

# Spring Boot

The Ripper




 @tolkv

 @lavcraft



@jekaborisov 

@jeka1978 



# За что мы не любим Spring Boot 2.0 в РФ

```
private static boolean isAlpha(char ch) {
```

```
    return ch >= 'a' && ch <= 'z';
```

```
}
```

```
..
```

```
@ConfigurationProperties
```

# За что мы не любим Spring Boot 2.0 в РФ

```
private static boolean isAlpha(char ch) {  
    return ch >= 'a' && ch <= 'z';  
}
```

..

@ConfigurationProperties

как:

пройти:

в-библиотеку: не знаю

Пролетарии всех стран, соединяйтесь!

О Р Г А Н  
Мурманского  
Обкома и Горкома ВКП(б)  
и Оргкомитета Президиума  
Верховного Совета РСФСР  
по Мурманской области

5 ЯНВАРЯ  
1939 г.  
ЧЕТВЕРГ  
№ 4 (1073)  
За выпуск 2700. Цена 10 коп.

# ПОЛЯРНАЯ ПРАВДА

## Присяга родине

Сегодня на Мурмане улане...  
Присяга...  
Верховного Совета СССР...

и нет ему равных. И вот в...  
граждан СССР были...  
в Красноярске, 904 дня...  
наша самоотдача, жертвенность...

Во всех своих выступлениях...  
те чувства возмущены...  
забыл за то одну минуту...  
Богатейшие материальные...

Присягну ласкати прарики...  
содержанию обидки...  
своему благополучию...  
Востаня прарики — это...

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советскими...  
Самод Советский Социалисти-...  
присягу своему народу, своей...  
Родины, утвердить сле-...  
дующий текст военной присяги...  
для Рабоче-Крестьянской Кр-...  
А. Завести своем до присягу ро-

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Утвердить...  
...и...  
...и...  
...и...  
...и...  
...и...

# ЗАБАСТОВКА БИНОВ ПОДАВЛЕНА, СИНГЛТОНЫ НЕ ВЫЙДУТ ИЗ КОНТЕЙНЕРА ДО КОНЦА СРОКА



# ПОЛЯРНАЯ ПРАВДА

ОРГАН  
Мурманского  
Обкома и Горкома ВКП(б)  
и Оргкомитета Президиума  
Верховного Совета РСФСР  
по Мурманской области

5 ЯНВАРЯ  
1938 г.  
ЧЕТВЕРГ  
№ 4 (1073)  
За 1938 год. Цена 10 коп.

## Присяга родине

Слова на торжественном приеме Президиума Верховного Совета СССР в честь военной присяги Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота, а также ряд о речи заместителя и заместительницы председателя Президиума Верховного Совета СССР в честь военной присяги Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота...

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советской Конституцией СССР, утверждая сформулированный текст военной присяги для Рабоче-Крестьянской Красной Армии...  
«Я, гражданин Союза Советских Социалистических Республик, вступаю в ряды Рабоче-Крестьянской Красной Армии...

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Генерал-полковник тов. М. М. Калинин, заместитель Председателя Президиума Верховного Совета СССР, выступил на торжественном приеме в честь военной присяги Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота...

# ЛЕНИВЫЕ СИНГЛТОНЫ МОГУТ НЕ СОЗДАВАТЬСЯ С ВЕРСИИ 4.3

В честь торжественного приема Президиума Верховного Совета СССР в честь военной присяги Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота, а также ряд о речи заместителя и заместительницы председателя Президиума Верховного Совета СССР в честь военной присяги Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота...

Президиум Президиума Верховного Совета СССР  
М. КАЛНИН  
Секретарь Президиума Верховного Совета СССР  
А. ГОРКИН  
Москва, Кремль.  
5 января 1938 года.

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянского Военно-

В соответствии с Советской Конституцией СССР, утверждая сформулированный текст военной присяги для Рабоче-Крестьянской Красной Армии...  
«Я, гражданин Союза Советских Социалистических Республик, вступаю в ряды Рабоче-Крестьянской Красной Армии...



...и торжественно принял присягу...  
...и торжественно принял присягу...  
...и торжественно принял присягу...



5 ЯНВАРЯ  
1939 г.  
ЧЕТВЕРГ  
№ 4 (1073)

# ПОЛЯРНАЯ ПРАВДА

ОРГАН  
Мурманского  
Обкома и Горкома ВКП(б)  
и Оргкомитета Президиума  
Верховного Совета РСФСР  
по Мурманской области

## Присяга родине

Слова на торжественном заседании Президиума Верховного Совета СССР в торжественной обстановке присягнули Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянскому Красному Флоту, а также ряд офицеров и командиров в партизанских отрядах, работавших по защите нашей Родины в тылу, партизаны и командиры Рабоче-Крестьянской Красной Армии, Рабоче-Крестьянского Красного Флота и бойцы партизанской борьбы.

В этот день знамя нашей Родины — торжественно передано в руки Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота.

Об этом своем величайшем долге солдаты и командиры Рабоче-Крестьянской Красной Армии и Рабоче-Крестьянского Красного Флота и бойцы партизанской борьбы.

Наша Родина в опасности! Мы должны быть готовы к тому, чтобы в любой момент выступить на защиту нашей Родины.

Мы должны быть готовы к тому, чтобы в любой момент выступить на защиту нашей Родины.

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советскими Конституцией и Уставом, утвержденный текст военной присяги для Рабоче-Крестьянской Красной Армии.

«Я, гражданин Союза Советских Социалистических Республик, вступаю в ряды Рабоче-Крестьянской Красной Армии, обязуюсь

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Готовить и принимать военную присягу на торжественном заседании Президиума Верховного Совета СССР.

1. Каждый военнослужащий при поступлении в ряды Рабоче-Крестьянской Красной Армии, Рабоче-Крестьянского Военно-Воздушного Флота и войск партизанской

борьбы должен принести военную присягу в торжественной обстановке.

# Они объявили голодовку и их жизни в опасности

Президиум Президиума Верховного Совета СССР  
М. КАЛНИН,  
Секретарь Президиума Верховного Совета СССР  
А. ГОРКИН.  
Москва, Кремль.  
3 января 1939 года.

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянского Военно-



профессионалы...  
Секретарь Президиума Верховного Совета СССР  
А. ГОРКИН.  
Москва, Кремль.  
3 января 1939 года.



# ПОЛЯРНАЯ ПРАВДА

ОРГАН Мурманского Обкома и Горкома ВКП(б) и Оргкомитета Президиума Верховного Совета РСФСР по Мурманской области

5 ЯНВАРЯ 1939 г. ЧЕТВЕРГ № 4 (1073) За 1939 год. Цена 10 коп.

## Присяга родине

Слова на торжественном приеме Президиума Верховного Совета СССР в честь военной присяги Рабоче-Крестьянской Красной Армии в Рабоче-Крестьянском Красно-Флотском полку в честь юбилейной годовщины в кинотеатре «Юность» под руководством Ю. М. Дубинина. В зале присутствовали: Председатель Президиума Верховного Совета СССР И. В. Сталин, члены Президиума Верховного Совета СССР, представители Рабоче-Крестьянской Красной Армии, Рабоче-Крестьянского Красно-Флотского полка и представители народа Мурманской области.

В этот день мы должны, в этот праздник СССР быть сплоченными и крепкими, как в дни нашей военной борьбы за свободу и независимость нашей Родины.

Об этом своем величайшем долге солдаты полка были предупреждены еще в детстве. Были воспитаны на героических подвигах наших предков, которые встали на защиту нашей Родины, когда она была атакована врагом.

Наша Родина — это наша жизнь, это наша судьба, это наша честь. Мы должны быть готовы к любой жертве ради ее защиты. Мы должны быть готовы к любой смерти ради ее защиты.

Мы должны быть готовы к любой смерти ради ее защиты. Мы должны быть готовы к любой смерти ради ее защиты. Мы должны быть готовы к любой смерти ради ее защиты.

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советской Конституцией СССР, утвержденной 31 декабря 1936 года, и в целях укрепления единства и сплоченности Рабоче-Крестьянской Красной Армии...

Указ Президиума Верховного Совета СССР от 1 января 1939 года. Текст присяги Рабоче-Крестьянской Красной Армии.

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Генерал-полковник И. В. Сталин...

1. Каждый военнослужащий при поступлении в ряды Рабоче-Крестьянской Красной Армии, Рабоче-Крестьянского Красно-Флотского полка и войск внутренней охраны...

2. Военнослужащие, поступающие на службу в войска, должны быть ознакомлены с текстом присяги и подписать ее в присутствии командира подразделения.

# Йорген Холлер не подписал заявление XmlBeanDefinitionReader-а об уходе на пенсию

Президиум Президиума Верховного Совета СССР И. В. СТАЛИН, Члены Президиума Верховного Совета СССР М. КАЛНИН, А. ГОРКИН. Москва, Кремль. 3 января 1939 года.

## УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянского Военно-



# ПОЛЯРНАЯ ПРАВДА

ОРГАН  
Мурманского  
Обкома и Горкома ВКП(б)  
и Оргкомитета Президиума  
Верховного Совета РСФСР  
по Мурманской области

## Присяга родине

Слова на торжествах устно  
Присяга Верховного Совета  
СССР — торжественный зов  
Рабоче-Крестьянской Красной  
Армии и Рабоче-Крестьянского  
Красного Флота к защите нашей  
Родины.

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советской  
Смоу Советских Социалисти-  
ческих Республик, утвердить сле-  
дующий текст военной присяги  
для Рабоче-Крестьянской Красной  
Армии:

«Я, гражданин Союза Совет-  
ских Социалистических Республик,  
присягаю в рядах Рабоче-Крестьян-  
ской Красной Армии, преданно  
служить к коммунистическому делу»

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Утвердить нижеследующие по-  
ложения о порядке принятия воен-  
ной присяги:

«1. Каждый военнослужащий  
при поступлении в ряды Рабоче-  
Крестьянской Красной Армии, Ра-  
боче-Крестьянского Военно-Возду-  
шно-Флота и войск внутренней  
защиты принимает военную при-  
сягу в соответствии с Со-  
ветской Конституцией Социалисти-  
ческих Республик 132 и 133. Военнослужа-  
щий СССР



## ClassPathBeanDefinitionScanner ищет компоненты, но находит сервисы

Президиум Президиума Верховного Совета СССР  
М. КАЛИНИН,  
Секретарь Президиума Верховного Совета СССР  
А. ГОРКИН.  
Москва, Кремль.  
3 января 1939 года.

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянского Военно-

ная гражданская служба и строе-  
вых рядах внутренней службы,  
улицы и военных заведений и  
таможен военной службы, но не  
позднее 2-х месяцев со дня при-  
бытия в военную часть.

б) тексты военной присяги в  
следующей форме: «Я, гражданин  
Социалистических Республик, —  
во исполнение воле вождя,  
и Милости прощенья и  
спасения, а также в защиту  
коммуны нашей и мира в му-  
ндале алтарей трудовой  
защиты под руководством командира

присягу свою совершаю  
в соответствии с Конституцией  
Социалистических Республик  
132 и 133. Военнослужащий  
СССР

Примечание: 1. Каждый  
военнослужащий СССР, при  
призыве в армию, военно-  
воздушный флот или в войска  
внутренней защиты, должен  
принять военную присягу в со-  
ответствии с Конституцией  
Социалистических Республик  
132 и 133. Военнослужащий  
СССР





5 ЯНВАРЯ 1938 г. ЧЕТВЕРГ № 4 (1073) За один рубль. Цена 10 коп.

# ПОЛЯРНАЯ ПРАВДА

ОРГАН Мурманского Обкома и Горкома ВКП(б) и Оргкомитета Президиума Верховного Совета РСФСР по Мурманской области

## Присяга родине

Слова на торжественном приеме в честь юбилея Советского Союза в театре оперы и балета Мурманской Краевой Армии и Рабоче-Крестьянской Красной Армии. В зале собралось много молодежи, которая в этот вечер принимала участие в торжественном приеме. В зале собралось много молодежи, которая в этот вечер принимала участие в торжественном приеме.

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянской Красной Армии

В соответствии с Советской Конституцией и действующим законодательством Советом Советов Социалистических Республик, утвержденным постановлением Президиума Верховного Совета СССР от 10 января 1938 года, текст военной присяги Рабоче-Крестьянской Красной Армии устанавливается следующий:

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О порядке принятия военной присяги

Генеральным маршалом Советского Союза, Верховным Главнокомандующим Рабоче-Крестьянской Красной Армии, утвержденным постановлением Президиума Верховного Совета СССР от 10 января 1938 года, порядок принятия военной присяги устанавливается следующий:



Теперь он выглядит вот так:

Война будет победоносной для Советского Союза. Мы будем бороться до последней капли крови за свободу и независимость нашей Родины. Мы будем бороться до последней капли крови за свободу и независимость нашей Родины.

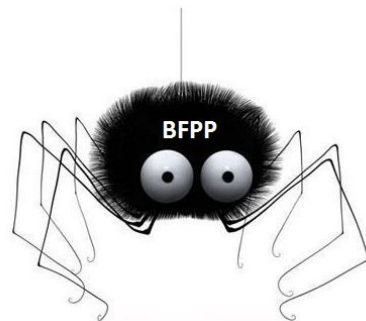
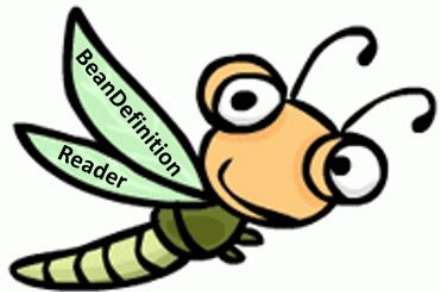
Президиум Президиума Верховного Совета СССР М. КАЛНИН, Секретарь Президиума Верховного Совета СССР А. ГОРКИН. Москва, Кремль. 3 января 1938 года.

### УКАЗ ПРЕЗИДИУМА ВЕРХОВНОГО СОВЕТА СССР О тексте военной присяги Рабоче-Крестьянского Военно-

Военно-воздушных сил. Текст военной присяги. Присяга принимается в торжественной обстановке. Присяга принимается в торжественной обстановке.



# Ретроспектива











# Spring Boot

The Ripper





Железный банк Браавоса

The image shows the title card for the TV series "Game of Thrones". It features a dark, ornate banner with the title "GAME OF THRONES" in a glowing, golden font. The banner is set against a background of a large, intricately carved wooden archway, possibly a gate or a throne, with a bright, golden light emanating from behind it. The archway is supported by four animal heads: a dragon on the left, a lion on the right, a bear on the bottom left, and a stag on the bottom right. The overall atmosphere is dramatic and epic.

# GAME OF THRONES

Демо

# Что **не любит** программист

## Думать о:



1. Зависимостях...
2. Настройках и конфигурациях

## Делать:

3. Деплоймент и запуск (тест/локально/прод)

# Что **не** любит программист

Думать о:

-  1. Зависимостях...
-  2. Настройках и конфигурациях

Делать:

-  3. Деплоймент и запуск (тест/локально/прод)

Зависимости



# Как выглядит наш pom.xml

```
<parent>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-parent</artifactId>  
  <version>1.5.13.RELEASE</version>  
</parent>
```

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.13.RELEASE</version>
</parent>
```




```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-dependencies</artifactId>
  <version>1.5.13.RELEASE</version>
</parent>
```

```
2414     <groupId>org.webjars</groupId>
2415     <artifactId>webjars-locator</artifactId>
2416     <version>${webjars-locator.version}</version>
2417 </dependency>
2418 <dependency>
2419     <groupId>org.yaml</groupId>
2420     <artifactId>snakeyaml</artifactId>
2421     <version>${snakeyaml.version}</version>
2422 </dependency>
2423 <dependency>
2424     <groupId>redis.clients</groupId>
2425     <artifactId>jedis</artifactId>
2426     <version>${jedis.version}</version>
2427 </dependency>
2428 <dependency>
2429     <groupId>wsdl4j</groupId>
2430     <artifactId>wsdl4j</artifactId>
2431     <version>${wsdl4j.version}</version>
2432 </dependency>
2433 <dependency>
2434     <groupId>xml-apis</groupId>
2435     <artifactId>xml-apis</artifactId>
2436     <version>${xml-apis.version}</version>
2437 </dependency>
2438 </dependencies>
2439 </dependencyManagement>
```



В чём проблема получить  
**dependency management** таким путём?

A close-up photograph of a baby with light brown hair and large, dark eyes. The baby is wearing a bright red zip-up jacket over a white shirt. The baby's mouth is slightly open, showing small teeth. A white speech bubble with a black outline is positioned to the left of the baby's face, containing the Russian text "Это чужой parent!". The background is dark and out of focus, suggesting an indoor setting.

Это чужой parent!

ВЕРНИТЕ

НАШИХ

РОДИТЕЛЕЙ



```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>io.spring.platform</groupId>
      <artifactId>platform-bom</artifactId>
      <version>Brussels-SR2</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

# Тоже самое в **build.gradle**

```
dependencyManagement {  
    imports {  
        mavenBom 'org.springframework.cloud:spring-cloud-dependencies:Dalston.RELEASE'  
    }  
}
```



# Какие теперь зависимости?

```
'org.springframework.boot:spring-boot-starter-web'
```

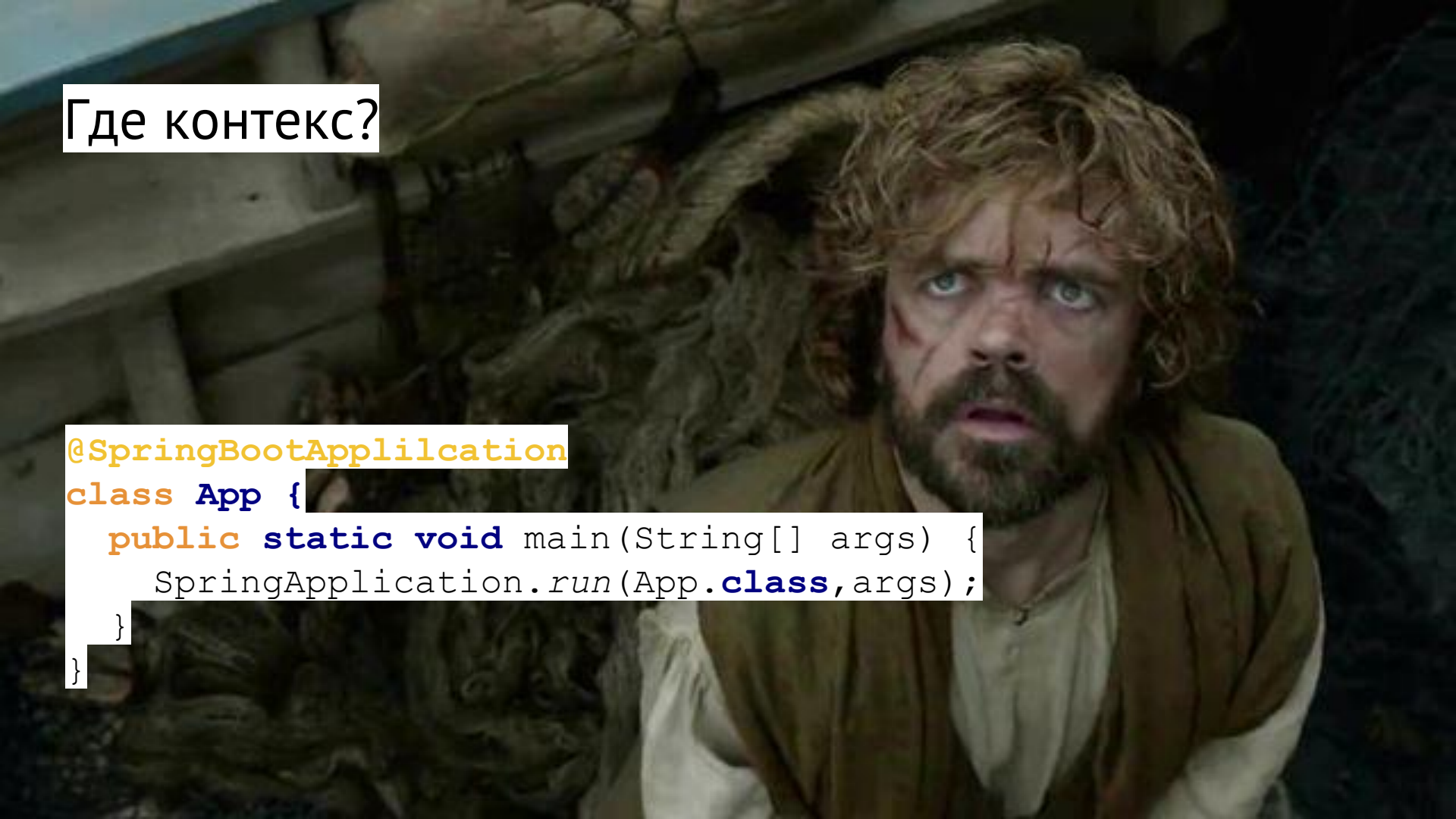
```
'org.springframework.boot:spring-boot-starter-data-jpa'
```

```
'com.h2database:h2'
```

Настройки контекста

Где контекст?

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



Вот контекст!

```
@SpringBootApplication
class App {
    public static void main(String[] args) {
        ApplicationContext context =
            SpringApplication.run(App.class, args);
    }
}
```

# Загадочные аргументы

Дано: `RipperApplication.class`

```
public... main(String[] args) {  
    SpringApplication.run(?, args);  
}
```

1. `RipperApplication.class`
2. `String.class`
3. `"context.xml"`
4. `new ClassPathResource("context.xml")`
5. `Package.getPackage("conference.spring.boot.ripper")`



```
SpringApplication.run(Object[] sources String[] args)
```



# Загадочные аргументы

Голосуем!

Дано: `RipperApplication.class`

```
public... main(String[] args) {  
    SpringApplication.run(?, args);  
}
```

1. `RipperApplication.class`
2. `String.class`
3. `"context.xml"`
4. `new ClassPathResource("context.xml")`
5. `Package.getPackage("conference.spring.boot.ripper")`

Все верны







```
SpringApplication.run(Object[] sources, String[] args)
```

```
# APPLICATION SETTINGS (SpringApplication)
```

```
spring.main.sources= # class name, package name, xml location
```

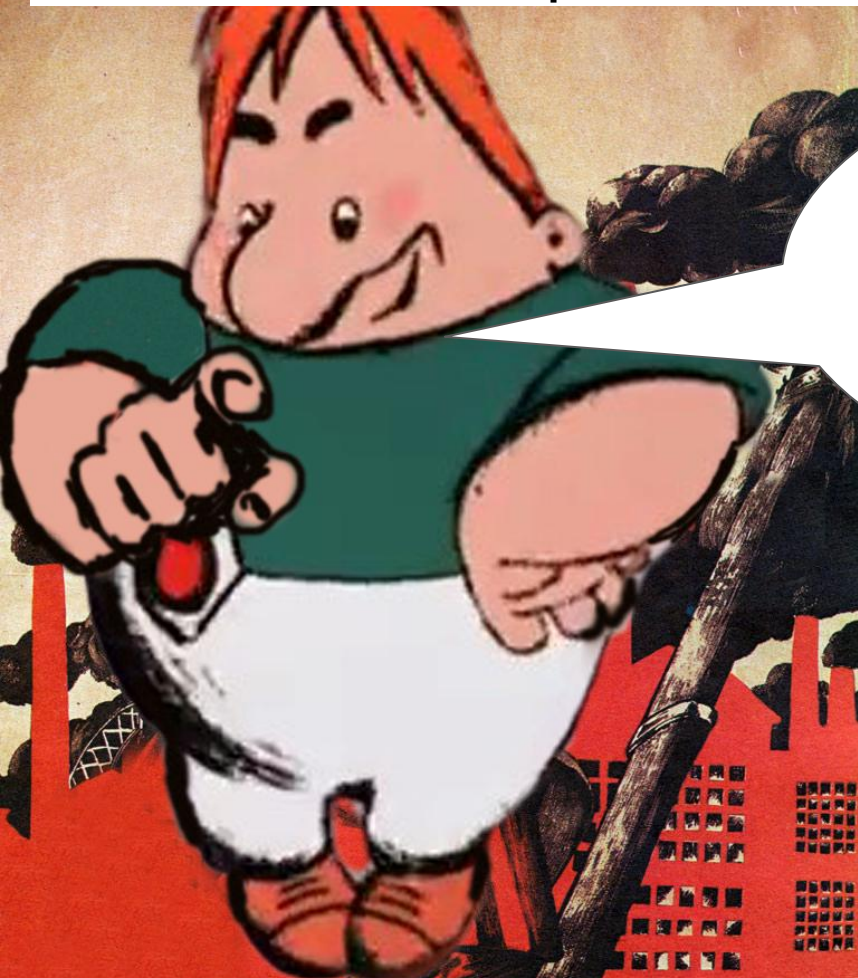
```
spring.main.web-environment= # true/false
```

```
spring.main.banner-mode=console # Log/off
```

Разрешите представиться



# Давайте поговорим о контексте



Малыш знает про  
`ClassPathXmlApplicationContext`  
А какие контексты знаешь Ты?



# Типы `ConfigurableApplicationContext`

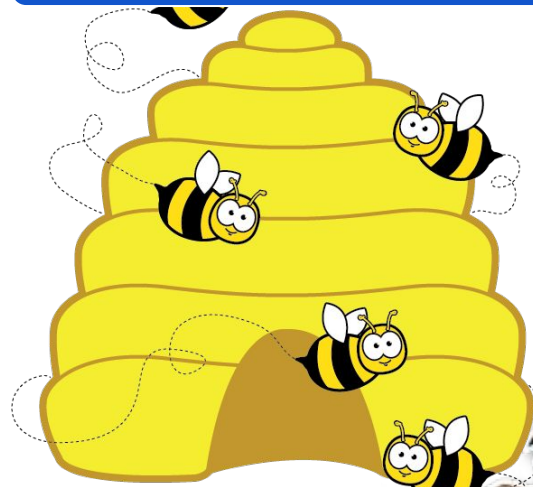
- AnnotationConfigApplicationContext (org.springframework.context.annotation)
- AnnotationConfigEmbeddedWebApplicationContext (org.springframework.boot.context.embedded)
- AnnotationConfigWebApplicationContext (org.springframework.web.context.support)
- ClassPathXmlApplicationContext (org.springframework.context.support)
- ConfigurableWebApplicationContext (org.springframework.web.context)
- EmbeddedWebApplicationContext (org.springframework.boot.context.embedded)
- FileSystemXmlApplicationContext (org.springframework.context.support)
- GenericApplicationContext (org.springframework.context.support)
- GenericGroovyApplicationContext (org.springframework.context.support)
- GenericWebApplicationContext (org.springframework.web.context.support)
- GenericXmlApplicationContext (org.springframework.context.support)
- GroovyWebApplicationContext (org.springframework.web.context.support)
- StaticApplicationContext (org.springframework.context.support)
- StaticWebApplicationContext (org.springframework.web.context.support)
- XmlEmbeddedWebApplicationContext (org.springframework.boot.context.embedded)

# Это я решаю какой контекст создать

Я до создания  
контекста ещё много  
всего делаю, но об  
этом потом.



Web Context



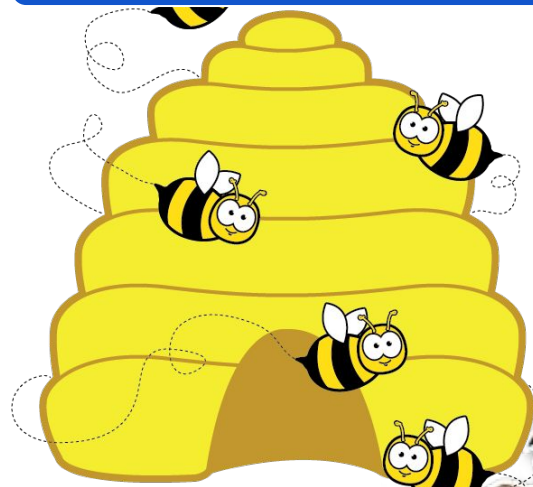
Generic Context



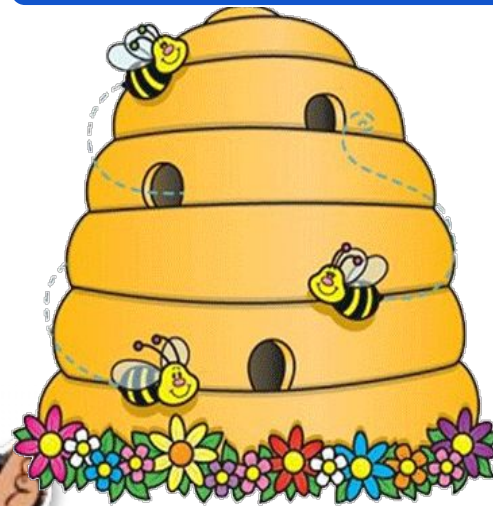
Я САМЫЙ  
ТЯЖЕЛО  
БОЛЬНОЙ  
ЧЕЛОВЕК  
НА СВЕТЕ



## Web Context



## Generic Context



Если в classpath есть  
Servlet.class...



# Если есть

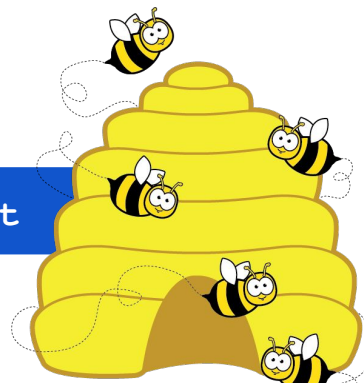
`javax.servlet.Servlet`

+

`ConfigurableWebApplicationContext`



`AnnotationConfigEmbedded WebApplicationContext`



# Иначе



`AnnotationConfig ApplicationContext`



И что там в контексте то?

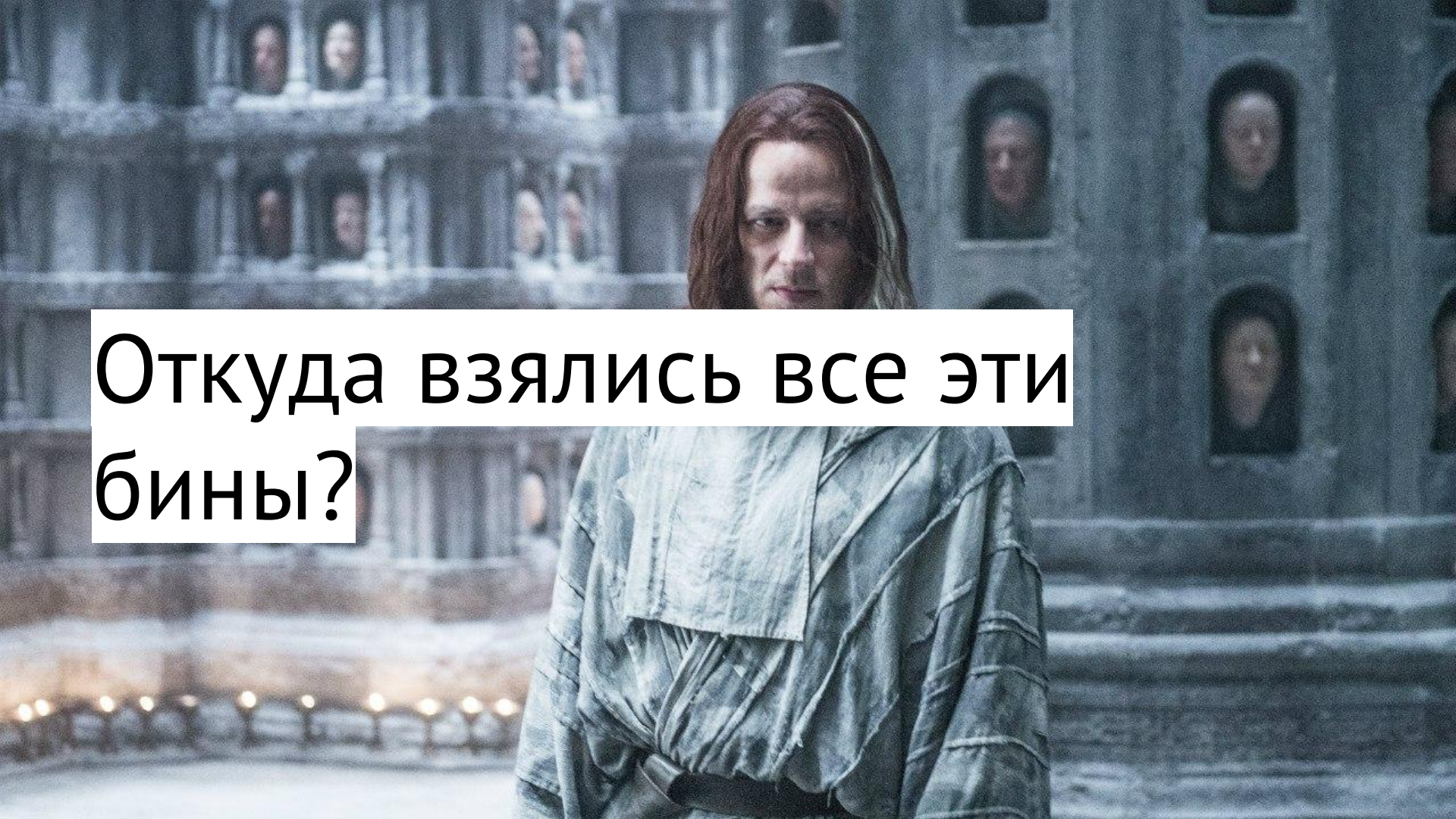
# Откуда 436 spring beans?

The screenshot shows an IDE's 'Evaluate' window. The expression `run.getBeanDefinitionNames()` has been evaluated, resulting in a list of 17 bean names. The first element of the list is `{String[436]@9293}`, which is highlighted with a red box. A red arrow points from the title 'Откуда 436 spring beans?' to this box. The list of bean names includes various Spring framework internal processors, application-specific beans like 'moneyRavenApplication', and controllers like 'profileController' and 'repositoryController'.

Expression: `run.getBeanDefinitionNames()`

Result:

- 0 = "org.springframework.context.annotation.internalConfigurationAnnotationProcessor"
- 1 = "org.springframework.context.annotation.internalAutowiredAnnotationProcessor"
- 2 = "org.springframework.context.annotation.internalRequiredAnnotationProcessor"
- 3 = "org.springframework.context.annotation.internalCommonAnnotationProcessor"
- 4 = "org.springframework.context.annotation.internalPersistenceAnnotationProcessor"
- 5 = "org.springframework.context.event.internalEventListenerProcessor"
- 6 = "org.springframework.context.event.internalEventListenerFactory"
- 7 = "moneyRavenApplication"
- 8 = "org.springframework.boot.autoconfigure.internalCachingMetadataReaderFactory"
- 9 = "ironBankController"
- 10 = "transferMoneyProphecyBackend"
- 11 = "whiteListProphetService"
- 12 = "halBrowser"
- 13 = "repositoryEntityController"
- 14 = "alpsController"
- 15 = "profileController"
- 16 = "repositoryController"

A man with long, straight brown hair and a serious expression stands in the center of the frame. He is wearing a heavy, grey, textured robe with a dark belt. The background is a large, grey stone building with many arched windows, some of which contain faces. The scene is lit with a cool, blueish-grey light, suggesting an overcast day or a specific lighting effect. In the foreground, there are some blurred lights and a stone ledge.

Откуда взялись все эти  
бины?

Не знаю как, но мы тоже так сделаем



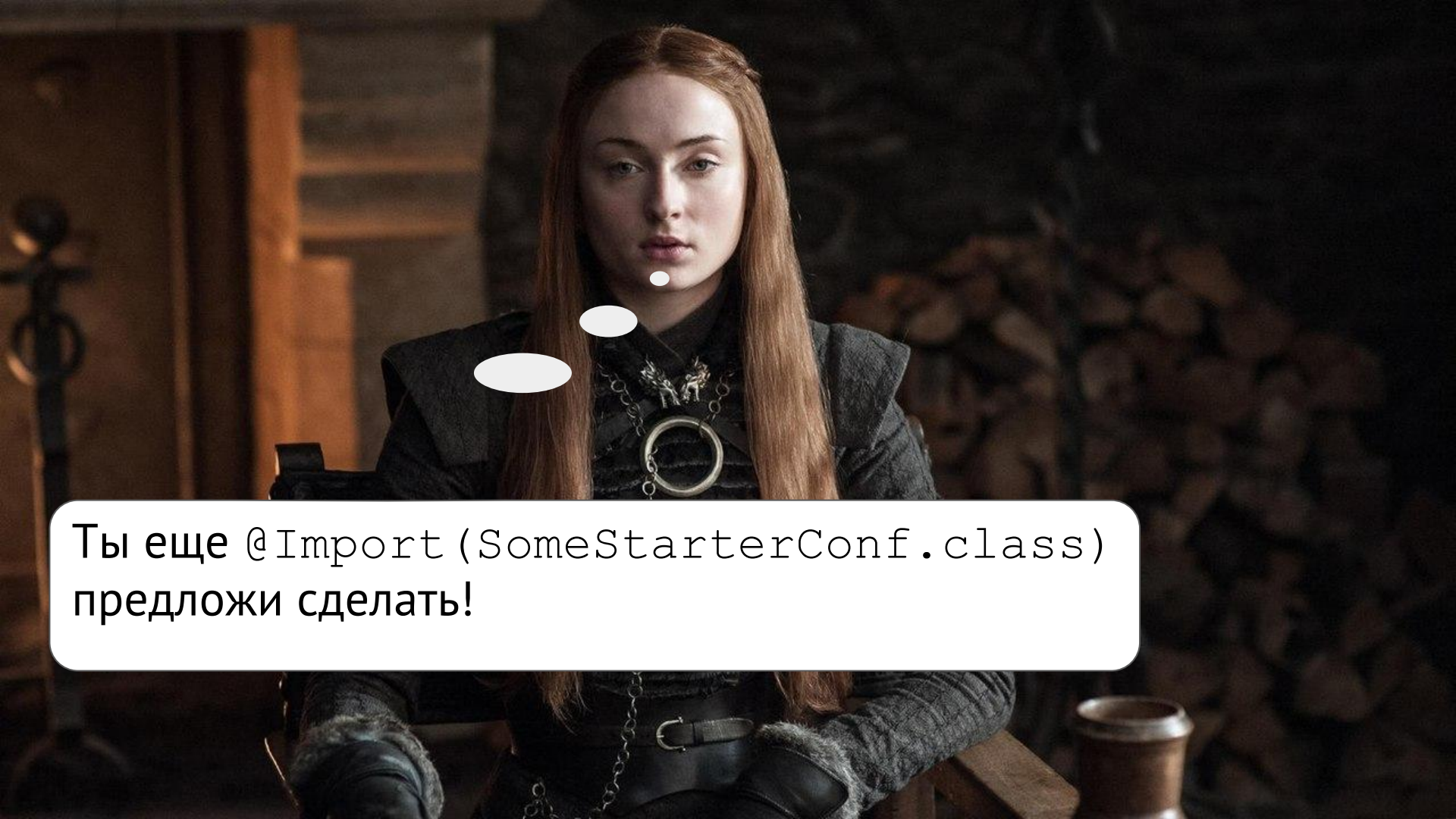


# Железный закон №1

В любой непонятной ситуации посылай **ворона**



Может @EnableSomeStarter ?



Ты еще `@Import (SomeStarterConf.class)`  
предложи сделать!

A man with a beard and dark hair is shown from the chest up, looking slightly to the right with a thoughtful expression. A white thought bubble is positioned above his head. The background is dark with some blurred lights.

Мы будем использовать `spring.factories!`



# Что за `spring.factories` ?



## § 44.1 Understanding auto-configured beans

Under the hood, auto-configuration is implemented with standard `@Configuration` classes. Additional `@Conditional` annotations are used to constrain when the auto-configuration should apply. Usually auto-configuration classes use `@ConditionalOnClass` and `@ConditionalOnMissingBean` annotations. This ensures that auto-configuration only applies when relevant classes are found and when you have not declared your own `@Configuration`.

You can browse the source code of [spring-boot-autoconfigure](#) to see the `@Configuration` classes that we provide (see the [META-INF/spring.factories](#) file).



Даёшь инверсию контроля для стартеров



# Железный закон №1.1

В любой непонятной ситуации посылай **ворона**

@SpringBootApplication

**Всему голова**

@SpringBootApplication



# @SpringBootApplication

→ @ComponentScan

→ @Configuration

→ @EnableAutoConfiguration

# @SpringBootApplication

```
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@SpringBootConfiguration
@EnableAutoConfiguration
@ComponentScan(excludeFilters = {
    @Filter(type = FilterType.CUSTOM, classes = TypeExcludeFilter.class),
    @Filter(type = FilterType.CUSTOM, classes =
AutoConfigurationExcludeFilter.class) })
public @interface SpringBootApplication {
```

...

}

# @SpringBootApplication

```
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@SpringBootConfiguration
@EnableAutoConfiguration
@ComponentScan(excludeFilters = {
    @Filter(type = FilterType.CUSTOM, classes = TypeExcludeFilter.class),
    @Filter(type = FilterType.CUSTOM, classes =
AutoConfigurationExcludeFilter.class) })
public @interface SpringBootApplication {
    ...
}
```



# @EnableAutoConfiguration

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@AutoConfigurationPackage
@Import({EnableAutoConfigurationImportSelector.class})
public @interface EnableAutoConfiguration {
    String ENABLED_OVERRIDE_PROPERTY = "spring.boot.enableautoconfiguration";
    Class<?>[] exclude() default {};
    String[] excludeName() default {};
}
```

# @EnableAutoConfiguration

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@AutoConfigurationPackage
@Import({EnableAutoConfigurationImportSelector.class})
public @interface EnableAutoConfiguration {
    String ENABLED_OVERRIDE_PROPERTY = "spring.boot.enableautoconfiguration";
    Class<?>[] exclude() default {};
    String[] excludeName() default {};
}
```

# @EnableAutoConfiguration

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@AutoConfigurationPackage
@Import({EnableAutoConfigurationImportSelector.class})
public @interface EnableAutoConfiguration {
    String ENABLED_OVERRIDE_PROPERTY = "spring.boot.enableautoconfiguration";
    Class<?>[] exclude() default {};
    String[] excludeName() default {};
}
```



@EnableAutoConfiguration

... Starter

Iron Starter

Web Starter

Boot Starter

ImportSelector



# @EnableAutoConfiguration

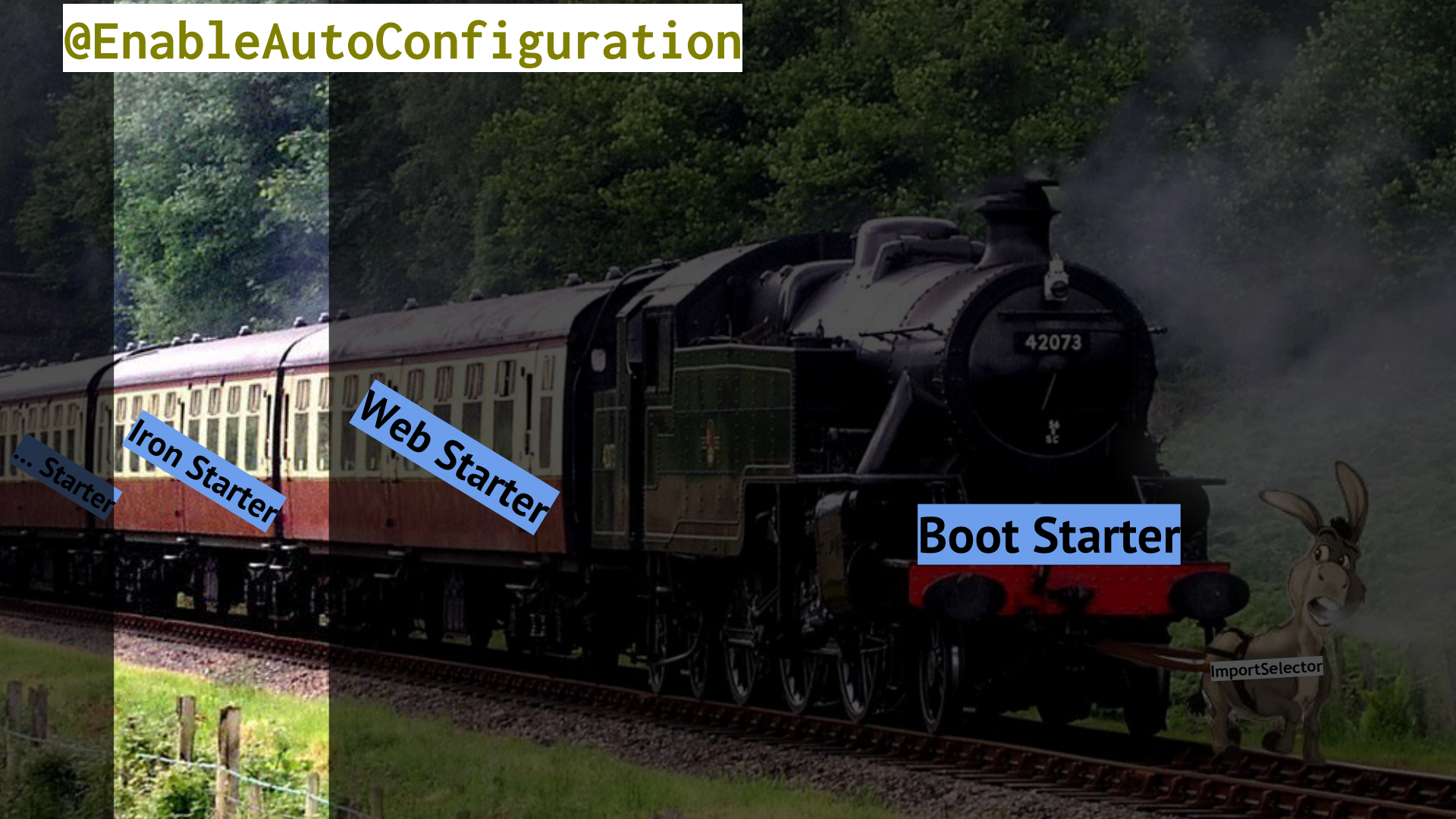
... Starter

Iron Starter

Web Starter

Boot Starter

ImportSelector



# Spring Factories Loader



# SpringFactoriesLoader

```
static <T> List<T> loadFactories(  
    Class<T> factoryClass,  
    ClassLoader cl  
)
```

```
static List<String> loadFactoryNames(  
    Class<?> factoryClass,  
    ClassLoader cl  
)
```





Даёшь инверсию контроля для стартеров



# spring-boot-autoconfigure.jar/spring.factories

```
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
  org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration
  org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
  org.springframework.boot.autoconfigure.aop.AopAutoConfiguration,\
  org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,\
  org.springframework.boot.autoconfigure.batch.BatchAutoConfiguration,\
  org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration.\
```

...

...

# spring-boot-autoconfigure.jar/spring.factories

```
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\n  org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration\n  org.springframework.boot.autoconfigure.EnableAutoConfiguration=\n  org.springframework.boot.autoconfigure.aop.AopAutoConfiguration,\n  org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,\n  org.springframework.boot.autoconfigure.batch.BatchAutoConfiguration,\n  org.springframework.boot.autoconfigure.cache.CacheAutoConfiguration.\n
```

...

...

spring-boot-autoconfigure.jar

```
@Import(CacheConfigurationImportSelector.class)  
public class CacheAutoConfiguration {
```

## Где то внутри CacheAutoConfiguration

```
for (int i = 0; i < types.length; i++) {  
    Imports[i] = CacheConfigurations.getConfigurationClass(types[i]);  
}  
return imports;
```

# CacheConfigurations

```
private static final Map<CacheType, Class<?>> MAPPINGS;  
static {  
    Map<CacheType, Class<?>> mappings = new HashMap<CacheType, Class<?>>();  
    mappings.put(CacheType.GENERIC, GenericCacheConfiguration.class);  
    mappings.put(CacheType.EHCACHE, EhCacheCacheConfiguration.class);  
    mappings.put(CacheType.HAZELCAST, HazelcastCacheConfiguration.class);  
    mappings.put(CacheType.INFINISPAN, InfinispanCacheConfiguration.class);  
    mappings.put(CacheType.JCACHE, JCacheCacheConfiguration.class);  
    mappings.put(CacheType.COUCBASE, CouchbaseCacheConfiguration.class);  
    mappings.put(CacheType.REDIS, RedisCacheConfiguration.class);  
    mappings.put(CacheType.CAFFEINE, CaffeineCacheConfiguration.class);  
    addGuavaMapping(mappings);  
    mappings.put(CacheType.SIMPLE, SimpleCacheConfiguration.class);  
    mappings.put(CacheType.NONE, NoOpCacheConfiguration.class);  
    MAPPINGS = Collections.unmodifiableMap(mappings);  
}
```



Захардкодили

A close-up photograph of a woman with dark, curly hair. She has a shocked or surprised expression, with her mouth wide open and her hands covering her face. The background is blurred, suggesting an outdoor setting with greenery. A white rectangular box is superimposed over the center of the image, containing the Russian text "И что, они всегда грузятся?".

И что, они всегда грузятся?

Спокойствие, есть @Conditional  
Они фильтруются





Почему бин то есть то нет?

Какие условия?



Не знаю как, но мы тоже так сделаем





# Железный закон №1.2

Посылай **ворона** только в продакшене

@ConditionalOnProduction – Демо

# Паззлер

```
@Configuration
@ConditionalOnСуроваяЗима
public class UndeadArmyConfiguration {
    ...
}
```

```
@Configuration
public class DragonIslandConfiguration {
    @Bean
    @ConditionalOnСуроваяЗима
    public DragonGlassFactory dragonGlassFactory() {
        return new DragonGlassFactory();
    }
    ...
}
```

# ConditionalOnСуроваяЗима

1. 100 \$
2. 200 \$
3. 300 или 400 \$
4. между 500 и 1000 \$
5. Больше 1000 \$

\* Спросить погоду стоит денег – 100 золотых драконов за раз

\* \$ = золотой дракон



# ConditionalOnСуроваяЗима

1. 100 \$
2. 200 \$
3. 300 или 400 \$
4. между 500 и 1000 \$
5. Больше 1000 \$

\* Спросить погоду стоит денег – 100 золотых драконов за раз

\* \$ = золотой дракон




```
@Configuration
@ConditionalOnСуроваяЗима
public class UndeadArmyConfiguration {
    ...
}

@Configuration
public class DragonIslandConfiguration {
    @Bean
    @ConditionalOnСуроваяЗима
    public DragonGlassFactory dragonGlassFactory() {
        return new DragonGlassFactory();
    }
    ...
}
```

## Правильный ответ

1. 100 \$

2. 200 \$

 300 или 400 \$

4. между 500 и 1000 \$

5. Больше 1000 \$






## Правильный ответ

1. 100 \$

2. 200 \$

 300 или 400 \$

4. между 500 и 1000 \$

5. Больше 1000 \$

300 – если `UndeadArmyConfiguration` в стартере

400 – если `UndeadArmyConfiguration` в коде приложения

Как это  
работает ?



Как то так, через хитро  
закрученную жопу  
оно и работает





Лечу  
куда хочу

Лететь то куда?



# Железный закон №1.3

Отправлять ворона только если есть получатели

1. список получателей
2. без списка не летим
3. автокомплит



# Conditional и друзья

@ConditionalOnBean

@ConditionalOnClass

@ConditionalOnCloudPlatform

@ConditionalOnExpression

@ConditionalOnJava

@ConditionalOnJndi

@ConditionalOnMissingBean

@ConditionalOnMissingClass

@ConditionalOnNotWebApplication

@ConditionalOnProperty

@ConditionalOnResource

@ConditionalOnSingleCandidate

@ConditionalOnWebApplication

...



Лететь то куда?

# Железный закон №1.3

Отправлять ворона только если есть получатели

1. список получателей ← `application.yml`
2. без списка не летим ← `@ConditionalOnProperty`
3. автокомплит ← `@ConfigurationProperty`

# Железный закон №1.4

Отправляем ворона только если нет кастомного





# Conditional и друзья

@ConditionalOnBean

@ConditionalOnClass

@ConditionalOnCloudPlatform

@ConditionalOnExpression

@ConditionalOnJava

@ConditionalOnJndi

**@ConditionalOnMissingBean**

@ConditionalOnMissingClass

@ConditionalOnNotWebApplication

@ConditionalOnProperty

@ConditionalOnResource

@ConditionalOnSingleCandidate

@ConditionalOnWebApplication

...

# Железный закон №1.4

Отправляем ворона только если нет кастомного

@ConditionalOnMissingBean



# ConditionalOnPuzzler

```
@Configuration
public class КонфигурацияКазни {

    @Bean
    @ConditionalOnClass ({Мыло.class, Веревка.class})
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
    public ФабрикаКазни виселицы() { return new ФабрикаВиселиц("..."); }

}
```

# ConditionalOnPuzzler

```
@Configuration
```

```
public class КонфигурацияКазни {
```

```
    @Bean
```

```
    @ConditionalOnClass ({Мыло.class, Веревка.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни виселицы() { return new ФабрикаВиселиц("..."); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Стул.class, Ток.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни стулья() { return new ФабрикаЭлектрическихСтульев("ВЖ ВЖ");
```

```
}
```

# ConditionalOnPuzzler

```
@Configuration
```

```
public class КонфигурацияКазни {
```

```
    @Bean
```

```
    @ConditionalOnClass ({Мыло.class, Веревка.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни виселицы() { return new ФабрикаВиселиц("..."); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Стул.class, Ток.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни стулья() { return new ФабрикаЭлектрическихСтульев("вж вж"); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Гильотина.class, ХорошееНастроение.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни гильотины() { return new ФабрикаГильотин("хрусть хрусть"); }
```

```
}
```

# ConditionalOnPuzzler

```
@Configuration
```

```
public class КонфигурацияКазни {
```

```
    @Bean
```

```
    @ConditionalOnClass ({Мыло.class, Веревка.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни виселицы() { return new ФабрикаВиселиц ("..."); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Стул.class, Ток.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни стулья() { return new ФабрикаЭлектрическихСтульев ("вж вж"); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Гильотина.class, ХорошееНастроение.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни гильотины() { return new ФабрикаГильотин ("хрусть хрусть"); }
```

```
}
```

1. ClassDefNotFound ?
2. Так вообще нельзя, не компилируется
3. Будет работать
4. Будет отлично работать

# ConditionalOnPuzzler

@Configuration

```
public class КонфигурацияКазни {
```

```
    @Bean
```

```
    @ConditionalOnClass ({Мыло.class, Веревка.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни виселицы() { return new ФабрикаВиселиц ("..."); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Стул.class, Ток.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни стулья() { return new ФабрикаЭлектрическихСтульев ("вж вж"); }
```

```
    @Bean
```

```
    @ConditionalOnClass ({Гильотина.class, ХорошееНастроение.class})
```

```
    @ConditionalOnMissingBean ({ФабрикаКазни.class})
```

```
    public ФабрикаКазни гильотины() { return new ФабрикаГильотин ("хрусть хрусть"); }
```

```
}
```

1. ClassDefNotFound ?
2. Так вообще нельзя, не компилируется
3. **Будет работать**
4. Будет отлично работать



Потому что ASM



# Железный закон №1.5

ВКЛ/ВЫКЛ ворона

@ConditionalOnProperty



# OnPropertyCondition

```
@Conditional (OnPropertyCondition.class)
public @interface ConditionalOnProperty {
```

```
String[] value() default {};
```



```
String prefix() default "";
```

```
String[] name() default {};
```

```
String havingValue() default "";
```



Что если нужно изменить?

```
boolean matchIfMissing() default false;
```

```
boolean relaxedNames() default true;
```

```
}
```

# OnPropertyCondition

```
@ConditionalOnProduction
@ConditionalOnProperty(
    name = "ворон.вкл"
)
public IronBankApplicationListener applicationListener() {
    ...
}
```

# OnPropertyCondition

```
@ConditionalOnProduction
@ConditionalOnProperty(
    name = "ворон.вкл",
    havingValue = "true"
)
public IronBankApplicationListener applicationListener() {
    ...
}
```

# OnPropertyCondition

```
@ConditionalOnProduction
@ConditionalOnProperty(
    name = {
        "ворон.вкл",
        "зима.вкл",
        "старки.вкл"
    },
    havingValue = "true"
)
public IronBankApplicationListener applicationListener() {
    ...
}
```

# OnPropertyCondition

```
@ConditionalOnProduction  
@ConditionalOnProperty(  
    name = {  
        "ворон.вкл",  
        "зима.вкл",  
        "старки.вкл"  
    },  
    havingValue = "true" ← Не массив  
)  
public IronBankApplicationListener applicationListener() {  
    ...  
}
```

# OnPropertyCondition

```
@ConditionalOnProduction
@ConditionalOnProperty(name = "ворон.вкл", havingValue="true")
@ConditionalOnProperty(name = "зима.вкл", havingValue="true")
@ConditionalOnProperty(name = "старки.вкл", havingValue="false")
public IronBankApplicationListener applicationListener() {
    ...
}
```

# OnPropertyCondition

```
@ConditionalOnProduction
@ConditionalOnProperty(name = "ворон.вкл", havingValue="true")
@ConditionalOnProperty(name = "зима.вкл", havingValue="true")
@ConditionalOnProperty(name = "старки.вкл", havingValue="false")
public IronBankApplicationListener applicationListener() {
    ...
}
```

@ConditionalOnProperty ← Не @Repeatable





**AllNestedConditions**

& &


**AnyNestedCondition**

# OnPropertyCondition

```
@Retention (RetentionPolicy.RUNTIME)  
@Target ({ ElementType.TYPE, ElementType.METHOD })  
@Conditional (CompositeCondition.class)  
public @interface ConditionalOnRaven {  
}
```

# OnPropertyCondition

```
public class OnRavenCondition extends AllNestedConditions {  
  
    @ConditionalOnProperty(  
        name = "ворон.куда",  
        havingValue = "false")  
    public static class OnRavenProperty { }  
  
    @ConditionalOnProperty(  
        name = "ворон.вкл",  
        havingValue = "true",  
        matchIfMissing = true)  
    public static class OnRavenEnabled { }  
  
    ...  
}
```



# OnPropertyCondition

@ConditionalOnRaven

```
public IronBankApplicationListener applicationListener() {  
    ...  
}
```






WINTER IS ~~COMING~~

Here

Сейчас будет спойлер



Сезон 8 – Зима Тута  
Кофебрейк

Внимание спойлеры



A shirtless man with a beard and a bloody forehead, looking shocked, with a bloody wound on his chest. He is standing in front of a dark wooden wall with a large, dark, rectangular object behind him.

В 8 сезоне

Железный банк изменит правила выдачи кредитов  
— Демо



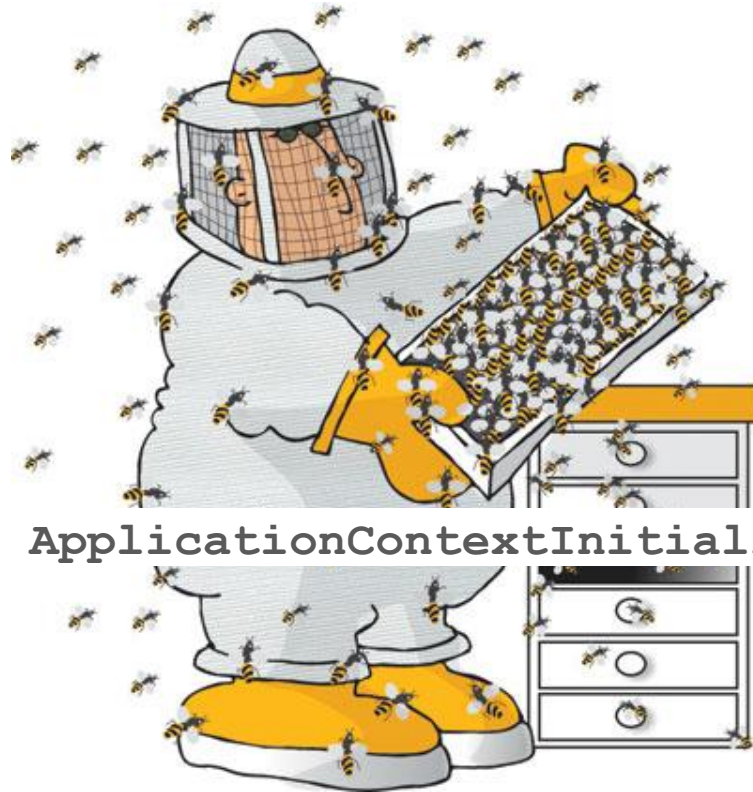
# Железный закон №2

Без профиля - нельзя

From Lea



# Demo



ApplicationContextInitializer

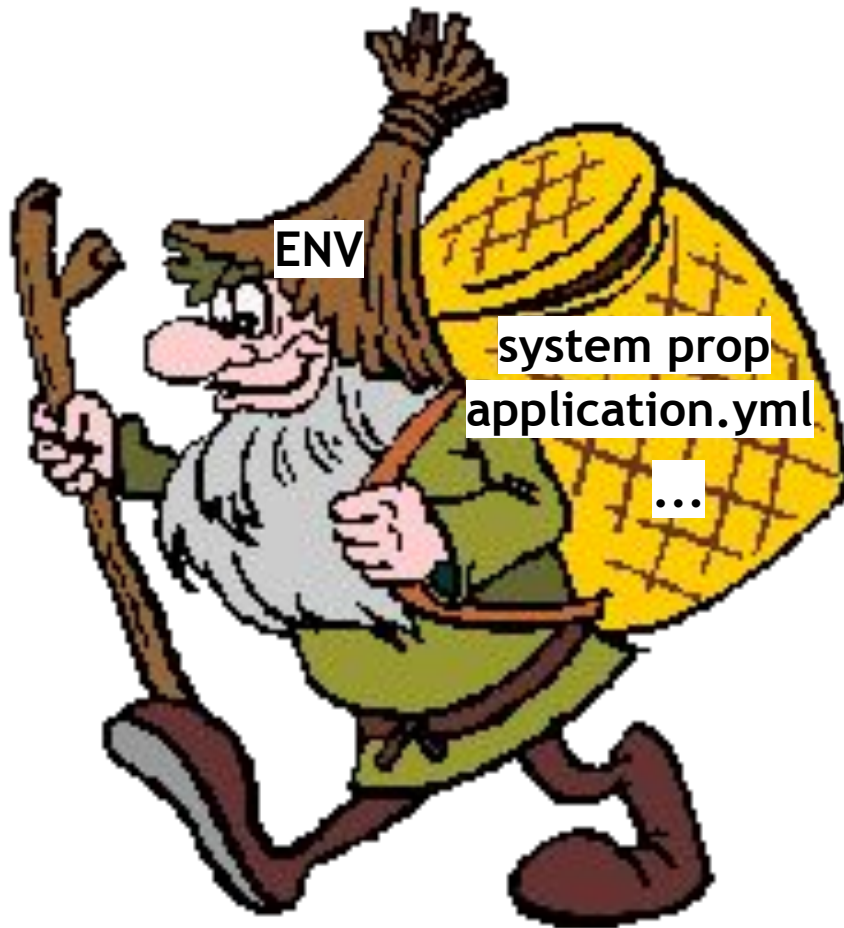
# Environment


Property sources:

- systemProperties
- system environment
- random property
- application.properties
- ...

Active Profiles

Default Profiles



A man with short brown hair, wearing a dark, textured sweater, is shown from the chest up. He is holding a wooden handle, possibly part of a tool or a musical instrument. The background is dark and out of focus. A white text box is overlaid on the image, containing the Russian text: "Почему бы не определить профиль автоматически?".

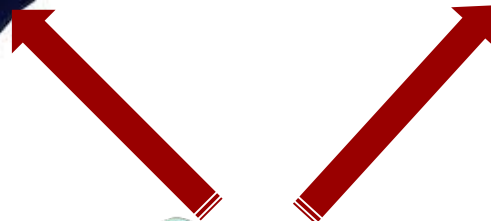
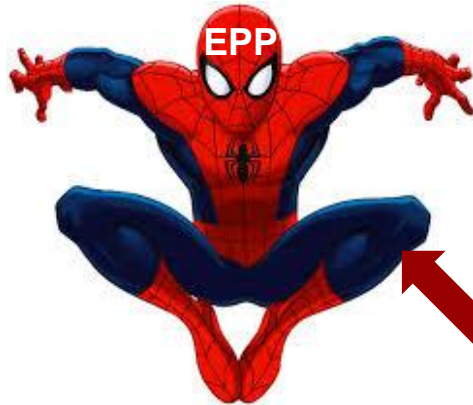
Почему бы не определить профиль  
автоматически?

Напишем... EnvironmentPostProcessor

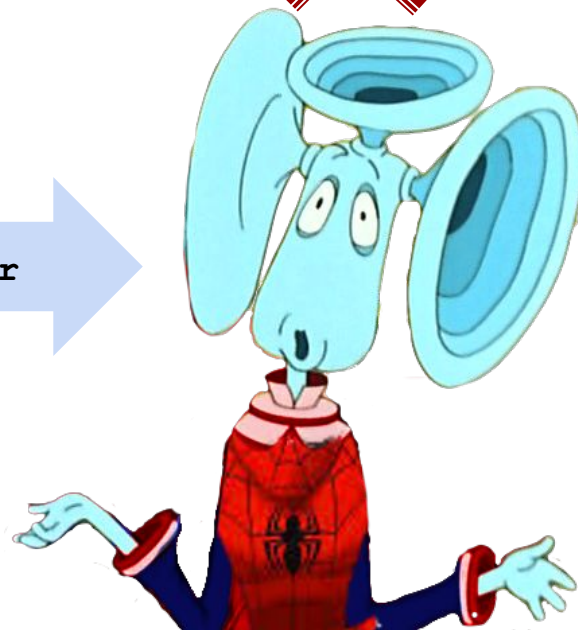


demo





ConfigFileApplicationListener



# Карлсон строит environment





Карлсон дает листенеру environment и себя



# ConfigFileApplicationListener

Слушает

- `ApplicationPreparedEvent`
- `ApplicationEnvironmentPreparedEvent`

Загружает

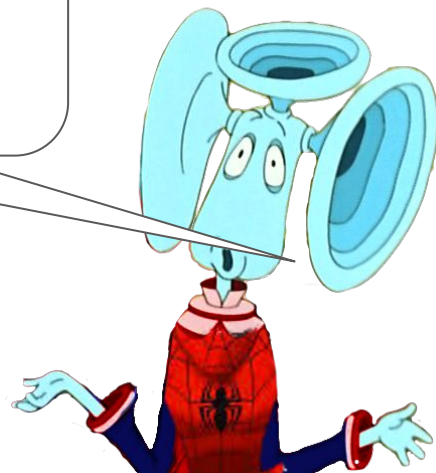
- `application.yml`
- `application.properties`
- `env vars`
- `cmd args`



# onApplicationEnvironmentPreparedEvent



Сгоняй за  
EnvironmentPostProcessor-ами



# onApplicationEnvironmentPreparedEvent

```
SpringFactoriesLoader.  
loadFactories (  
    EnvironmentPostProcessor.class  
...)
```

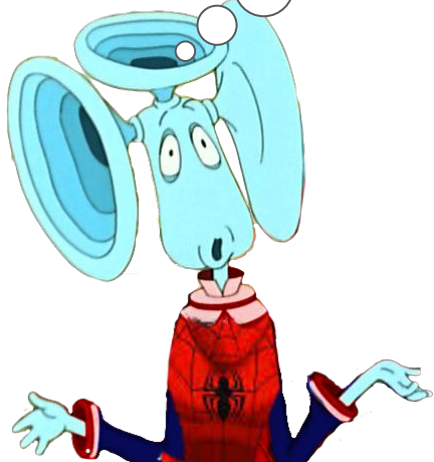


onApplicationEnvironmentPreparedEvent



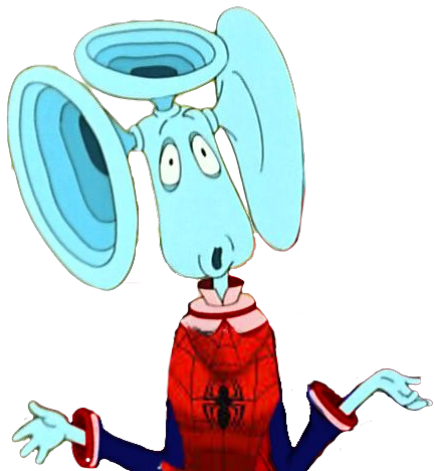
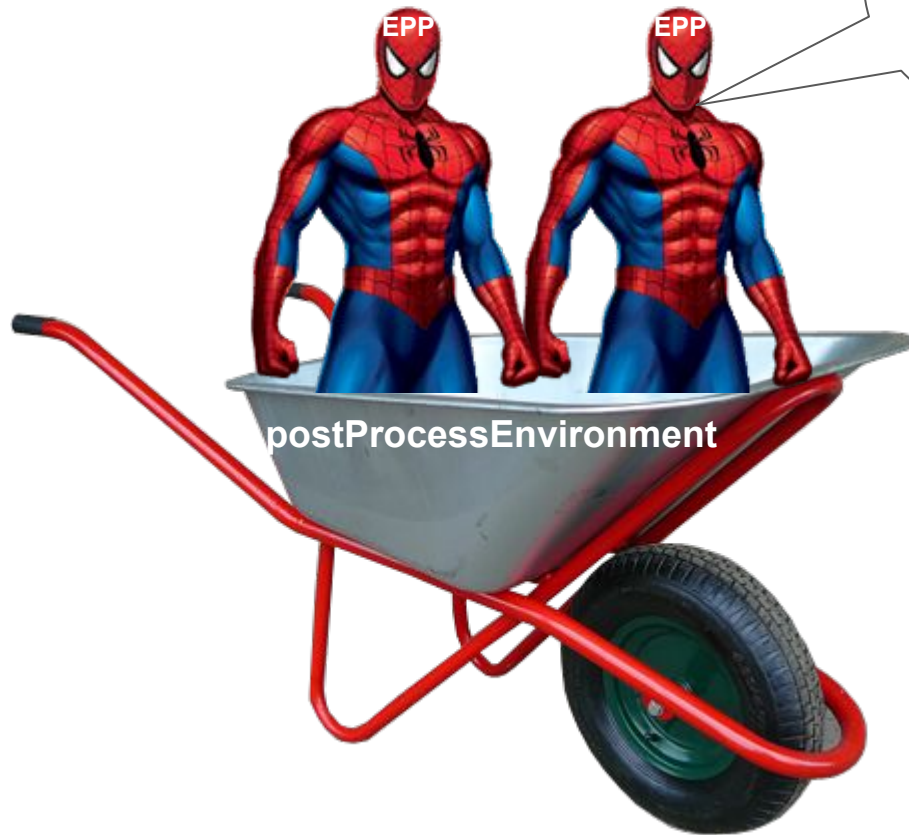
# onApplicationEnvironmentPreparedEvent

ща, отсортирую и  
поедем...



# onApplicationEnvironmentPreparedEvent

Как то он  
на нас  
ПОХОЖ...



# onApplicationEnvironmentPreparedEvent



Я с вами  
пацаны...





onApplicationEnvironmentPreparedEvent



# ConfigFileApplicationListener

Слушает

- `ApplicationPreparedEvent`
- `ApplicationEnvironmentPreparedEvent`

Загружает

- `application.yml`
- `application.properties`
- `env vars`
- `cmd args`



# ConfigFileApplicationListener

Слушает

- `ApplicationPreparedEvent`
- **`ApplicationEnvironmentPreparedEvent`**

Загружает

- `application.yml`
- `application.properties`
- `env vars`
- `cmd args`



# А я слышала

про другие события!

`ContextStartedEvent`

`ContextStoppedEvent`

`ContextClosedEvent`

`ContextRefreshedEvent`



Это Spring Boot,  
Тут больше event-ов



# Application Events

ApplicationStartingEvent

ApplicationEnvironmentPreparedEvent

ApplicationPreparedEvent

ContextRefreshedEvent

EmbeddedServletContainerInitializedEvent

ApplicationReadyEvent


ApplicationFailedEvent


...

# Application Events

ApplicationStartingEvent 

ApplicationEnvironmentPreparedEvent 

ApplicationPreparedEvent 

ContextRefreshedEvent 

EmbeddedServletContainerInitializedEvent 

ApplicationReadyEvent 

ApplicationFailedEvent 

ContextClosedEvent 

# Application Events

ApplicationStartingEvent

ApplicationEnvironmentPreparedEvent

ApplicationPreparedEvent



ContextRefreshedEvent

EmbeddedServletContainerInitializedEvent

ApplicationReadyEvent

ApplicationFailedEvent

ContextClosedEvent





# Application Events

ApplicationStartingEvent

ApplicationEnvironmentPreparedEvent

ApplicationPreparedEvent



ContextRefreshedEvent

EmbeddedServletContainerInitializedEvent

ApplicationReadyEvent

ApplicationFailedEvent

ContextClosedEvent



# Application Events

ApplicationStartingEvent

ApplicationEnvironmentPreparedEvent

ApplicationPreparedEvent

ContextRefreshedEvent

EmbeddedServletContainerInitializedEvent

ApplicationReadyEvent

ApplicationFailedEvent

ContextClosedEvent



# А это, забыли?

`ContextStartedEvent`

`ContextStoppedEvent`



# Найди их место

ApplicationStartingEvent



ApplicationEnvironmentPreparedEvent



ApplicationPreparedEvent



ContextRefreshedEvent



EmbeddedServletContainerInitializedEvent



ApplicationReadyEvent



ApplicationFailedEvent

ContextClosedEvent

**ContextStartedEvent**

**ContextStoppedEvent**



Эти ивенты

никогда не работают

А ты их вызывать не  
пробовал?



# А ты их вызывать не пробовал?

`ctx.start();` → `ContextStartedEvent`

`ctx.stop();` → `ContextStoppedEvent`



## Где вылетит эксепшен

`ctx.stop();` (1)

`ctx.start();` (2)

`ctx.close();` (3)

`ctx.start();` (4)



## Где вылетит эксепшен

`ctx.stop();` (1)

`ctx.start();` (2)

`ctx.close();` (3)

`ctx.start();` (4)

1. на строчке (1)

2. на строчке (2)

3. на строчке (3)

4. на строчке (4)

## Где вылетит эксепшен

`ctx.stop();` (1)

`ctx.start();` (2)

`ctx.close();` (3)

`ctx.start();` (4)

1. на строчке (1)

2. на строчке (2)

3. на строчке (3)

4. на строчке (4)

5. не вылетит вообще

## Где вылетит эксепшен

`ctx.stop();` (1)

`ctx.start();` (2)

`ctx.close();` (3)

`ctx.start();` (4)



**на строчке (4)**

Stop before Start



## Где вылетит эксепшен

`ctx.stop();` (1)

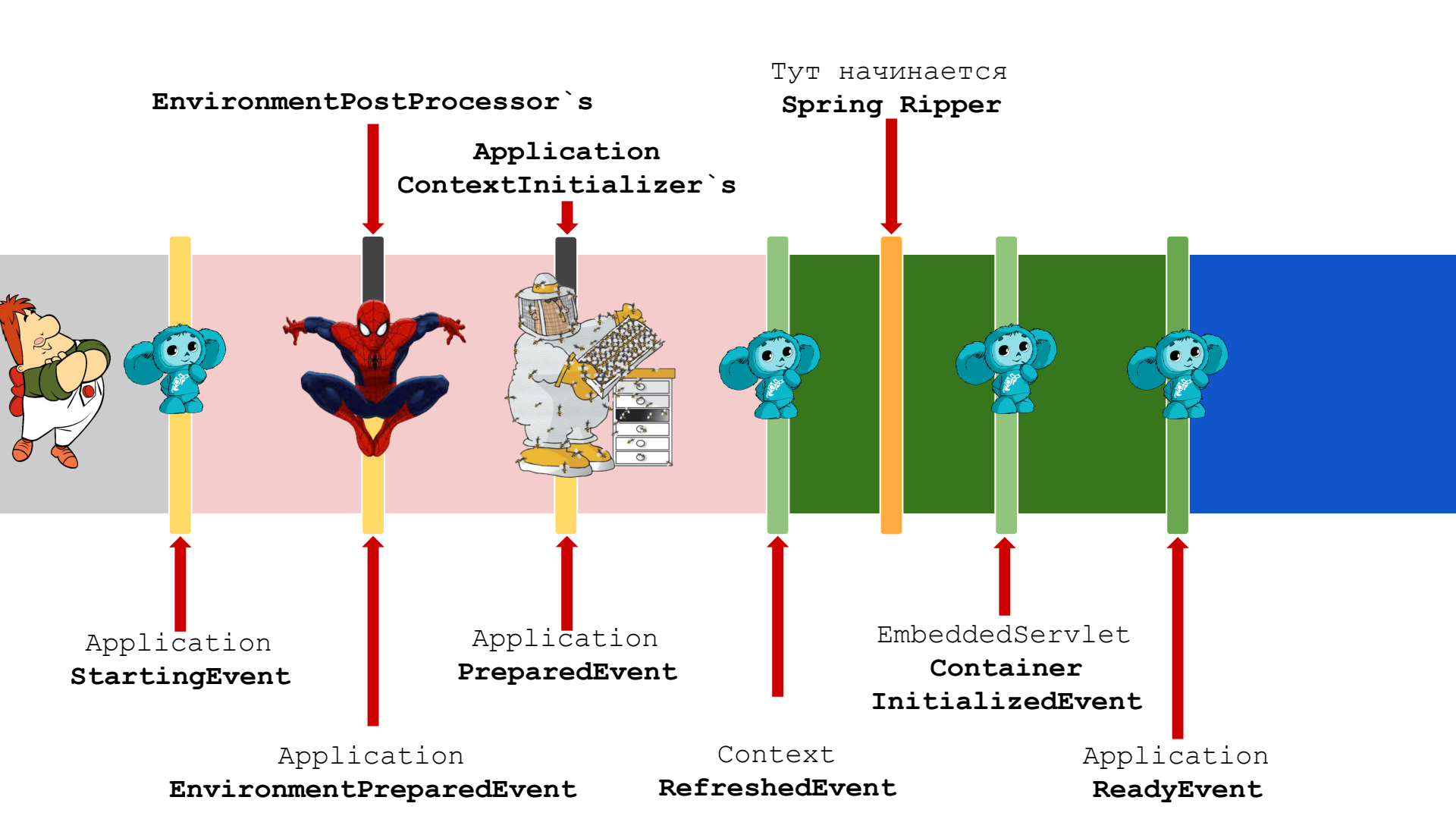
`ctx.start();` (2)

**`ctx.close();`** (3)

`ctx.start();` (4)



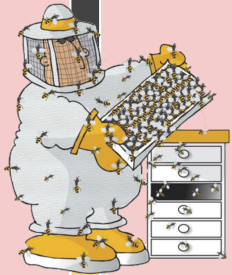
на строчке (4)



**EnvironmentPostProcessor`s**

Тут начинается  
**Spring Ripper**

**Application  
ContextInitializer`s**



Application  
**StartingEvent**

Application  
**EnvironmentPreparedEvent**

Application  
**PreparedEvent**

Context  
**RefreshedEvent**

EmbeddedServlet  
**Container  
InitializedEvent**

Application  
**ReadyEvent**

# Как запустать

1. `tomcat war`
2. `idea`
3. `java -jar/war`



# Кто пакует Jar со всеми зависимостями

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



Јар как Јар, но если зайти внутрь...



# Анатомия SpringBoot Jar

jar

| - META-INF

| - BOOT-INF

| - libs

| - classes

| - org

| - springframework

| - boot



# Анатомия SpringBoot Jar

jar

|- META-INF



**MANIFEST.MF**

|- BOOT-INF

  |- libs

...

  |- classes

Spring-Boot-Version: 1.5.3.RELEASE

|- org

Implementation-Vendor: Pivotal Software, Inc.

  |- springframework

Main-Class:

    |- boot

org.springframework.boot.loader.JarLauncher

Start-Class:ru...ripper.OurMainClass

Spring-Boot-Classes: BOOT-INF/classes/

Spring-Boot-Lib: BOOT-INF/lib/

...

# Анатомия SpringBoot Jar

```
jar
|- META-INF → MANIFEST.MF
|- BOOT-INF
  |- libs
  |- classes
|- org
  |- springframework
    |- boot
      ...
      Spring-Boot-Version: 1.5.3.RELEASE
      Implementation-Vendor: Pivotal Software, Inc
      Main-Class:
      org.springframework.boot.loader.JarLauncher
      Start-Class: ru...ripper.OurMainClass
      Spring-Boot-Classes: BOOT-INF/classes/
      Spring-Boot-Lib: BOOT-INF/lib/
      ...
```

# Анатомия SpringBoot Jar

```
jar
|- META-INF → MANIFEST.MF
|- BOOT-INF
  |- libs
  |- classes
|- org
  |- springframework
    |- boot
      ↓
      JarLauncher
```

...

Spring-Boot-Version: 1.5.3.RELEASE

Implementation-Vendor: Pivotal Software, Inc.

**Main-Class:**  
org.springframework.boot.loader.JarLauncher

Start-Class: ru...ripper.OurMainClass

Spring-Boot-Classes: BOOT-INF/classes/

Spring-Boot-Lib: BOOT-INF/lib/

...

А кто прописывает манифест этому джару?



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

**MANIFEST.MF**

...

Start-Class: conference.spring.boot.ripper.OurMainClass

...

# Как выбирается mainClass если его не указать

Spring boot plugin сканирует проект – ищет мейны

- если есть только один – то это он :)
- если больше одного – смотрит где стоит **@SpringBootApplication** и выбирает его
- если **@SpringBootApplication** нет или >1 – ошибка о множественных main при сборке



A close-up photograph of two men dressed in heavy, dark fur clothing, likely representing a primitive or survivalist setting. They are standing in a snowy, overcast environment. The man on the left has a thick, reddish-brown beard and is looking slightly to the right. The man on the right has dark hair and a mustache, looking towards the camera with a serious expression. Both men have white thought bubbles above their heads. The background is a bright, overcast sky with some snow on the ground.

java -jar  
- слишком сложно для нас

Вот если бы можно:  
**жмя и запустилось...**

# Хочу executable jar: Maven

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

## Gradle

```
springBoot {
  executable = true
}
```

Demo. Executable jar

# Внутренний мир executable Jar

- Бабушка хочет чтобы **жмя** и работало
- windows “click click”
- `$ ./app.jar`
- ...

Demo. Executable jar structure

# Как выглядит Jar?

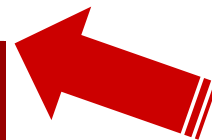
Script

**0xf4ra -> а это ZIP!**  
jar archive

# Как выглядит Jar?

Script

**0xf4ra -> а это ZIP!**  
jar archive

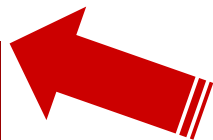


Начало файла

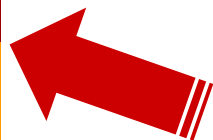
# Как выглядит Jar?

Script

**0xf4ra -> а это ZIP!**  
jar archive



Начало файла



Начало zip архива



## Немного выводов

The background is a detailed steampunk illustration. It depicts a city or industrial complex built upon a map of a landmass. The architecture is heavily mechanical, featuring brass, copper, and iron tones. There are large gears, pipes, and structures that resemble a factory or a power plant. A winding river or canal flows through the city. In the foreground, large, ornate letters spelling out 'DIN' are visible on a tiled floor. The overall atmosphere is one of intricate mechanical engineering and industrial revolution.

1. **Spring Boot** вышел за пределы **Spring Context**
2. Starter от команды **Spring** отличаются от public convention не всегда в лучшую сторону
3. в **Spring Boot** это целый мир, в котором нет магии

Я вижу, это  
Ваш первый  
визит к  
стоматологу ?




 @tolkv

 @lavcraft



@jekaborisov 

@jeka1978 



Вопросы?

