
@Configuration
class GsonAutoConfiguration {

    @Bean
    public Gson gson() {
        return new Gson();
    }

}
```

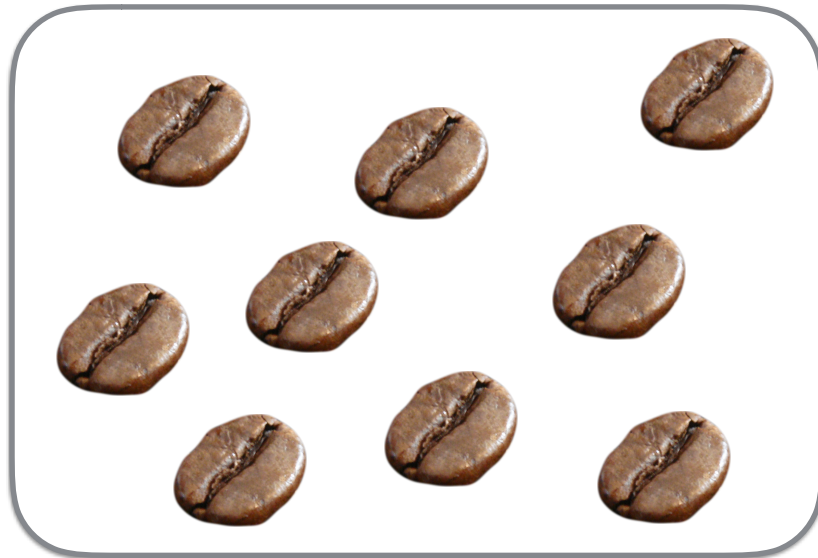
User configuration

@ComponentScan



Auto configuration

@EnableAutoConfiguration



@Configuration

public class AppConfiguration {

 @Bean

 @ConditionalOnMissingBean

 public CustomBean customBean() {

 return new CustomBean();

 }

}

@Configuration

public class AppConfiguration {

 @Bean

 @Conditional~~On~~MissingBean

 public CustomBean customBean() {

 return new CustomBean();

 }

}



Customizing the environment



Spring (Boot) events

Run the application!



ApplicationStartingEvent

Logging initialization begins

ApplicationEnvironment
PreparedEvent

Config files are read

Environment post processors
are called

Logging initialization completes

ApplicationPreparedEvent

Logging initialization completes



ApplicationPreparedEvent

Application context is refreshed



ContextRefreshedEvent

Embedded servlet container
connectors are started

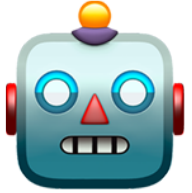


EmbeddedServletContainer
InitializedEvent



ApplicationReadyEvent

CommandLineRunners and
ApplicationRunners are called



Failure Analyzer

FailureAnalyzer

NoSuchBeanDefinition

Bind

BeanCurrentlyInCreation

FailureAnalyzer

NoUniqueBeanDefinition

PortInUse

DataSourceBeanCreation

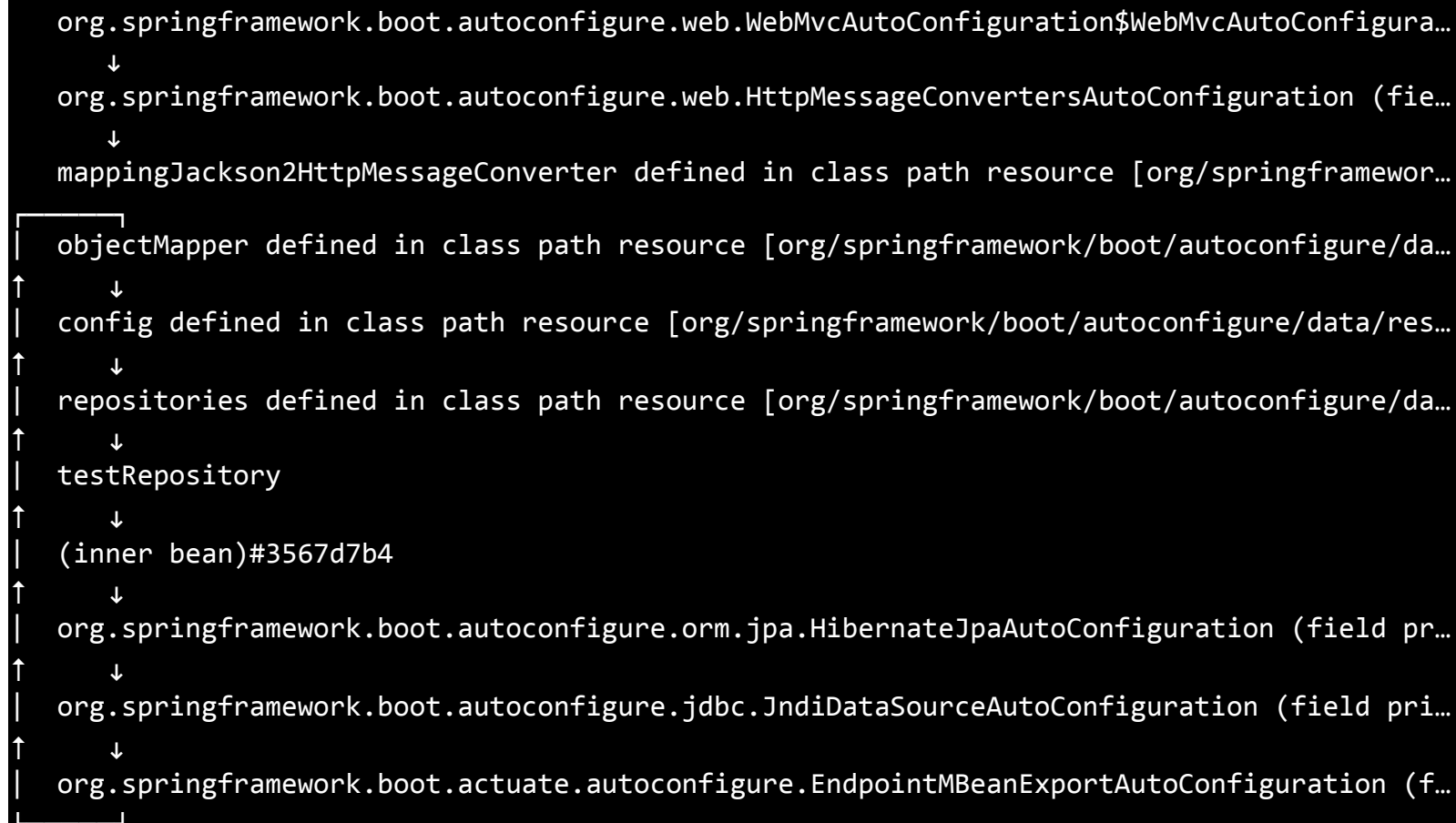

```
at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDepe...  
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDepend...  
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$...  
... 226 common frames omitted
```

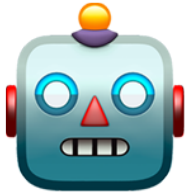
```
Caused by: org.springframework.beans.factory.BeanCreationException: Could not autowire fi...  
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$...  
at org.springframework.beans.factory.annotation.InjectionMetadata.inject(InjectionMet...  
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor....  
... 246 common frames omitted
```

```
Caused by: org.springframework.beans.factory.BeanCurrentlyInCreationException: Error crea...  
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.beforeSingl...  
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleto...  
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBe...  
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBean...  
at org.springframework.beans.factory.support.DefaultListableBeanFactory.findAutowireC...  
at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDepe...  
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDepend...  
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$...  
... 248 common frames omitted
```

340 lines

The dependencies of some of the beans in the application context form a cycle:





Custom Failure Analyzer

Spring Boot Startup Performance

“My app takes a lot of time to start”

“My app takes a lot of time to start”

“Classpath scanning is slow”

“My app takes a lot of time to start”

“Classpath scanning is slow”



Question:

A Spring Boot application (with 200 beans) takes 6.5 seconds to start.

How long does classpath scanning take?

Question:

A Spring Boot application (with 200 beans) takes 6.5 seconds to start.

How long does classpath scanning take?

a) 1.5 sec

b) 1 sec

c) 500 ms

d) 100 ms

Question:

A Spring Boot application (with 200 beans) takes 6.5 seconds to start.

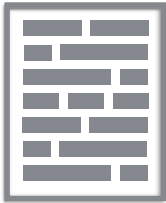
How long does classpath scanning take?



d) 100 ms

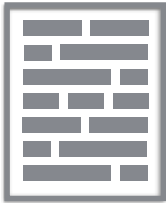
```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-context-indexer</artifactId>  
  <optional>true</optional>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-context-indexer</artifactId>  
  <optional>true</optional>  
</dependency>
```



META-INF/spring.components

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-context-indexer</artifactId>  
  <optional>true</optional>  
</dependency>
```



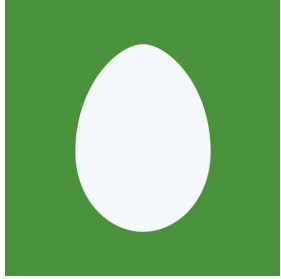
META-INF/spring.components

com.example.FooService=org.springframework.stereotype.Component
...

“My app **still takes a lot of time to start”**

“My app **still takes a lot of time to start”**



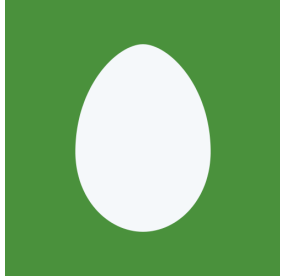


@SpringDeveloper

Strange behavior on macOS:

Spring Boot app taking up to 13 secs to start.

Run iTunes. App starts in 1-2 secs.



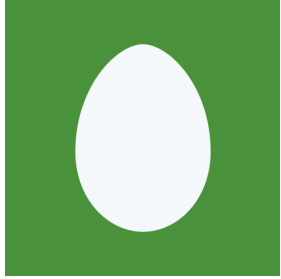
@SpringDeveloper

Strange behavior on macOS:

Spring Boot app taking up to 13 secs to start.

Run iTunes. App starts in 1-2 secs.

a) iTunes is great



@SpringDeveloper

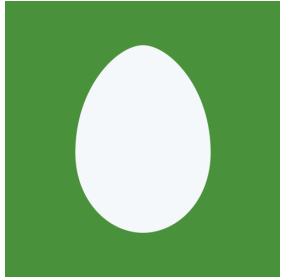
Strange behavior on macOS:

Spring Boot app taking up to 13 secs to start.

Run iTunes. App starts in 1-2 secs.

a) iTunes is great

b) **@StartFasterWithITunes**



@SpringDeveloper

Strange behavior on macOS:

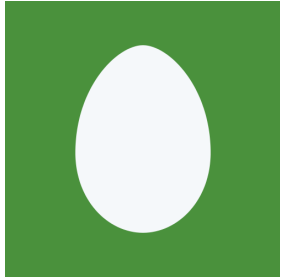
Spring Boot app taking up to 13 secs to start.

Run iTunes. App starts in 1-2 secs.

a) iTunes is great

b) **@StartFasterWithITunes**

c) localhost DNS resolution in macOS Sierra



@SpringDeveloper

Strange behavior on macOS:

Spring Boot app taking up to 13 secs to start.

Run iTunes. App starts in 1-2 secs.

a) iTunes is great

b) @StartFasterWithITunes

c) localhost DNS resolution in macOS Sierra

Startup performance: (not so) usual suspects

- Network configuration
- Antivirus
- Lack of entropy, use "urandom" (fixed in SB 1.4, see #6229)
- Static code initialization
- Use the JVM "noverify" flag
- See: <https://github.com/dsyer/spring-boot-startup-bench>

Thank you!

Stéphane Nicoll
@snicoll

Brian Clozel
@bclozel

Thank you!

<https://github.com/snicoll-demos/hello-service-auto-configuration>

Stéphane Nicoll
@snicoll

Brian Clozel
@bclozel