

尼斯湖水鬼的博客

SpringBoot开发, SpringCloud开发, Java开发



RSS订阅

SpringBoot开发详解 (十) -- 使用JPA访问数据库下篇及使用Page进行数据分页

2017年06月04日 23:09:19

阅读数: 5115

JPA调用EntityManager实现

JPA调用EntityManager:

我们在上篇中使用过JPA其中一种方案来访问数据库，就是继承JpaRepository接口，今天我们使用另一种方案来实现JPA的访问数据库。所需要引用的依赖和上篇是一致的，包括实体类的创建在这就不写第二遍了，我们直接看DAO层如何实现。

首先，我们先写一个接口定义查询方法：

```
1 public interface JpaEntity {
2
3     List<Address> findAll();
4
5     Address findAddress(Long addressId);
6 }
```

然后我们再写一个实现类来实现这个接口：

```
1 @Repository
2 public class JpaEntityImpl implements JpaEntity {
3
4     @PersistenceContext
5     private EntityManager entityManager;
6
7     @Override
8     public List<Address> findAll() {
9         return this.entityManager.createQuery("select t from Address t",Address.class)
10            .getResultList();
11    }
12
13    @Override
14    public Address findAddress(Long addressId) {
15        return this.entityManager.createQuery("select t from Address t where t.addressId = ?"
16            .setParameter(1,addressId)
17            .getSingleResult();
18    }
19 }
```

之后我们先给实体类重写toString方法，便于后面结果的查看：

```
1 @Override
2 public String toString() {
3     return "Address{" +
4         "addressId=" + addressId +
5         ", userId=" + userId +
6         ", addressInfo='" + addressInfo + '\'' +
7         '}';
8 }
```

然后我们依旧通过单元测试来验证刚才写的代码的通过性：

```
1 @RunWith(SpringRunner.class)
2 @SpringBootTest
3 public class JpaEntityImplTest {
```

个人资料



尼斯湖水鬼

关注

原创 12 粉丝 185 喜欢 23 评论 43

等级: 博客 4 访问: 13万+

积分: 1090 排名: 4万+



oa系统

最新文章

- SpringBoot开发详解 (十二) -- SpringBoot中执行定时任务
- SpringBoot开发详解 (十一) -- Redis在SpringBoot中的整合
- SpringBoot开发详解 (九) -- 使用JPA访问数据库上篇
- SpringBoot开发详解 (八) -- 使用Swagger2构建API文档
- SpringBoot开发详解 (七) -- Mybatis整合Spring Boot

个人分类

spring boot 12篇

归档

- | | |
|---------|----|
| 2017年7月 | 2篇 |
| 2017年6月 | 1篇 |
| 2017年5月 | 5篇 |
| 2017年4月 | 3篇 |
| 2017年2月 | 1篇 |

热门文章

- SpringBoot开发详解 (五) --Controller接收参数以及参数校验
阅读量: 55581
- SpringBoot开发详解 (三) --SpringBoot配置文件YML注意事项
阅读量: 17424
- SpringBoot开发详解 (十二) -- SpringBoot中执行定时任务
阅读量: 12598

```
5    @AUTOWIRED
6    private JpaEntityImpl jpaEntity;
7
8    @Test
9    public void findAll() throws Exception {
10
11        List<Address> list = jpaEntity.findAll();
12        System.out.println(list.size());
13        for (Address address : list){
14            System.out.println(address.toString());
15        }
16    }
17
18    @Test
19    public void findAddress() throws Exception {
20        Address address = jpaEntity.findAddress(Long.valueOf(9));
21        System.out.println(address.toString());
22    }
23
24 }
```

通过测试大家可以看到，我们通过EntityManager访问数据库也是没问题的。

```
Run JpaEntityImplTest [FindAll]
JpaEntityImplTest [com.zpp.dao.ssm]
  ✓ findAll 82ms
    2017-06-04 22:03:18.707 INFO 1700 — [main] s.d.spring.web.cache.CachingAspect : Caching aspect applied for cache models with key com.zpp.pojo.Result(true)
    2017-06-04 22:03:18.707 INFO 1700 — [main] s.ds.schema.ModelContextKeyGenerator : Cache Key Generated: com.zpp.pojo.Result(true)
    2017-06-04 22:03:18.707 INFO 1700 — [main] s.d.spring.web.cache.CachingAspect : Caching aspect applied for cache
    2017-06-04 22:03:18.707 INFO 1700 — [main] s.d.spring.web.cache.OperationsKeyGenerator : Cache key generated: /xml/updateUser/{id}.com.zpp.controller.xmlController.updateUserDefaultGenericCachingStrategy
    2017-06-04 22:03:18.707 INFO 1700 — [main] s.d.spring.web.cache.CachingAspect : Caching aspect applied for cache operations with key /xml/updateUser/{id}.com.zpp.controller.xmlController.updateUserDefaultGenericCachingStrategy
    2017-06-04 22:03:18.709 INFO 1700 — [main] s.w.s.ClassAnnotationResourceGrouping : Group for method updateuser was xml-controller
    2017-06-04 22:03:18.709 INFO 1700 — [main] s.w.s.ClassAnnotationResourceGrouping : Group for method updateuser was xml-controller
    2017-06-04 22:03:18.709 INFO 1700 — [main] s.w.s.r.o.CachingOperationNameGenerator : Generating unique operation name
    updateuserUsingPUT_1
    2017-06-04 22:03:18.727 INFO 1700 — [main] com.zpp.dao.JpaEntityTables : Started JpaEntityImplTest in 5.967 seconds (JVM startup took 0.001 seconds)
    Hibernate: select address0_.address_id as address_id_0_ , address0_.add_info as add_info2_ , address0_.user_id as user_id3_ from tp_address address0_
    2 Address(addressId=1, userId=3, addressInfo='SH')
    2 Address(addressId=2, userId=3, addressInfo='BJ')
    2 Address(addressId=3, userId=5, addressInfo='BJ')
    2 Address(addressId=4, userId=6, addressInfo='GZ')
    2 Address(addressId=5, userId=7, addressInfo='WU')
    2 Address(addressId=6, userId=8, addressInfo='WU')
    2 Address(addressId=7, userId=15, addressInfo='HLD')
    2 Address(addressId=8, userId=15, addressInfo='XJ')
    2 Address(addressId=9, userId=15, addressInfo='DL')

    2017-06-04 22:03:18.821 INFO 1700 — [Thread-2] o.s.w.c.GenericWebApplicationContext : Closing org.springframework.web.context.support.GenericWebApplicationContext@005f0241: startup date [Sun Jun 04 22:03:13 CST 2017]; root of context hierarchy
    2017-06-04 22:03:18.826 INFO 1700 — [Thread-2] j.local.ContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
    2017-06-04 22:03:18.832 INFO 1700 — [Thread-2] com.alibaba.druid.pool.DruidDataSource : [dataSource-1] closed

Process finished with exit code 0
```

大家通过控制台的输出可以看到，其实hibernate在查询数据库时，会自动生成所需要查询的SQL语句。

JPA分页实现

JPA使用Page实现分页：

不知道大家之前实现分页是如何实现的，一般的思路有两种，一个是数据库控制，也就是通过SQL语句，像Mysql的limit啊，SQLserver的Top啊（oracle的分页太复杂不举例了……），还有就是通过代码实现，一般我平时的话会使用PageHelper来实现。当然在SpringBoot中PageHelper也是支持的。

SpringBoot开发详解（六）-- 异常统一管理
以及AOP的使用

阅读量: 9583

SpringBoot开发详解（十一）--Redis在SpringBoot中的整合

因燃量：8411

最新评论

SpringBoot开发详解 (十一...)

xia_zai_pin_dao_：按照博主的思路写下来后，程序运行是正确的，但是在redis的客户端却是乱码，而且看不到值，但是用代...

SpringBoot开发详解 (六) ...

qq_35981283: 很好的文章

SpringBoot开发详解（一）...

qq_31211685: 很详细，赞一个

SpringBoot开发详解 (六) ...

weixin_42096420: 楼主写的很好，感谢，接下来的我都要看看，太棒了

SpringBoot开发详解（五）...

sss1342746626: 感谢楼主。。终于发现一个接收
数据好用的方法了。。。赞



| 联系我们

请扫描二维码联系客服



 webmaster@csdn.net

 400-660-0108

关于 招聘 广告服务 百度

```

6
7 <dependency>
8   <groupId>com.github.pagehelper</groupId>
9   <artifactId>pagehelper-spring-boot-starter</artifactId>
10  <version>1.1.0</version>
11 </dependency>

```

不过其实SpringBoot已经有了一个比较强大的分页插件，就是Page，那让我们看看Page在JPA中是如何使用的。

首先，我们在继承JpaRepository写一个findAllAddress方法，传入参数为Pageable，而返回不再是List，而是Page。

```

1 Page<Address> findAllAddress(Pageable pageable);

```

这里Pageable是指需要传入的分页信息，而Page则是分页后的数据内容。我们继续使用测试用例来测试我们的代码。

```

1 @RunWith(SpringJUnit4ClassRunner.class)
2 @SpringBootTest(classes = Round1Application.class)
3 public class AddressRepositoryTest {
4
5     @Autowired
6     private AddressRepository addressRepository;
7
8     @Test
9     public void findAllAddress() throws Exception {
10
11         int page = 0, size = 10;
12         Sort sort = new Sort(Sort.Direction.DESC, "addressId");
13         Pageable pageable = new PageRequest(page, size, sort);
14         Page<Address> a = findAllAddress(pageable);
15         System.out.println(a.getTotalPages());
16         System.out.println(a.getTotalElements());
17         List<Address> aa = a.getContent();
18         for (Address u : aa) {
19             System.out.println(u.toString());
20         }
21     }
22 }
23 }

```

这里我们使用Sort来定义根据那一个字段进行排序，起始页一定要注意是从0开始的，而不是从1开始，我们规定每页包含10条内容，为了方便测试，我们在数据库中再加入一些内容：

对象 tp_address @blog (zzp)			
	address_id	add_info	user_id
1	SH	3	
2	XA	4	
3	BJ	5	
4	GZ	6	
5	WX	7	
6	NC	14	
7	HLJ	15	
8	XJ	15	
9	DL	15	
10	SH	3	
11	XA	4	
12	BJ	5	
13	GZ	6	
14	WX	7	
15	NC	14	
16	HLJ	15	
17	XJ	15	
18	DL	15	

http://blog.csdn.net/qq_31001665

好了，我们开始测试我们的代码，发现竟然报错了！！



```
java.lang.IllegalStateException: Failed to load ApplicationContext
    at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadContext(DefaultCacheAwareContextLoaderDelegate.java:124)
    at org.springframework.test.context.support.DefaultTestContext.getApplicationContext(DefaultTestContext.java:83)
    at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.setUpTestContextIfNecessary(DependencyInjectionTestExecutionListener.java:109)
    at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.prepareTestInstance(DependencyInjectionTestExecutionListener.java:71)
    at org.springframework.test.context.junit.SpringContextManager.createTestContext(SpringContextManager.java:230)
    at org.springframework.test.context.junit.SpringJUnit4ClassRunner.createTest(SpringJUnit4ClassRunner.java:220)
    at org.springframework.test.context.junit.SpringJUnit4ClassRunner.(SpringJUnit4ClassRunner.java:182) +4 internal calls
    at org.springframework.test.context.junit.SpringJUnit4ClassRunner.methodBlocker(SpringJUnit4ClassRunner.java:171)
    at org.springframework.test.context.junit.SpringJUnit4ClassRunner.runChild(SpringJUnit4ClassRunner.java:241) +5 internal calls
    at org.springframework.test.context.junit.SpringJUnit4ClassRunner.run(SpringJUnit4ClassRunner.java:231)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
    at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:70)
    at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:284) +5 internal calls
    at org.junit.runners.ParentRunner.access$000(ParentRunner.java:55)
    at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:264) +5 internal calls
    at org.junit.runners.ParentRunner.run(ParentRunner.java:364)
    at org.junit.runner.JUnitCore.run(JUnitCore.java:137)
    at org.junit.runner.JUnitCore.(JUnitCore.java:111)
    at org.junit.runner.JUnitCore.(JUnitCore.java:107) +5 internal calls
Caused by: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'documentationPluginsBootstrapper' defined in class path resource [org/springframework/doc/manuals/documentation/manuals/bootstrapper/DocumentationPluginsBootstrapper.class]: Unsatisfied dependency expressed through constructor parameter 0: nested exception is org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'requestMappingHandlerMapping' defined in URL [jar:file:/Users/zhu Zhengping/.m2/repository/org/springframework/spring-web/2.5.6/spring-web-2.5.6.jar!/org/springframework/web/plugins/mvc/requestMappingHandlerProvider.class]: Unsatisfied dependency expressed through constructor parameter 0: nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'requestMappingHandlerMapping' defined in class path resource [org/springframework/web/bootstrap/config/WebMvcAutoConfiguration$EnableWebMvcConfiguration.class]: Bean instantiation via factory method
```

我们写错什么了呢？看似我们什么也没写错，其实，使用Page来分页在JPA中是有严格的方法名匹配的，如果像这里我们想获得所有数据信息，就必须使用**findAll()** 来作为方法名。我们把之前写的**findAll()** 给注释了，并且将**findAllAddress**方法改为**findAll**后再进行测试，法相已经可以通过了，并且分页也为我们按照10条一页进行了划分。

```
Run AddressRepositoryTest.xml...  
AddressRepositoryTest (com.zp.sins)  
 findAll  
    key /xml/updateUser/{id}.con.zp.controller.XMLController.updateUser,DefaultGenericTypeNamingStrategy  
    2017-06-04 23:05:10.556 INFO 3495 — [main] s.w.s.c.AnnotationResource@grouping : Group for method updateUser was xml-controller  
    2017-06-04 23:05:10.556 INFO 3495 — [main] s.w.s.c.AnnotationResource@grouping : Group for method updateUser was xml-controller  
    updateUsingPUT_1  
    2017-06-04 23:05:10.568 INFO 3495 — [main] com.zp.dao.jpa.AddressRepositoryTest : Started AddressRepositoryTest in 6.097 seconds  
    Hibernate: select address0_.address_id as address_1_0_, address0_.add_info as add_info0_ , address0_.user_id as user_id3_0_ from tp_address address0_ where address0_.order_by_address0_.address_id desc limit ?  
    Hibernate: select count(address0_.address_id) as col_0_0_ from tp_address address0_  
    18  
    Address(addressId=18, userId=15, addressInfo='DL')  
    Address(addressId=17, userId=15, addressInfo='XJ')  
    Address(addressId=16, userId=14, addressInfo='LJ')  
    Address(addressId=15, userId=14, addressInfo='NC')  
    Address(addressId=14, userId=7, addressInfo='WX')  
    Address(addressId=13, userId=6, addressInfo='GZ')  
    Address(addressId=12, userId=6, addressInfo='LJ')  
    Address(addressId=11, userId=4, addressInfo='XA')  
    Address(addressId=10, userId=3, addressInfo='SH')  
    Address(addressId=9, userId=3, addressInfo='DL')  
    2017-06-04 23:05:10.691 INFO 3495 — [Thread-2] o.s.w.c.GenericWebApplicationContext : Closing org.springframework.web.context.support.GenericWebApplicationContext@600921fe[ startup date [Sun Jun 04 23:05:10 CST 2017], root of context hierarchy  
    2017-06-04 23:05:10.699 INFO 3495 — [Thread-2] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'  
    2017-06-04 23:05:10.784 INFO 3495 — [Thread-2] com.alibaba.druid.pool.DruidDataSource : [dataSource-3] closed  
  
Process finished with exit code 0
```

以上就是使用JPA的分页进行数据分页，其实不得不说，JPA其实是一个比较大的坑，如果不是熟练使用的话，很容易犯错，而且各种错都比较难以排查，比如这种命名规范的错误。所以大家如果使用分页的话，还是暂时先继续PageHelper吧，希望Spring之后能将Page封装的更为完美和易用。

以上所有的代码我已经上传到GitHub

round1-springboot

如果心急的小伙伴也可以去clone我已经完成的项目，这个项目中把一些常用功能都写了，并且都写注释啦！！！也希望大家能帮我点个赞哦～～～

MySpringBoot

版权声明：本文为博主原创文章，未经博主允许不得转载。https://blog.csdn.net/qg_31001665/article/details/72862460

文章标签： [数据库](#) [ipa](#) [spring-boot](#) [分页](#)

个人分类: [spring boot](#)



那都不是事！学会AI这些技能，让你叱咤人工智能领域！

时代趋势，顺流而上！掌握人工智能，避免就业危机！成为稀缺人才我们有方法...

12522

[查看更多>>](#)

想对作者说点什么？[我来说一句](#)

spring boot 第二课-完成单表操作及分页查询

第一部分 基本注释释意 1.前面做了一个入门例子，有了一个轻松的体验，今天来做一个模拟前后端交互的数据请求与响应的例子 如果你之前用过springmvc的注解就很好理解了。假如没有用过就需要

SpringBoot JPA实现增删改查、分页、排序、事务操作等功能

今天给大家介绍一下SpringBoot中JPA的一些常用操作，例如：增删改查、分页、排序、事务操作等功能。下面先来介绍一下JPA中一些常用的查询操作：//And --- 等价于SQL中的...

linzhiqiang0316 2016-09-23 16:38:50 阅读数：41634

聚类分析学习笔记(一) - CSDN博客

聚类分析的作用是根据在数据中发现的描述对象及其关系...今天看了看JSP的页面分页技术,顺便学习了一下,这次...过去用EntityBean,在EJB3后不推荐使用,而是用JPA...

2018-3-3

undefined

JPA中的Page与Pageable

Page是Spring Data提供的一个接口，该接口表示一部分数据的集合以及其相关的下一部分数据、数据总数等相关信息，通过该接口，我们可以得到数据的总体信息（数据总数、总页数...）以及当前数据的信息...

u011781521 2017-07-06 14:08:19 阅读数：5945

SpringBoot系列(6)--SpringBoot-JPA

JPA 应该都熟悉了，我就不多说了什么是JPA了。目前JPA主要实现由hibernate和openJPA等。 Spring Data JPA 是Spring Data 的一个子项目，它通过提供基于JP...

tony308001970 2017-07-12 00:17:53 阅读数：15305

springboot学习笔记5(JPA 实现分页、排序、返回map集合)

JPA (Java Persistence API) 中文就是java持久层api，是java EE 的一个规范。springboot 简化配置，对很多常用开源框架做了很好的集成，几乎无需配置即可直接使...

trntaken 2017-09-06 20:07:32 阅读数：4803

SpringBoot操作MongoDB实现分页查询

前几天写了一篇关于介绍SpringBoot的简单使用。以及使用SpringBoot JPA做了一次数据库的一个CURD (地址:<http://blog.csdn.net/canot/article/d...>)

canot 2016-06-03 00:34:25 阅读数：8693

Spring Data JPA: 分页和排序

转载自：Ricky之前我们学习了如何使用Jpa访问关系型数据库。通过Jpa大大简化了我们对数据库的开发工作。但是，之前的例子中我们只提到了最简单的CRUD（增删改查）操作。实际上，Spring D...

u011848397 2016-08-08 14:39:58 阅读数：25498

jpa 里用的分页实现类org.springframework.data.domain.PageImpl<T>

/* * Copyright 2008-2013 the original author or authors. ** Licensed under the Apache License, V...

u014236541 2016-03-25 19:12:38 阅读数：8997

用Jpa查询的数据，不能用PageHelper进行分页

PageHelper.startPage(requestVO.getPageNum(), requestVO.getPageSize()); 1》 List allCaseTypes =caset...

qq_32662595 2017-10-30 10:10:05 阅读数：452

解决Spring Spring Data JPA 错误： Page 1 of 1 containing UNKNOWN instances

解决Spring Spring Data JPA 错误： Page 1 of 1 containing UNKNOWN instances SpringBoot 整合 Spring-Data-JPA...

universsky2015 2017-09-13 14:36:31 阅读数：1513

Spring data jpa详解

1.什么是spring data jpa
spring data jpa是spring 基于ORM框架、jpa规范的基础上封装的一套jpa应用框架，可使开发者用极简的代码即可实现对数据的访问和操作...

yd201430320529 2017-07-12 23:30:58 阅读数: 2483

spring data jpa page interface

原文

u013275741 2014-09-23 17:15:24 阅读数: 1085

SpringBoot学习-支持PageHelp插件使用

分页是日常需求中常用的功能，今天来看看SpringBoot中支持的PageHelp的配置实现。在先前的百度中，我找了一种自定义BatisConfig类中配置分页插件的实现方法，研究了一段时间，在...

ZNG_XP 2017-09-28 11:06:03 阅读数: 577

有道智云 精准的文字识别OCR服务接口

为您的应用内置整图OCR文字检测、定位、和识别服务。点击这里 免费体验



springboot 分页(page)

第一步:在Repository继承JpaSpecificationExecutor接口,PagingAndSortingRepository接口 例如: 第二步:实现service分页业务@Ove...

cutterwolf 2017-09-05 11:23:50 阅读数: 1203

Spring Boot+Mybatis+Pagehelper分页

Spring Boot 集成MyBatis和Pagehelper分页插件 mybatis-spring-boot-starter依赖树如下： pom配置 project xmlns="http...

xiaolyuh123 2017-06-20 19:00:12 阅读数: 27123

SpringBoot学习 - (七) SpringBoot分页插件PageHelper

访问数据库采用mybatis框架1.添加pom文件依赖 org.springframework.boot spring-boot-starter-web ...

qq_28988969 2017-09-25 09:50:52 阅读数: 9642

spring-boot 集成mybatis的分页插件PageHelper和Generator (番外)

分页插件PageHelper 参看了pagehelper-spring-boot，使用起来非常方便。添加maven依赖 根据上面地址中的说明，只需要添加如下依赖，并且可以不再mybatis的依赖 ...

zl18310999566 2017-01-07 10:28:31 阅读数: 27432

Spring Boot整合PageHelper

网上有很多其他框架整合PageHelper的。但是spring boot的很少。官网也没有样例。所以，自己做，也顺便写博客记录一下，给大家参考。1. pagehelper版本: 5.1.2.2....

qq_33934809 2018-01-08 13:11:23 阅读数: 792

呼叫中心系统

呼叫中心系统基础知识

百度广告



Spring boot使用pagehelper 报空指针异常

项目中使用Mybatis+pagehelper正常运行没有问题，应需求要求需要配置多数据源，可是当我配置完后，新的数据源正常运行，原来查询的都出空指针异常了、错误如下：2017-09-20 19:...

yy756127197 2017-09-20 20:22:03 阅读数: 1258

Spring Boot自定义错误页面，Whitelabel Error Page处理方式

我已经是Spring Framework框架的忠实粉丝。对于企业软件开发者来说它提供了对常见问题的通用解决方案，包括那些你在未来开发中没有意识到的问题。但是，它构建的J2EE项目变得比较臃肿，需要被一...

github_32521685 2015-12-06 21:22:17 阅读数: 60769

spring boot yml整合pagehelper的几种方式和问题