Spring DATA

Spring Data JPA

JPA即Java Persistence(持久化) API，一个基于O/R（Object-Relational Mapping）映射的标准规范。

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。

2.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，定义如下：

public interface PersonRepository extends JpaRepository<Person, Long> {

　　 //定义数据访问操作的方法

}

继承JpaRepository接口意味着我们默认已经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> {

　　 List<T> findAll();

List<T> findAll(Sort sort);

　　 List<T> findAll(Iterable<ID> ids);

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

　　 void flush();

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

　　 void deleteInBatch(Iterable<T> entities);

　　 void deleteAllInBatch();

　　 T getOne(ID id);

}

3.配置使用Spring Data JPA

在Spring环境中，使用Spring Data JPA可通过@EnableJpaRepositories注解来开启Spring Data JPA的支持，@EnableJpaRepositories接收的value参数用来扫描数据访问层所在包下的数据访问的接口定义。

4.定义查询方法

在讲解查询方法前，假设我们有一张数据表叫PERSON，有ID（Number）、NAME（Varchar2）、AGE（Number）、ADDRESS（Varchar2）几个字段；对应的实体类叫Person，分别有id（Long）、name（String）、age（Integer）、address（String）。下面我们就以这个简单的实体查询作为演示。

（1）根据属性名查询

Spring Data JPA支持通过定义在Repository接口中的方法名来定义查询，而方法名是根据实体类的属性名来确定的。

1）常规查询。根据属性名来定义查询方法，示例如下：

public interface PersonRepository extends JpaRepository<Person, Long> {

　　 /\*\*

　　 \*

　　 \* 通过名字相等查询，参数为name

　　 \* 相当于JPQL:select p from Person p where p.name=?1

　　 \*/

　　 List<Person> findByName(String name);

　　 /\*\*

　　 \*

　　 \* 通过名字like查询，参数为name

　　 \* 相当于JPQL：select p from Person p where p.name like ?1

　　 \*/

　　 List<Person> findByNameLike(String name);

　　 /\*\*

\*

　　 \* 通过名字和地址查询,参数为name和address

　　 \* 相当于JPQL：select p from Person p where p.name=?1 and p.address=?2

　　 \*/

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

}

从代码可以看出，这里使用了findBy、Like、And这样的关键字。其中findBy可以用find、read、readBy、query、queryBy、get、getBy来代替。

而Like和and这类查询关键字，如表8-2所示：