Spring Data 学习笔记

- Spring data : spring 的一个子项目，**用于简化数据库访问**，支持 NoSQL 和关系型数据库，主要目的是是数据库访问变得快捷

Spring Data JPA 与 MyBatis

JPA 默认使用 hibernate 作为 ORM 实现,Spring Data JPA 与 MyBatis 对比,起始也就是 hibernate 与 MyBatis 对比。

JPA 抽象了 api， 为了替代 native sql，增加了学习成本，降低了性能。复杂的查询还是只能用 native sql。

携程金融使用的是 mybatis，那就先学 mybatis 吧

Spring Data JPA 学习笔记

官方教程:
https://docs.spring.io/spring-data/jpa/docs/2.3.5.RELEASE/reference/html/#preface

https://github.com/spring-projects/spring-data-jpa/blob/master/src/main/asciidoc/jpa.adoc

Spring Data JPA 是 Spring Data 的子模块。
使用 Spring Data，使得基于“repositories”概念的 JPA 实现更简单和容易。Spring Data JPA 的目标是大大简化数据访问层代码的编码。作为使用者，我们只需要编写自己的 repository 接口，接口中包含一些个性化的查询方法，Spring Data JPA 将自动实现查询方法。
JPA 默认使用 hibernate 作为 ORM 实现，所以，一般使用 Spring Data JPA 即会使用 hibernate。

入门视频

1、Spring Data JPA是什么?

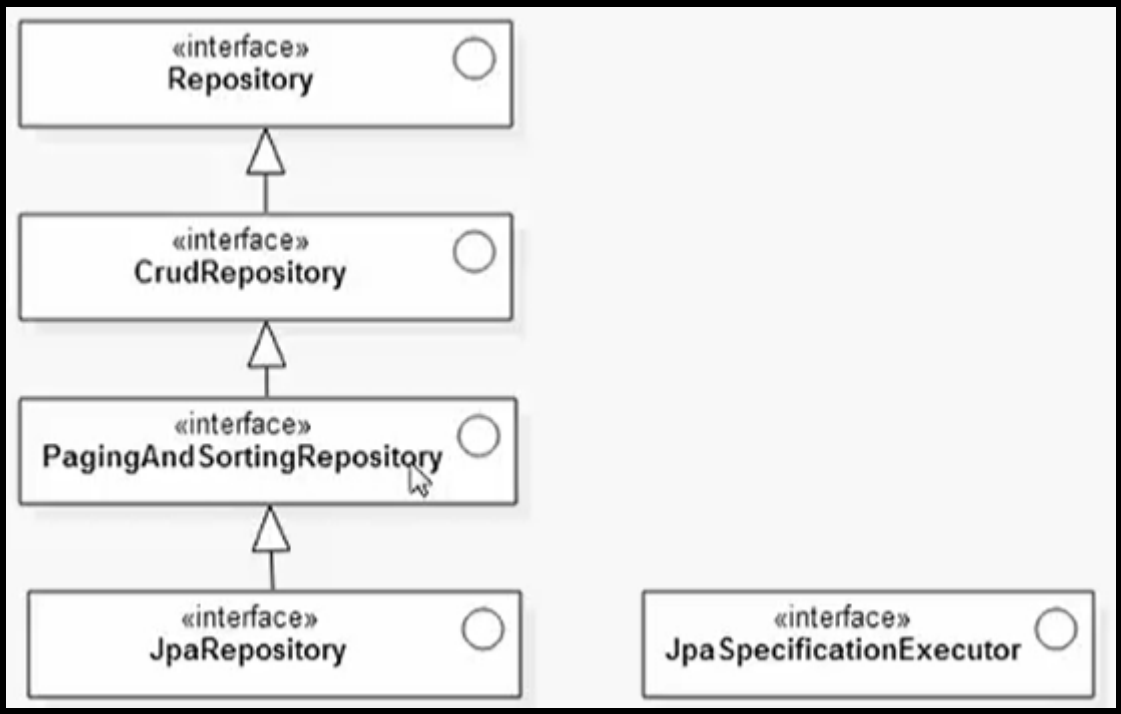
它是Spring基于**ORM框架**、**JPA规范**封装的一套**JPA应用框架**，可使开发者用**极简的代码**即可实现对数据的访问和操作。它提供了包括**增删改查**等在内的常用功能，且易于扩展！学习并使用Spring Data JPA可以**极大提高开发效率**!

JPA: java persistence api

2、Spring Data JPA 有什么?

主要看看Spring Data JPA 提供的编程接口

- **Repository:** 最顶层的接口，是一个**空接口**，目的是为了统一所有的Repository的类型，且能让组件扫描的时候自动识别。
- **CrudRepository:** Repository的子接口，提供**CRUD**的功能。
- **PagingAndSortingRepository:** CrudRepository的子接口, 添加**分页排序**的功能。
- **JpaRepository:** PagingAndSortingRepository的子接口,增加**批量操作**等功能
- **JpaSpecificationExecutor:** 用来做复杂查询的接口



JpaRepository接口方法：

- delete删除或批量删除
- findAll查找所有
- findOne查找单个
- save保存单个或批量保存
- saveAndFlush保存并刷新到数据库

查询操作的基本实现 — 基于方法名解析的概念

JpaRepository支持接口规范方法名查询。意思是如果在接口中定义的查询方法符合它的命名规则，就可以不用写实现。

例如：findByName这个方法表示从数据库中查询Name这个属性等于XXX的所有记录，类似于SQL语句：select * from xxTable where name=xxx这种形式

这段话有两个重点：

1、方法名需要在接口中设定

2、必须符合一定的命名规范

查询操作的基本实现 — 方法名构造方法

find+全局修饰+By+实体的属性名称+限定词+连接词+ ...(其它实体属性)+OrderBy+排序属性+排序方向

例如：

findDistinctByFirstNameIgnoreCaseAndLastNameOrderByAgeDesc(String firstName,String lastName){.....}

其中：Distinct是全局修饰（非必须），FirstName和LastName是实体的属性名，And是连接词，IgnoreCase是限定词，Age是排序属性，Desc是排序方向，限定词和连接词统称为“关键词”

常用词如下：

全局修饰：Distinct，Top，First

关键词：IsNull，IsNotNull，Like，NotLike，Containing，In，NotIn，IgnoreCase，Between，Equals，LessThan，GreaterThan，After，Before...

排序方向：Asc，Desc

连接词：And，Or

更多关键词请查看官方在线文档：

http://docs.spring.io/spring-data/jpa/docs/1.7.2.RELEASE/reference/html/

嵌套实体方法命名规则

构词法：主实体中子实体的名称+ _ +子实体的属性名称

例如：List<Person> findByAddress_ZipCode(ZipCode zipCode)

表示查询所有 Address（地址）的zipCode（邮编）为指定值的所有Person(人员)

```
//查询需求：从数据库中查询电话号码(phone)以指定字符串开始(例如：136)的，并且地址(address)中包含指定字符串（例如：路）的记录
//select * from user where phone like '136%' and address like '%路%'
List<User> findByPhoneStartingWithAndAdressContaining(String phone,String address);
```

```
k：从数据库中查询电话号码(phone)以指定字符串开始(例如：136)的，并且地址(address)中包含指定字符串（例如：路）的记录
ct * from user where phone like '136%' and address like '%路%' order by phone desc limit 0
ser> findTop2ByPhoneStartingWithAndAdressContainingOrderByPhoneDesc(String phone,String a
```

排序也可以不在方法名中声明 OrderBy,直接写到参数中：

```
User> findByPhoneStartingWithAndAddressContaining(String phone,String address,Sort sort);
```

```
Page<User> findByPhoneStartingWithAndAddressContaining(String phone,String address,Pa
                                (String phone,String address,Pageable pageable);
```

分页

```
//不分页带条件查询
public List<User> getUsersByConditionNoPage(String phone,String address);

//带分页条件查询(需要得到用户列表并且得到分页信息)
public Page<User> getUsersByConditionWithPage(String phone,String address,Integer page,Integer pageSize);
//带分页条件查询(得到用户列表)
//public List<User> getUsersByCondition(String phone,String address,Integer page,Integer pageSize);
```

使用 Criteria 查询

```java
//查询所有年龄大于20的实体
//JPQL: select u from User u where u.old > 20

CriteriaBuilder cb = entityManager.getCriteriaBuilder();
CriteriaQuery cq = cb.createQuery();
Root<User> root = cq.from(User.class);
cq.select(root);
Predicate pre = cb.greaterThan(root.get("").as(Integer.class), 20)
cq.where(pre);
```

https://www.jikexueyuan.com/course/1449.html