#### JVM原理手记

##### 之注解：绑定配置项

问题描述

Jvm 1.8

Spring5支持全注解配置，将注解（Annotation）这一历史悠久的Java特性拉回了时尚的舞台

示例代码演示了一个用于绑定配置项的注解@ParamTemplate

解决方案

/\*

\* 元注解@Target,@Retention,@Documented,@Inherited

\*

\* @Target 表示该注解用于什么地方，可能的 ElemenetType 参数包括：

\* ElemenetType.CONSTRUCTOR 构造器声明

\* ElemenetType.FIELD 域声明（包括 enum 实例）

\* ElemenetType.LOCAL\_VARIABLE 局部变量声明

\* ElemenetType.METHOD 方法声明

\* ElemenetType.PACKAGE 包声明

\* ElemenetType.PARAMETER 参数声明

\* ElemenetType.TYPE 类，接口（包括注解类型）或enum声明

\*

\* @Retention 表示在什么级别保存该注解信息。可选的 RetentionPolicy 参数包括：

\* RetentionPolicy.SOURCE 注解将被编译器丢弃

\* RetentionPolicy.CLASS 注解在class文件中可用，但会被VM丢弃

\* RetentionPolicy.RUNTIME VM将在运行期也保留注释，因此可以通过反射机制读取注解的信息。

\*

\* @Documented 将此注解包含在 javadoc 中

\*

\* @Inherited 允许子类继承父类中的注解

\*

\*/

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

@Documented

@Inherited

public @interface ParamTemplate {

public String key();

}

笔记

1. 使用Class.getDeclaredFields()方法获取注解
2. 使用Field.isAnnotationPresent()方法检查注解是否存在
3. 使用|Field.getAnnotation()方法获得注解

□