Spring Data Jpa的使用

# 关于Spring Data Jpa

在实际的开发工作中，开发人员基本面临着大量的增删改查操作，Spring Data Jpa是Spring Data的一个子项目，它通过提供基于JPA的Repository极大地减少了JPA作为数据访问方案的代码量。

# 使用实例

下面是一个在spring-boot中的如何配置以及使用spring data jpa的实例

工程配置

首先在pom.xml中添加以来，加入以下内容

<dependencies>

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-jpa</artifactId>

<version>1.11.3.RELEASE</version>

</dependency>

</dependencies>

在application.xml中配置：数据库连接信息（如使用嵌入式数据库则不需要）、自动创建表结构的设置，例如使用mysql的情况如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | spring.datasource.url=jdbc:mysql://localhost:3306/test  spring.datasource.username=root  spring.datasource.password=root  spring.datasource.driver-class-name=com.mysql.jdbc.Driver  spring.jpa.properties.hibernate.hbm2ddl.auto=create-drop |

到这里已经完成了数据的基本配置，接下来我们创建一个实体类来使用它。

## 创建实体

创建一个User实体，包含id（主键）、name（姓名）、age（年龄）属性，通过ORM框架其会被映射到数据库表中，由于配置了hibernate.hbm2ddl.auto，在应用启动的时候框架会自动去数据库中创建对应的表。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | @Entity  public class User {  @Id  @GeneratedValue  private Long id;  @Column(nullable = false)  private String name;  @Column(nullable = false)  private Integer age; |

## 定义数据访问层

使用Spring Data JPA建立数据访问层十分简单，只需定义一个继承JpaRepository的接口即可，下面针对User实体创建对应的Repository接口实现对该实体的数据访问，如下代码：

interface PersonRepository extends Repository<Person, Long> { … }

继承JpaRepository就意味着我们默认有了下面的数据访问操作方法。

@NoRepositoryBean

public interface JpaRepository<T, ID extends Serializable> extends PagingAndSortingRepository<T, ID>, QueryByExampleExecutor<T> {

List<T> findAll();

List<T> findAll(Sort var1);

List<T> findAll(Iterable<ID> var1);

<S extends T> List<S> save(Iterable<S> var1);

void flush();

<S extends T> S saveAndFlush(S var1);

void deleteInBatch(Iterable<T> var1);

void deleteAllInBatch();

T getOne(ID var1);

<S extends T> List<S> findAll(Example<S> var1);

<S extends T> List<S> findAll(Example<S> var1, Sort var2);

}

可以看到该接口本身已经实现了创建（save）、更新（save）、删除（delete）、查询（findAll、findOne）等基本操作的函数，这些操作上的函数就不用开发者在去定义。

## 定义查询方法

public interface PersonRepository extends CustomRepository<Person, Long> {

List<Person> findByName(String name);

List<Person> findByNameLike(String name);

List<Pserson> findByNameAndAddress(String name,String address);

}

在上例中，我们可以看到这个函数：findByNameAndAddress，它实现了按name和address查询User实体，这里就是Spring-data-jpa的一大特性：通过解析方法名创建查询。像Like和and这类查询的关键字，如下表中所示：

| **关键词** | **样品** | **JPQL片段** |
| --- | --- | --- |
| And | findByLastnameAndFirstname | … where x.lastname = ?1 and x.firstname = ?2 |
| Or | findByLastnameOrFirstname | … where x.lastname = ?1 or x.firstname = ?2 |
| Is,Equals | findByFirstname，findByFirstnameIs，findByFirstnameEquals | … where x.firstname = ?1 |
| Between | findByStartDateBetween | … where x.startDate between ?1 and ?2 |
| LessThan | findByAgeLessThan | … where x.age < ?1 |
| 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,NotNull | 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（附加参数绑定%） |
| EndingWith | findByFirstnameEndingWith | … where x.firstname like ?1（参数与前缀绑定%） |
| Containing | findByFirstnameContaining | … where x.firstname like ?1（绑定的参数%） |
| OrderBy | findByAgeOrderByLastnameDesc | … where x.age = ?1 order by x.lastname desc |
| Not | findByLastnameNot | … where x.lastname <> ?1 |
| In | findByAgeIn(Collection<Age> ages) | … where x.age in ?1 |
| NotIn | findByAgeNotIn(Collection<Age> age) | … where x.age not in ?1 |
| True | findByActiveTrue() | … where x.active = true |
| False | findByActiveFalse() | … where x.active = false |
| IgnoreCase | findByFirstnameIgnoreCase | … where UPPER(x.firstame) = UPPER(?1) |

关于解析方法名创建查询有篇文章写的很好这里不在赘述。<https://www.ibm.com/developerworks/cn/opensource/os-cn-spring-jpa/>

# 总结

Spring Data Jpa可以提供一种简洁的方式操作数据库，不需要开发人员写大量的增删改查代码，相对于Mybatis对于程序员开发效率有很大提升，但是缺点在于sql的自由度上有一定影响，需要根据项目情况选择技术选型。

# 参考资料

<http://www.cnblogs.com/chenpi/p/6357527.html>

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

<http://blog.didispace.com/springbootdata2/>