spring学习笔记-测试工具概述

|  |  |
| --- | --- |
|  | 🢂 内容概览 |
|  | Why：此文档用来做什么？它存在的意义是什么？为解决什么问题？   |  | | --- | |  |   What：当前包含了那些内容？   |  | | --- | |  |   How：此文档应如何参考？   |  | | --- | |  |   Who：此文档适用于那些人员阅读参考？   |  | | --- | |  |   Summary：摘要   |  | | --- | |  |   Reference：参考文献   |  | | --- | |  | |

目录

[1 简介 4](#_Toc463777027)

[1.1 mock对象 4](#_Toc463777028)

[1.1.1 Environment 4](#_Toc463777029)

[1.1.2 JNDI 4](#_Toc463777030)

[1.1.3 Servlet 4](#_Toc463777031)

[1.1.4 Portlet 4](#_Toc463777032)

[1.2 测试工具 4](#_Toc463777033)

[1.2.1 通用测试工具 4](#_Toc463777034)

[1.2.2 spring mvc 4](#_Toc463777035)

[1.3 JDBC相关测试 4](#_Toc463777036)

[2 测试注解 5](#_Toc463777037)

[2.1 基本目标 5](#_Toc463777038)

[2.2 context相关注解 5](#_Toc463777039)

[2.2.1 @ContextConfiguration 5](#_Toc463777040)

[2.2.2 @WebAppConfiguration 5](#_Toc463777041)

[2.2.3 @ContextHierarchy 6](#_Toc463777042)

[2.2.4 @ActiveProfiles 6](#_Toc463777043)

[2.2.5 @TestPropertySource 7](#_Toc463777044)

[2.2.6 @DirtiesContext 7](#_Toc463777045)

[2.2.7 @TestExecutionListeners 8](#_Toc463777046)

[2.3 JDBC相关注解 8](#_Toc463777047)

[2.3.1 @Commit 8](#_Toc463777048)

[2.3.2 @Rollback 9](#_Toc463777049)

[2.3.3 @BeforeTransaction 9](#_Toc463777050)

[2.3.4 @AfterTransaction 9](#_Toc463777051)

[2.3.5 @Sql 9](#_Toc463777052)

[2.3.6 @SqlConfig 9](#_Toc463777053)

[2.3.7 @SqlGroup 10](#_Toc463777054)

[2.4 标准注解的支持 10](#_Toc463777055)

[2.5 JUnit相关注解 11](#_Toc463777056)

[2.5.1 @IfProfileValue 11](#_Toc463777057)

[2.5.2 @ProfileValueSourceConfiguration 11](#_Toc463777058)

[2.5.3 @Timed  11](#_Toc463777059)

[2.5.4 @Repeat  11](#_Toc463777060)

[2.6 定制注解 11](#_Toc463777061)

# 简介

spring提供了一些列用于简化测试的工具类，放在spring-test包中。Spring-test包含两部分：mock对象（org.springframework.mock）和测试工具类（org.springframework.test）。

## mock对象

存放在org.springframework.mock包下；

### Environment

org.springframework.mock.env.MockEnvironment;

org.springframework.mock.env.MockPropertySource;

### JNDI

org.springframework.mock.jndi

### Servlet

org.springframework.mock.web

### Portlet

org.springframework.mock.web.portlet

## 测试工具

### 通用测试工具

org.springframework.test.util提供了单元测试和集成测试通用的工具类；

1. org.springframework.test.util.ReflectionTestUtils基于反射的测试工具，用于访问和修改修改常量值、非public的属性值，调用非public的setter、getter、普通方法等；
2. org.springframework.test.util.AopTestUtils提供AOP相关的测试方法；

### spring mvc

1. org.springframework.test.web package contains ModelAndViewAssert用于测试spring-mvc的ModelAndView对象；

## JDBC相关测试

1. org.springframework.test.jdbc.JdbcTestUtils提供了一系列JDBC测试相关的工具类，比如：统计一张表的记录数；
2. org.springframework.test.context.junit4.AbstractTransactionalJUnit4SpringContextTests提供基于junit4的JDBC测试工具；
3. org.springframework.test.context.testng.AbstractTransactionalTestNGSpringContextTests提供基于testng的JDBC测试工具；

# 测试注解

## 基本目标

1. 在测试时管理IOC容器缓存数据；
2. 为测试对象提供依赖注入；
3. 为集成测试提供合适的事务管理；
4. 提供spring相关的工具类，便于用例书写；

## context相关注解

### @ContextConfiguration

1. 类注解，在org.springframework.test.context包下；
2. 用于指定测试时，容器需要加载的spring配置；可以接收两种方式参数：xml和使用@Configuration标注的类；示例如下：

|  |
| --- |
|  |

1. 可以指定ApplicationContextInitializer的实现类，定制容器的初始化行为；

|  |
| --- |
|  |

1. 可以指定ContextLoader的实现类，定制容器的加载策略；

|  |
| --- |
|  |

### @WebAppConfiguration

1. 类注解，在org.springframework.test.context.web包下；
2. 用于指定使用WebApplicationContext作为容器上下文对象；默认使用：file:src/main/webapp作为web应用的根路径；

|  |
| --- |
|  |

1. 可以通过value属性显式指定web应用的根路径；路径可是是类路径（classpath:），也可是是文件路径（file:）；

|  |
| --- |
|  |

1. @WebAppConfiguration必须和@ContextConfiguration一起使用；

### @ContextHierarchy

1. 类注解，在org.springframework.test.context包下；
2. 用于指定多个Context之间的继承关系；一个@ContextHierarchy可以包含一个或多个@ContextConfiguration；

|  |
| --- |
|  |

### @ActiveProfiles

1. 类注解，在org.springframework.test.context包下；
2. 用于指定测试时需要使用的profile;

|  |
| --- |
|  |

### @TestPropertySource

1. 类注解，在org.springframework.test.context包下；
2. 用于加载properties文件，优先级高于使用@PropertySource；

|  |
| --- |
|  |

1. 也可以直接指定资源文件值，优先级高于从properties加载；

|  |
| --- |
|  |

### @DirtiesContext

1. 类注解&方法注解，在org.springframework.test.annotation包下；
2. 用于标记当前Context已经因为测试而存在脏数据，不可以再使用；Spring容器将会从测试缓存中清理脏Context，并基于元数据重建Context；
3. 在执行测试类前，标记；

|  |
| --- |
|  |

1. 在测试执行后，标记；（默认）

|  |
| --- |
|  |

1. 在测试类的每个方法执行后，标记；

|  |
| --- |
|  |

1. 在方法执行前，标记；

|  |
| --- |
|  |

1. 在方法执行后，标记；（默认）

|  |
| --- |
|  |

1. 如果@DirtiesContext标记的context是@ContextHierarchy标记context中的一个，则默认使用穷举算法，不仅清理当前标记的context，还有所有具有共同父context的子context；可以通过hierarchyMode指定清理context的策略；详见DirtiesContext.HierarchyMode 文档；

|  |
| --- |
|  |

### @TestExecutionListeners

1. 类注解，在org.springframework.test.context包下；
2. 指定当前Context需要的TestExecutionListeners监听，和@ContextConfiguration配合使用；

|  |
| --- |
|  |

## JDBC相关注解

### @Commit

1. 类注解&方法注解，在org.springframework.test.annotation包下；
2. 指定在执行完测试方法后，提交事务，等价于@Rollback(false)；

|  |
| --- |
|  |

### @Rollback

1. 类注解&方法注解，在org.springframework.test.annotation包下；
2. 用于标记执行完测试方法后，是否要回滚，默认回滚；

|  |
| --- |
|  |

### @BeforeTransaction

1. 方法注解，在org.springframework.test.context.transaction包下；
2. 用于指定在测试方法事务开始前，要执行的方法；

|  |
| --- |
|  |

### @AfterTransaction

1. 方法注解，在org.springframework.test.context.transaction包下；
2. 用于指定在测试方法事务结束后，要执行的方法；

|  |
| --- |
|  |

### @Sql

1. 方法注解，在org.springframework.test.context.jdbc包下；
2. 指定在测试期间，需要执行的SQL脚本；

|  |
| --- |
|  |

### @SqlConfig

1. 方法注解，在org.springframework.test.context.jdbc包下；
2. 指定执行@Sql时，需要的配置元数据；

|  |
| --- |
|  |

### @SqlGroup

1. 方法注解，在org.springframework.test.context.jdbc包下；
2. 用于在一个测试类上同时指定多个@Sql；

|  |
| --- |
|  |

## 标准注解的支持

1. spring测试类支持如下注解；
2. spring测试类不支持bean声明周期相关的注解，比如：@PostConstruct和@PreDestory等，推荐使用测试框架自己的生命周期控制注解代替；

|  |
| --- |
|  |

## JUnit相关注解

spring中与JUnit相关的注解只能用在JUnit相关环境中；

### @IfProfileValue

### @ProfileValueSourceConfiguration

### @Timed

### @Repeat

## 定制注解

spring中可以使用如下注解作为元数据进行测试注解定制；

|  |
| --- |
| @ContextConfiguration  @ContextHierarchy  @ActiveProfiles  @TestPropertySource  @DirtiesContext  @WebAppConfiguration  @TestExecutionListeners  @Transactional  @BeforeTransaction  @AfterTransaction  @Rollback  @Sql  @SqlConfig  @SqlGroup  @Repeat  @Timed  @IfProfileValue  @ProfileValueSourceConfiguration |

比如：

|  |
| --- |
|  |

创建一个定制标签：

|  |
| --- |
|  |

使用定制标签：

|  |
| --- |
|  |