Spring 4

Евгений Борисов

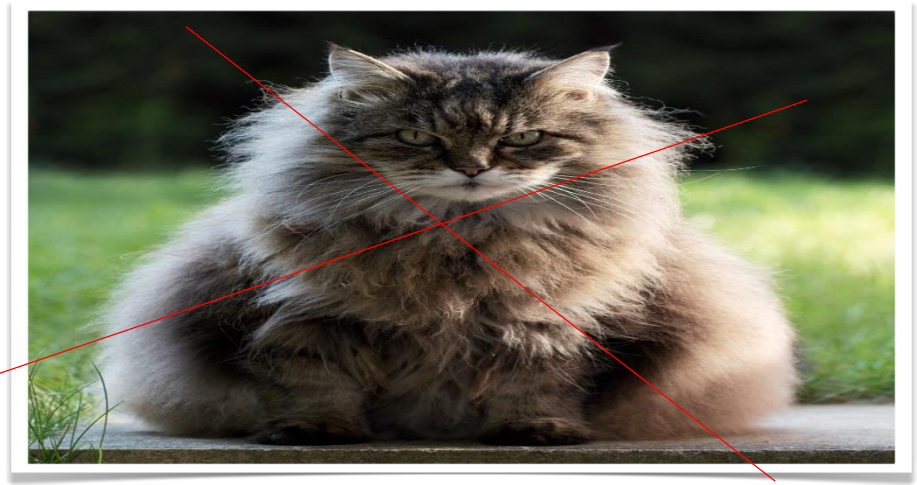
bsevgeny@gmail.com


THE SINGLETON


THE SINGLETON
12

Анти-Шипилёвский слайд и не только, а-ля «дисклеймер»

- В этом докладе **нет** боли, крови, кишок и расчленёнки
- Но в этом докладе не будет розовых котиков и пушистых пони



- Более того, в этом докладе нету даже Боромира


НЕЛЬЗЯ ПРОСТО ТАК...
Да ты уже достал всех

В данном докладе будут использованы

- Gradle 1
- Groovy 2
- Spring 4
- Java 8
- IntellJ ~~16~~

В данном докладе будут использованы

- Gradle 1
- Groovy 2
- Spring 4
- Java 8
- IntellJ 13

Сегодня в программе:

- Массовые зачистки которые устроил спринг 4
- Новые фишки из JEE 7 которые будут поддерживаться
- Полная поддержка для новых фишек в Java 8
- Новые фишки в спринг 4
- Интеграция Спринга и Груви
- **И наконец…**

Смерть XML-у


ГДЕ ВСЕ НАШИ УСТАРЕВШИЕ БИБЛИОТЕКИ?
В ПЕЧКУ ИХ!!!

Спринг 4 начинает новую точку отсчёта

- **В спринге**:
  - Выброшены все устаревшие библиотеки
  - Выброшены почти все устаревшие методы
- **Для джавы поддерживается только**:
  - Java SE 6+
  - Java EE 6+ (Servlet 3.0 focused, Servlet 2.5 compatible)
- **Для 3-d party библиотек**
  - Поддержка с 2010 года и старше
  - Например Hibernate 3.6+, Quartz 1.8+, EhCache 2.1+

Что новое из JEE поддерживается?

- **JMS 2.0**
- **JTA 1.2**
- **JPA 2.1**
- **Bean Validation 1.1**
- **JSR-236 Concurrency Utilities**
- **JSR-107 JCache**

Благодаря лямбда-выражениям и указателям на методы…

- **JdbcTemplate, JmsTemplate, TransactionTemplate… будут лучше**

До **После**

Давайте посмотрим как это выглядит на деле

JSR 310

- `@DateTimeFormat(pattern="M/d/yy h:mm")`
  `private LocalDateTime dateTimeWithFormat;`

- `@DateTimeFormat(iso= DateTimeFormat.ISO.DATE)`
  `private LocalDateTime dateTimeWithFormat;`

- Будет работать для Spring MVC, SpEL и.т.п

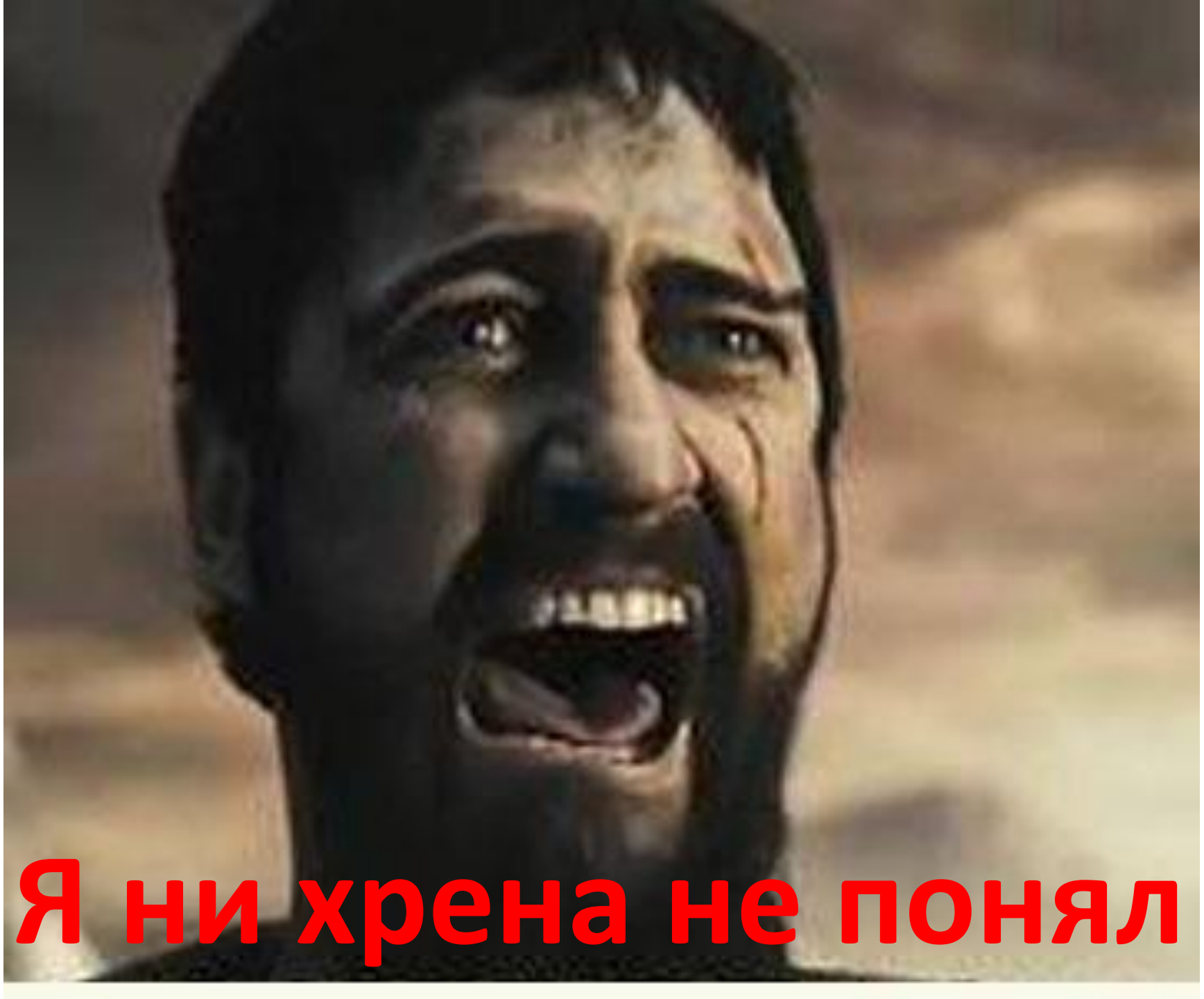
Повторные аннотации

- В джаве 8 появилась новая мета аннотация: `@Repeatable`
- Все аннотации являющиеся `@Repeatable` можно поставить несколько раз над одним и тем же типом

Мминуточку, а что будет возвращать getAnnotation когда их будет несколько?

Повторные аннотации

- Каждая повторная аннотация указывает другую аннотацию, которая в себе держит лист первых аннотаций
- И в Runtime, повторная аннотация, приведет к созданию хандлера, который будет держать в себе лист этих аннотаций


Я ни хрена не понял

Что можно вытащить через Reflection?

- Метадату класса
- Метадату филда
- Метадату метода
- Метадату конструктора
  - Annotations, Modifiers, Exception types
  - Тип параметров
- А можно ли вытащить название параметров?

Тут надо сначала показать фокус…

Новые фишки в спринге

- Вот тут то мне карты и пошла…

Порядок впрыскивания

- Когда лист определялся через XML порядок бинов в нём можно было контролировать.
- А что будет, если:
- ```
  @Autowired
  private List<Quoter> quoters;
  ```
- В лучшем случае по алфавиту
- А вот Спринг 4 решил эту проблему
  ```

Кастомные аннотации

Кастомные аннотации

- Можно писать свои аннотации, которые «наследуют от спринговых»
- Вот только пользоваться их параметрами не получится.
- Можно конечно определить свои, такие же, но толку не будет
- **Спринг 4 это изменил**

Кастомные аннотации

- ```
  @Retention(RetentionPolicy.RUNTIME)
  @Transactional
  public @interface MyTransactional {
      Propagation propagation() default Propagation.REQUIRES_NEW;
  }
  ```

- ```
  @MyTransactional
  public class MyService {…}
  ```

Впрыскивание с учётом дженерика

- ```
  @Component
  public class MyRepository<Customer> implements
  Repository {
  }
  ```

- ```
  @MyTransactional
  public class MyService {
      @Autowired
      private Repository<Customer> repository;
  }
  ```

@Conditional

- Предоставляет более гибкую и динамическую модель определения бинов
- Наподобие профилей из Spring 3.1, но лучше
- Может быть использован для умных значений по умолчанию

Игра

Обычный режим

Режим Бога

Спринг и Груви – братья навек

Что вы думаете про бины написанные на Груви


А в 4 Спринге они АОР
для груви починили…

Почему ты пишешь бины на груви?

- Потому что я могу
- Потому что так быстрее и проще
- Потому что если бин груви завернуть в скрипт, то он может динамически перезагружаться
-
```
<beans>
    <lang:groovy refresh-check-delay="1000"
                 id="calculator"
          script-source="classpath:calculator.groovy"/>
</beans>
```


Смерть XML-ам!!!

Чем был хорош XML

- Можно менять конфигурацию, не перекомпилируя
- Не программисты могли с ним работать
- Очень удобен для delivery-ориентированных компаний

Конфигурация на Груви

- Можно не только менять конфигурацию не перекомпилируя, но и не перезагружая сервер
- Груви скрипт для не програмистов можеть быть не сложнее xml-а
- Еще более удобно для delivery-ориентированных компаний
- Крайне удобно для тестирования и разработки

Как он будет выглядеть?

- beans = {

    bookRepository(BookRepositoryImpl)

    bookStore(BookStoreImpl) {
      repository = ref(bookRepository)
    }
  }

- `new GroovyScriptApplicationContext("classpath:context.groovy");`

Что есть сегодня?

- Есть Spring 4 – milestone 3
- Java 8 developer preview
- Есть intellIJ 13 - early access
- Gradle 1.8 Не все тесты будут работать
- Groovy 2.1

Так что на спринг 4 переходить?