

存储过程	(6)
JPA	(23)
HIBERNATE	(79)
JSONP	(17)
事务	(36)
游标	(3)
反射机制	(24)
JXL与POI	(31)
REST	(11)
WEBSERVICE	(45)
C C++	(2)
HTML	(11)
HttpURLConnection	(10)
JNI	(1)
HTTP请求	(17)
TOMCAT	(28)
SERVLET	(25)
JDK JRE	(3)
JSP标签	(9)
IO	(8)
资源文件	(5)
定时任务调度	(29)
COOKIE	(4)
IFRAME	(5)
浏览器兼容性	(16)
动态代理	(7)
设计模式	(32)
SPRING-MVC	(114)
SPRING模板类	(20)
批量处理	(5)
JDBC	(4)
JAX-WS	(3)
ECLIPSE	(16)
EXE4J	(1)
多线程	(37)
算法	(22)
编码	(16)
分页	(2)
SOCKET	(2)
闭包	(1)
SQL HQL语句	(23)
TEXTAREA显示	(5)
性能优化	(1)
DISCUZ	(1)
JERSEY	(6)
面试	(49)
CLOB与BLOB	(6)
HTTPCLIENT	(44)
SMB	(3)
JCIFS	(2)
SVN	(6)
断点续传	(1)
EJB	(3)
文件预览	(4)

6. 通过JUnit 4 执行

右键方法名，选择则 “Run As” → “JUnit Test” 即可

附录1：整体测试类文件



```
/* @(#) UserDaoTest.java
 *
 */
package com.phj.dao;

import javax.annotation.Resource;

import org.junit.Test;
import org.junit.runner.RunWith;

import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.AbstractJUnit4SpringContextTests;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.phj.entity.User;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = "classpath:applicationContext.xml")
public class UserDaoTest extends AbstractJUnit4SpringContextTests {

    @Resource
    private UserDaoInterface userDao;

    @Test
    public void saveTest() {

        User user1 = new User();
        user1.setUsername("tom");
        user1.setPassword("123456");
        user1.setNickName("tom");
        user1.setEmail("tom@gmail.com");

        User user2 = new User();
        user2.setUsername("admin");
        user2.setPassword("123456");
        user2.setNickName("admin");
        user2.setEmail("admin@admin.com");

        User user3 = new User();
        user3.setUsername("feihong");
        user3.setPassword("123456");
        user3.setNickName("phj");
        user3.setEmail("test@gmail.com");

        userDao.save(user1);
        userDao.save(user2);
        userDao.save(user3);
    }
}
```

OK，到此为止就可以使用Spring的测试框架了。

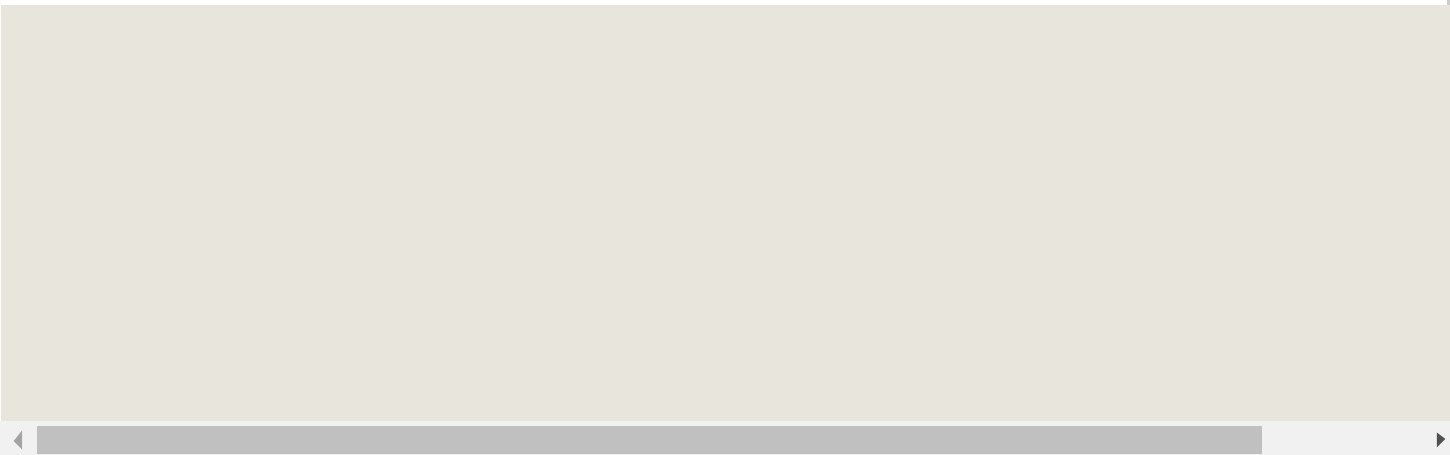
- 发送邮件 (4)
- NIO (32)
- SPRING-DATA-JPA (7)
- EL表达式 (7)
- OGNL表达式 (3)
- 读取配置文件 (5)
- 打印页面 (2)
- 下载文件 (3)
- SQL SERVER (5)
- DOM4J (7)
- Apusic金蝶 (4)
- 上传文件 (17)
- 其他 (2)
- 读写文件 (13)
- GOOGLE CHART (1)
- 匿名函数 (1)
- DBLINK (4)
- JACOB (1)
- IBATIS/MYBATIS (59)
- MAVEN (82)
- JAVA 打包 (3)
- AJAX (7)
- PUSHLET (7)
- COMET (5)
- 过滤器&监听器 (5)
- DWR (1)
- JVM (18)
- XSL (5)
- VELOCITY (2)
- LINUX (23)
- JMS (13)
- LDAP (1)
- JOTM (3)
- JTA (4)
- MongoDB&BSON (25)
- UML (2)
- LOGBACK (6)
- 安全方面 (11)
- 验证码 (1)
- CXF (7)
- JSTL (4)
- 短信猫 (1)
- ZIP解压压缩 (4)
- BAT批处理命令 (1)
- RMI (1)
- JSR-VAlID (3)
- 自定义注解 (16)
- TILES (1)
- ITEXT-PDF (2)
- 命令行 (1)
- 进制转换 (5)
- IDEA (21)
- java 异常处理 (10)
- 泛型 (20)

使用注解时,Junit报错:

```
[java]
01. java.lang.NoSuchFieldError: NULL
02.     at org.junit.runners.ParentRunner.<init>(ParentRunner.java:48)
03.     at org.junit.runners.BlockJUnit4ClassRunner.<init>(BlockJUnit4ClassRunner.java:59)
04.     at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.<init>
      (SpringJUnit4ClassRunner.java:104)
05.     at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
06.     at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccesso
07.     at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructo
08.     at java.lang.reflect.Constructor.newInstance(Constructor.java:513)
09.     at org.junit.internal.requests.ClassRequest.buildRunner(ClassRequest.java:33)
10.     at org.junit.internal.requests.ClassRequest.getRunner(ClassRequest.java:28)
11.     at org.eclipse.jdt.internal.junit4.runner.JUnit4TestReference.<init>
      (JUnit4TestReference.java:28)
```

经检查是jar包冲突,我在建项目的时候,myeclipse有自动加入junit4,而后面我又自己加入Junit4.8.1。

如果报以下错误，则是junit包版本太低，需要4.5或以上的版本



其他问题：

@ContextConfiguration({"classpath:applicationContext.xml","classpath:spring/buyer/applicationContext-service.xml"})

导入配置文件，这里我的applicationContext配置文件是根据模块来分类的。如果有多个模块就引入多个“applicationContext-service.xml”文件。如果所有的都是写在“applicationContext.xml”中则这样导入：

@ContextConfiguration(locations = "classpath:applicationContext.xml")

@RunWith(SpringJUnit4ClassRunner.class)SpringJUnit支持，由此引入Spring-Test框架支持！

@Transactional这个非常关键，如果不加入这个注解配置，事务控制就会完全失效！

@TransactionConfiguration(transactionManager = "transactionManager", defaultRollback = true)这里的事务关联到配置文件中的事务控制器（transactionManager = "transactionManager"），同时指定自动回滚（defaultRollback = true）。这样做操作的数据才不会污染数据库！

AbstractTransactionalDataSourceSpringContextTests要想构建这一系列的无污染纯绿色事务测试框架就必须找到这个基类！（即所有事务均不生效）

```
[java]
01. import com.wonders.frame.ias.service.IasScheduleTask;
02. import org.junit.Test;
03. import org.junit.runner.RunWith;
04. import org.springframework.stereotype.Component;
05. import org.springframework.test.context.ContextConfiguration;
06. import org.springframework.test.context.junit4.AbstractJUnit4SpringContextTests;
07. import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
08. import org.springframework.transaction.annotation.Transactional;
09.
10. import javax.annotation.Resource;
11.
12. /**
13.  * Created with IntelliJ IDEA.
14.  * User: zhoushun
15.  * Date: 2015/3/30
```

- OSGI (6)
- GRADLE (3)
- EXCEL导出 (26)
- JACKSON (22)
- JUNIT (8)
- 反向代理 (1)
- BONECP (1)
- JIT (2)
- GRAILS (3)
- GROOVY (10)
- WEBSOCKET (11)
- GIT (22)
- 调试工具 (2)
- MEMCACHE (10)
- 单元测试工具 (3)
- GC (18)
- REDIS (38)
- HESSIAN (5)
- MYSQL (77)
- 微信开发 (9)
- ZOOKEEPER (39)
- HIVE (11)
- HBASE (10)
- 原子变量 (13)
- SPRING-BOOT (16)
- 爬虫 (2)
- XPP3 (1)
- ASPECTJ (1)
- 抽奖概率 (8)
- 深入Java集合学习系列 (13)
- 序列化 (7)
- FREEMARKER (13)
- SITEMESH (5)
- XML解析-Digester (4)
- 页面静态化 (5)
- NODEJS/GRUNT (20)
- DROOLS (1)
- 条形码/二维码 (9)
- HADOOP (2)
- LUCENE (6)
- JSOUP (15)
- NGINX (7)
- JSBRI (0)
- JSBRIDGE (2)
- 并发集合 (17)
- EMOJI (9)
- HTML5 (10)
- LOMBOK (1)
- SPRING工具类 (11)
- guava (8)
- JAVA 8 (2)
- 加解密 (17)
- EAR (3)
- 水印 (1)

```
16.      * Time: 12:40
17.      * To change this template use File | Settings | File Templates.
18.      */
19.      @Component
20.      @Transactional
21.      @RunWith(SpringJUnit4ClassRunner.class)
22.      @ContextConfiguration(locations = "classpath*:applicationContext.xml")
23.      //public class IasTest extends AbstractTransactionalJUnit4SpringContextTests{
24.      public class IasTest extends AbstractJUnit4SpringContextTests {
25.          @Resource
26.          private IasScheduleTask task;
27.
28.          @Test
29.          public void taskTest(){
30.              task.urgeTask();
31.          }
32.      }
```

顶

2

踩

0

- ▲ 上一篇

使用POI导出EXCEL 并设置格式
- ▼ 下一篇

Servlet 生命周期、工作原理

相关文章推荐

- maven中使用spring的test包结合junit4进行测试。

• 【直播】计算机视觉原理及实战—屈教授

• @RunWith和 SpringJUnit4ClassRunner ---->ju...

• 【套餐】深度学习入门视频课程—唐宇迪

• 使用JUnit4测试Spring

• 【套餐】Hadoop生态系统零基础入门--侯勇蛟

• Spring整合JUnit4进行单元测试

• 【套餐】嵌入式Linux C编程基础--朱有鹏
- 使用Spring @Async异步执行方法的笔记

• 【套餐】2017软考系统集成项目——任铄

• SSM框架——详细整合教程（ Spring 4.0.2+Sprin...

• 【套餐】Android 5.x顶级视频课程——李宁

• JUnit4执行cases背后的故事(1)---JUnitCore源码...

• JUnit4执行cases背后的故事(2)---默认执行器Bloc...

• 如何搭建SSM(SpringMvc+mybatis+Spring)java...

• spring 3以及之后版本的异步方法


查看评论

暂无评论

发表评论

用户名：qq_36596145

评论内容：



- Thrift (13)
- Java并发包学习 (10)
- SonarQube (1)
- JAVA类加载 (3)
- REACT (19)
- SHIRO (4)
- SPEL (2)
- WEBDRIVER (2)
- PYTHON (3)
- 分布式锁 (7)
- Disruptor/LMAX (2)
- AngularJS (2)
- Vue.js (1)
- Netty (50)
- SOCKETIO (9)
- 消息MQ (2)
- ProtoBuf (12)
- TIF (0)
- SPRING-MVC原理分析 (31)
- ASM (2)
- TestNG (5)
- Mockito (3)
- LBS排序 (3)
- zookeeper&hbase (16)
- IO多路复用 (5)
- 架构 (1)
- cglib (4)
- Epoll (0)
- AIO (6)
- Dubbo (1)
- Jetty (10)
- 远程通讯 (1)
- RPC (1)
- SPI (4)
- CORS跨域 (8)
- 中文相关 (7)
- XPATH (3)
- Aviator (1)
- HtmlUnit (2)
- 埋点 (3)
- FTP、SFTP (7)
- Spring源码分析 (3)
- Spring-ws (4)
- Java对象引用 (8)
- XStream (2)
- JS原理 (1)
- WEBPACK (5)
- 枚举 (5)
- alibaba/fastjson (2)
- 动画效果 (3)
- 熔断降级 (2)
- SpringMvc异常处理 (5)
- SpringMvc实践 (2)
- SSM/SSH (2)

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

- ModelMapper (1)
- ES6/7 (3)
- Java线程专栏 (18)
- 线程中断 (9)
- 匿名内部类 (3)
- MVCC (3)
- Spring生命周期 (8)
- ElasticSearch (11)
- DB主从同步 (7)
- JAVA7-8 (1)
- Selenium (3)
- SPRING-TEST (5)
- RestTemplate (3)
- RedisTemplate (6)
- Mybatis 学习 (18)
- 数据库优化拆分 (13)
- Spring schema (1)
- Spring 线程池 (3)
- Spring 属性替换&转换 (7)
- Spring 注入&注解&扫描 (14)
- Spring Bean管理 (4)
- Spring (filter/listener/event) (9)
- Spring 事务 (12)
- Spring 定时任务 (5)
- Canvas (4)
- RequireJs (2)
- SPRING IOC源码 (1)
- 人脸/OPENCV/图像处理 (11)

文章存档	
2017年08月	(34)
2017年07月	(92)
2017年06月	(17)
2017年05月	(25)
2017年04月	(37)
展开	

阅读排行	
IntelliJ IDEA WEB项目的部署...	(126528)
SpringMVC——接收请求参数..	(121851)
解决Setting property 'sourc...	(108034)
HttpClient post 请求实例	(79872)
SpringMVC数据绑定全面示例...	(62493)
js中获取时间new Date()详细...	(57395)
SQL plus连接远程Oralce数据...	(56535)
[MyBatis]诡异的Invalid boun...	(52717)
HttpClient设置HTTP请求头He...	(52454)
读取Java文件到byte数组的三...	(50981)

评论排行	
解决Setting property 'sourc...	(49)
java实现附件预览 (openoffic...	(46)
Spring MVC @Transactiona...	(13)



微信关注CSDN
获得无限技术资源

 快速回复

 我要收藏

 返回顶部

SpringMVC——接收请求参数..	(11)
IntelliJ IDEA WEB项目的部署...	(11)
[MyBatis]诡异的Invalid boun...	(10)
Eclipse插件：Eclipse Color T...	(9)
interrupt、interrupted 、isIn...	(9)
Spring Data JPA入门	(8)
信息: TLD skipped. URI: http...	(7)

推荐文章

* CSDN日报20170725——《新的开始，从研究生到入职亚马逊》

* 深入剖析基于并发AQS的重入锁(Reentrant Lock)及其Condition实现原理

* Android版本的"Wannacry"文件加密病毒样本分析(附带锁机)

* 工作与生活真的可以平衡吗？

* 《Real-Time Rendering 3rd》提炼总结——高级着色：BRDF及相关技术

* 《三体》读后思考-泰勒展开/维度打击/黑暗森林

最新评论

interrupt、interrupted 、isInterrupted ...
sinat_1006：@Muzi_fly:我也有同感，不知道是否应该继续看了

maven 打包源代码 jar 与 jar-no-fork
SuperThreeX：看了一丢丢，我的理解是：执行source:jar这个 goal 时会先执行generate-sour...

IntelliJ IDEA WEB项目的部署配置
CAT_SEVEN：楼主 转载一下

HttpClient 4.3.6 使用MultipartEntityBuil...
al893422：服务端的怎么获取参数啊？

JAVA方法 字符串与unicode的相互转换
lzy20ls：for (int i = 0; i < str.length() - 6;) 这里6应该改成5

Grails完美解决一对多（One to many）...
阳光demo：楼主你好，最近在用 grails 语言开发项目，grails sdk 用的是 2.4 版本，...

Spring MVC @Transactional注解方式事...
小choi：实在是太感谢了！！！！

解决Setting property 'source' to 'org.e...
wuxunzan6022：解决了，但是报了一个新问题

关于如何使用websocket 接收并显示图片...
秋天不再有：可是大量的就不行了

深入Spring:自定义注解加载和使用
qq_36027146：受益匪浅

关闭

