

☺ 纯洁的微笑 (/)

Home (/) Spring-Boot (/spring-boot.html)

spring data jpa介绍

Spring-Cloud (/spring-cloud.html) Archives (/archives.html)

首先了解什么是...

- spring data jpa

基本查询 (/link.html) About (/about.html)

- 预先生成方法

- 自定义简单查询

复杂查询

springboot(五): spring data jpa的使用

📅 2016/08/20

- 自定义SQL查询

- 多表查询

多数据源的支持

- 同源数据库的多源支持

- 异构数据库多源支持

其它

在上篇文章springboot(二): web综合开发

参考
(<http://www.ityouknow.com/springboot/2016/02/03/spring-boot-web.html>)中简单介绍了一下spring data jpa的基础性使用, 这篇文章将更加全面的介绍spring data jpa 常见用法以及注意事项

使用spring data jpa 开发时, 发现国内对spring boot jpa全面介绍的文章比较少案例也比较零碎, 因此写文章总结一下。本人也正在翻译Spring Data JPA 参考指南

(<https://www.gitbook.com/book/ityouknow/spring-data-jpa-reference-documentation/details>),有兴趣的同学欢迎联系我, 一起加入翻译中!

spring data jpa介绍

首先了解JPA是什么?

JPA(Java Persistence API)是Sun官方提出的Java持久化规范。它为Java开发人员提供了一种对象/关联映射工具来管理Java应用中的关系数据。他的出现主要是为了简化现有的持久化开发工作和整合ORM技术,结束现在Hibernate, TopLink, JDO等ORM框架各自为营的局面。值得注意的是, JPA是在充分吸收了现有Hibernate, TopLink, JDO等ORM框架的基础上发展而来的, 具有易于使用, 伸缩性强等优点。从目前的开发社区的反应上看, JPA受到了极大的支持和赞扬, 其中就包括了Spring与JB3.0的开发团队。

spring data jpa介绍

- 首先了解JPA是什么?

- spring data jpa

- 基本查询

- 预先生成方法

- 自定义简单查询

- 复杂查询

- 分页查询

- 自定义SQL查询

- 多表查询

- 数据库代理

- 数据库支持

- 异构数据库多源支持

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

- 参考

- 其它

基本查询

基本查询也分为两种, 一种是spring data默认已经实现, 一种是根据查询的方法来自动解析成SQL。

预先生成方法

spring data jpa 默认预先生成了一些基本的CURD的方法, 例如: 增、删、改等等

1 继承JpaRepository

```
public interface UserRepository extends JpaRepository<User, Long> {
}
```

2 使用默认方法

spring data jpa介绍

- 首先了解JPA是什么?

@Test

spring data jpa

```
public void testBaseQuery() throws Exception {
```

基本查询

```
User user=new User();
```

```
userRepository.findAll();
```

```
userRepository.findOne(1l);
```

复杂查询

```
userRepository.save(user);
```

```
userRepository.delete(user);
```

```
userRepository.count();
```

```
userRepository.exists(1l);
```

```
userRepository..
```

多数据源的支持

- 同源数据库的多源支持

就不解释了根据方法名就看出意思来

其它

自定义简单查询

自定义的简单查询就是根据方法名来自动生成SQL，主要的语法

是 findXXBy , readAXXBy , queryXXBy , countXXBy , getXXBy 后面跟属性名称:

```
User findByUserName(String userName);
```

也使用一些加一些关键字 And 、 Or

```
User findByUserNameOrEmail(String username, String email);
```

修改、删除、统计也是类似语法

```
Long deleteById(Long id);

Long countByUsername(String userName);
```

spring data jpa介绍

- 首先了解JPA是什么?

基本上SQL体系中的关键词都可以使用, 例如: LIKE、 IgnoreCase、 OrderBy。

基本查询

预先生成方法

```
List<User> findByEmailLike(String email);
```

- 自定义简单查询

复杂查询

```
User findByIdByUsernameIgnoreCase(String userName);
```

- 分页查询

```
List<User> findByUserNameOrderByEmailDesc(String email);
```

- 自定义SQL查询

- 多表查询

具体的关键字, 使用方法和生成SQL如下表所示

多数据源的支持

Keyword	Sample	JPQL snippet
其它 And 参考	findByLastnameAndFirstname	... where x.lastname = ? 1 and x.firstname = ?2
Or	findByLastnameOrFirstname	... where x.lastname = ? 1 or x.firstname = ?2
Is,Equals	findByFirstnamesIs,findByFirstnameEquals	... where x.firstname = ? 1
Between	findByStartDateBetween	... where x.startDate between ?1 and ?2
LessThan	findByAgeLessThan	... where x.age < ?1

Keyword	Sample	JPQL snippet
LessThanEqual	findByAgeLessThanEqual	... where x.age ≤ ?1
GreaterThan	findByAgeGreaterThan	... where x.age > ?1
GreaterThanEqual	findByAgeGreaterThanEqual	... where x.age ≥ ?1
After	findByStartDateAfter	... where x.startDate > ?1
Before	findByStartDateBefore	... where x.startDate < ?1
IsNull	findByAgeIsNull	... where x.age is null
IsNotNull	findByAge(Is)NotNull	... where x.age not null
Like	findByFirstnameLike	... where x.firstname like ?1
NotLike	findByFirstnameNotLike	... where x.firstname not like ?1
StartingWith	findByFirstnameStartingWith	... where x.firstname like ?1 (parameter bound with appended %)
EndingWith	findByFirstnameEndingWith	... where x.firstname like ?1 (parameter bound with prepended %)

Keyword	Sample	JPQL snippet
Containing	findByFirstnameContaining	... where x.firstname like ?1 (parameter bound wrapped in %)
spring data jpa介绍 - 首先了解JPA是什么? spring data jpa OrderBy 基本查询	findByAgeOrderByLastnameDesc	... where x.age = ?1 order by x.lastname desc
- 预先生成方法 自定义简单查询 Not 复杂查询	findByLastnameNot	... where x.lastname <> ?1
In - 分页查询 - 自定义SQL查询	findByAgeIn(Collection ages)	... where x.age in ?1
Not in - 多表查询	findByAgeNotIn(Collection age)	... where x.age not in ?1
多数据源的支持 TRUE - 同源数据库的多源支持	findByActiveTrue()	... where x.active = true
FALSE - 异构数据库多源支持	findByActiveFalse()	... where x.active = false
其它 参考 IgnoreCase	findByFirstnameIgnoreCase	... where UPPER(x.firstname) = UPPER(?1)

复杂查询

在实际的开发中我们需要用到分页、删选、连表等查询的时候就需要特殊的方法或者自定义SQL

分页查询

分页查询在实际使用中非常普遍了, spring data jpa已经帮我们实现了分页的功能, 在查询的方法中, 需要传入参数 Pageable ,当查询中有多个参数的时候 Pageable 建议做为最后一个参数传入

```
Page<User> findALL(Pageable pageable);
```

spring data jpa介绍

```
Page<User> findByUserName(String userName,Pageable pageable);
```

- 首先了解JPA是什么?

- spring data jpa

Pageable是spring封装的分页实现类, 使用的时候需要传入页数、每页条数和排序规则

- 预先生成方法

@Test

```
public void testPageQuery() throws Exception {
```

- 分页查询
int page=1,size=10;

- 自定义SQL查询
Sort sort = new Sort(Direction.DISC, "id");

- 多表查询
Pageable pageable = new PageRequest(page, size, sort);

- 多数据源的支持
userRepository.findAll(pageable);

- 同源数据库的多源支持
userRepository.findByUserName("testName", pageable);

- 异构数据库多源支持

限制查询

参考

有时候我们只需要查询前N个元素, 或者支取前一个实体。

```
userRepository.findFirstOrderByLastnameAsc();
```

```
User findTopOrderByAgeDesc();
```

```
Page<User> queryFirst10ByLastname(String lastname, Pageable pageable);
```

```
List<User> findFirst10ByLastname(String lastname, Sort sort);
```

```
List<User> findTop10ByLastname(String lastname, Pageable pageable);
```

自定义SQL查询

其实Spring data 觉大部分的SQL都可以根据方法名定义的方式来实现，但是由于某些原因我们想使用自定义的SQL来查询，spring data也是完美支持的；在SQL的查询方法上面使用 @Query 注解，如涉及到删除和修改在需要加上 @Modifying .也可以根据需要添加 @Transactional 对事物的支持，查询超时的设置等

spring data jpa介绍

的设置等

- 首先了解JPA是什么？

- spring data jpa

@Modifying

基本查询

@Query("update User u set u.userName = ?1 where u.id = ?2")

- 预先生成方法

int modifyByIdAndUserId(String userName, Long id);

- 自定义简单查询

@Transactional

@Modifying

分页查询

@Query("delete from User where id = ?1")

- 自定义SQL查询

void deleteByUserId(Long id);

- 多表查询

@Query(timeout = 10)

@Query("select u from User u where u.emailAddress = ?1")

- 同源数据库的多源支持

User findByEmailAddress(String emailAddress);

- 异构数据库多源支持

其它

多表查询

多表查询在spring data jpa中有两种实现方式，第一种是利用hibernate的级联查询来实现，第二种是创建一个结果集的接口来接收连表查询后的结果，这里主要第二种方式。

首先需要定义一个结果集的接口类。


```

public interface HotelSummary {

    City getCity();

    String getName();

    Double getAverageRating();

    default Integer getAverageRatingRounded() {
        return getAverageRating() == null ? null : (int) Math.round(getAverageRating());
    }

    // 复杂查询
    // - 分页查询
    // - 自定义SQL查询

```

查询的方法返回类型设置为新创建的接口

多数据源的支持

@Query("select h.city as city, h.name as name, avg(r.rating) as averageRating " "from Hotel h left outer join h.reviews r where h.city = ?1 group by h")

Page<HotelSummary> **findByCity**(City city, Pageable pageable);

其它

参考

@Query("select h.name as name, avg(r.rating) as averageRating " "from Hotel h left outer join h.reviews r group by h")

Page<HotelSummary> **findByCity**(Pageable pageable);

使用

```

Page<HotelSummary> hotels = this.hotelRepository.findByCity(new PageRequest(0, 10, Direction.ASC,
for(HotelSummary summar:hotels){
    System.out.println("Name" +summar.getName());
}

```

在运行中Spring会给接口（HotelSummary）自动生产一个代理类来接收返回的结果，代码汇总使用 getXX 的形式来获取

spring data jpa介绍 多数据源的支持

- spring data jpa

同数据库的多源支持

- 预先生成方法

日常项目中因为使用的分布式开发模式，不同的服务有不同的数据源，常常需要在一个项目中使用多个数据源，因此需要配置spring data jpa对多数据源的使用，一般分一下为三步：

- 分页查询
- 1 配置多数据源
 - 自定义SQL查询
- 2 不同源的实体类放入不同包路径
 - 多表查询
- 3 声明不同的包路径下使用不同的数据源、事务支持

- 同源数据库的多源支持

异构数据库多源支持

其它
参考
比如我们的项目中，即需要对mysql的支持，也需要对mongodb的查询等。

实体类声明 @Entity 关系型数据库支持类型、声明 @Document 为mongodb支持类型，不同的数据源使用不同的实体就可以了

```
interface PersonRepository extends Repository<Person, Long> {
    ...
}
```

@Entity
spring data jpa介绍
public class Person {
 - 首先了解JPA是什么?
 } - spring data jpa

基本查询
interface UserRepository extends Repository<User, Long> {
 - 预先生成方法
 ...
 } - 自定义简单查询

复杂查询
@Document
public class User {
 - 自定义SQL查询
 ...
 } - 多表查询

多数据源的支持

- 同源数据库的多源支持
 但是, 如果User用户既使用mysql也使用mongodb呢, 也可以做混合使用
 - 异构数据库多源支持

其它
interface JpaPersonRepository extends Repository<Person, Long> {
 ...
 }

```
interface MongoDBPersonRepository extends Repository<Person, Long> {
    ...
}
```

@Entity
@Document
public class Person {
 ...
 }

也可以通过对不同的包路径进行声明, 比如A包路径下使用mysql,B包路径下使用mongoDB

```
@EnableJpaRepositories(basePackages = "com.neo.repositories.jpa")
@EnableMongoRepositories(basePackages = "com.neo.repositories.mongo")
```

```
interface Configuration { }
spring data jpa介绍
```

- 首先了解JPA是什么?
- spring data jpa

基本查询

其它

- 预先生成方法
- 自定义简单查询

使用枚举

- 分页查询

使用枚举的时候, 我们希望数据库中存储的是枚举对应的String类型, 而不是枚举的索引值, 需要在属

性上面添加@Enumerated(EnumType.STRING) 注解

多数据源的支持

```
@Enumerated(EnumType.STRING)
```

```
@Column(name = "type", nullable = true)
```

```
private UserType type;
```

其它

参考

不需要和数据库映射的属性

正常情况下我们在实体类上加入注解 @Entity , 就会让实体类和表相关连如果其中某个属性我们不需要和数据库来关联只是在展示的时候做计算, 只需要加上 @Transient 属性既可。

```
@Transient
private String userName;
```

源码案例

这里有一个开源项目几乎使用了这里介绍的所有标签和布局，大家可以参考：

示例代码-github (<https://github.com/cloudfavorites/favorites-web>)

示例代码-码云 (<https://gitee.com/ityouknow/favorites-web>)

spring data jpa介绍

参考 首先了解JPA是什么？

- spring data jpa

基本查询

Spring Data JPA - Reference Documentation (<http://docs.spring.io/spring-data/jpa/docs/current/reference/html/>)

- 预先生成方法

- 自定义简单查询

Spring Data JPA——参考文档 中文版 (<https://www.gitbook.com/book/ityouknow/spring-data-jpa-reference-documentation/details>)

- 分页查询

- 自定义SQL查询

- 多表查询

多数据源的支持

作者：纯洁的微笑

同源数据库的多源支持

异构数据库多源支持

出处：www.ityouknow.com (<http://www.ityouknow.com>)

版权所有，欢迎保留原文链接进行转载：)

其它

参考



扫码关注有惊喜

(转载本站文章请注明作者和出处 纯洁的微笑-ityouknow (<http://www.ityouknow.com>))

Showing 15 Comments

- 首先了解JPA是什么?

- spring data jpa

2 (<https://github.com/ityouknow/blog-comments/issues/155>) 条评论

未登录用户 ∨

基本查询

- 预先生成方法

自定义简单查询
说点什么

复杂查询

- 分页查询

- 自定义SQL查询

- 支持 Markdown 语法 (<https://guides.github.com/features/mastering-markdown/>)

- 多表查询

使用 Github 登录

多数据源的支持

- 同源数据库的多源支持

- 异构数据库多源支持



其它

qiwiwi (<https://github.com/qiwiwi>) 发表于 3 个月前

参考

请问注解写在那里以后service实现还要写@Transactional
(<https://github.com/transactional>)注解吗liuminda (<https://github.com/liuminda>) 发表于 大约 1 个月前

阅读了, 谢谢大神的分享, 辛苦了。

Post Directory

知乎 (<https://www.zhihu.com/people/ityouknow>) 微博 (<http://weibo.com/ityouknow>)

Github (<https://github.com/ityouknow>)

(/feed.xml) Power by Yummy Jekyll (<https://github.com/DONGChuan/Yummy-Jekyll>) 京ICP备15067287号-3 TOP

资源 (/share/2017/10/01/resource-sharing.html) 故事 (/life.html) 架构 (/arch.html) Jvm (/jvm.html) FastDFS (/fastdfs.html)

spring data jpa介绍

- 首先了解JPA是什么?
- MongoDB (/mongodb.html) Docker (/docker.html) Code (/open-source.html)
- spring data jpa

基本查询

- 预先生成方法
- 自定义简单查询

复杂查询

- 分页查询
- 自定义SQL查询
- 多表查询

多数据源的支持

- 同源数据库的多源支持
- 异构数据库多源支持

其它

参考