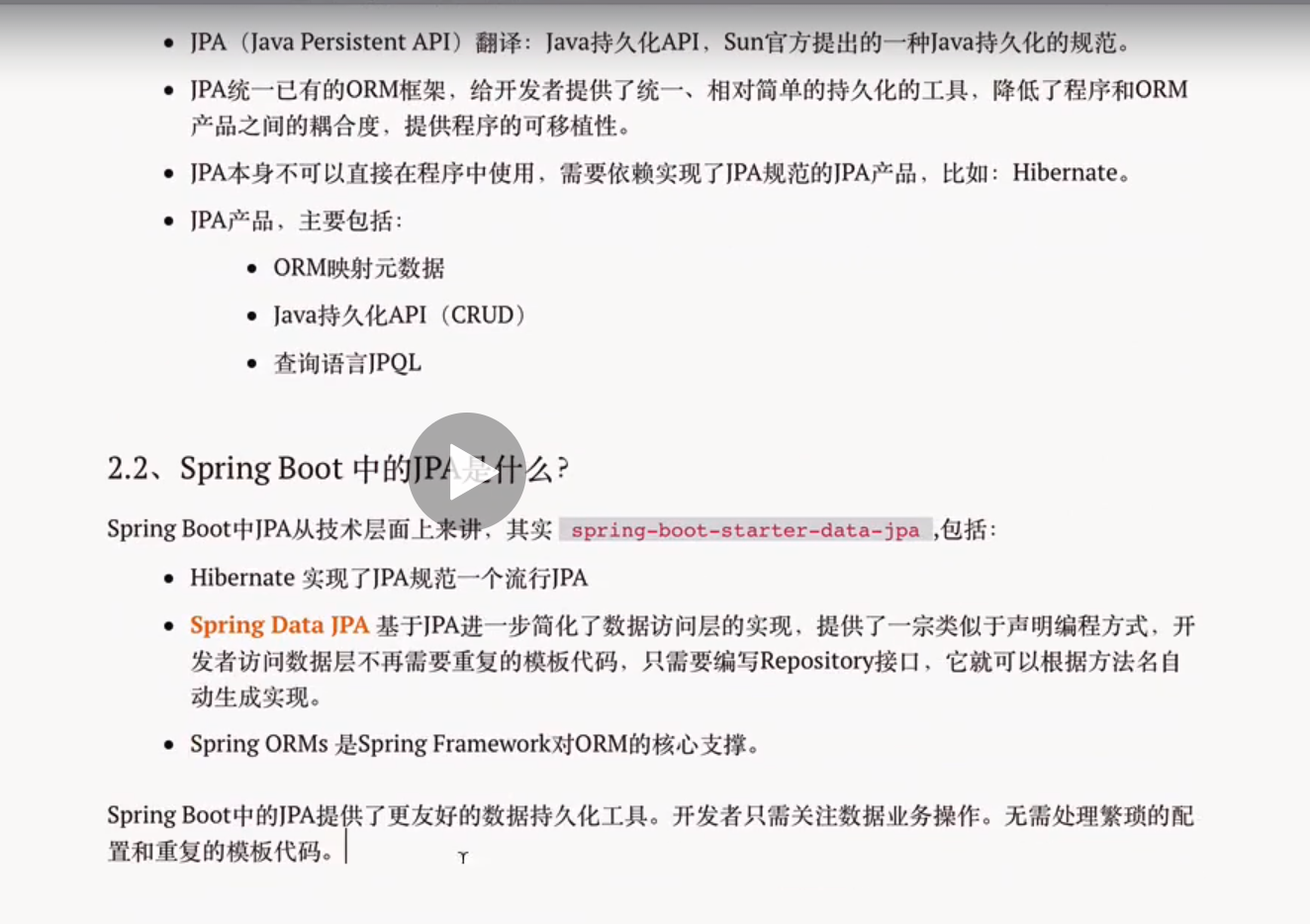
JPA： java persistence API



Spring boot JPA:只需要关注CRUD操作，无需关注具体代码实现，自动根据方法名生成实现。

JPA基本步骤：

课程实例：

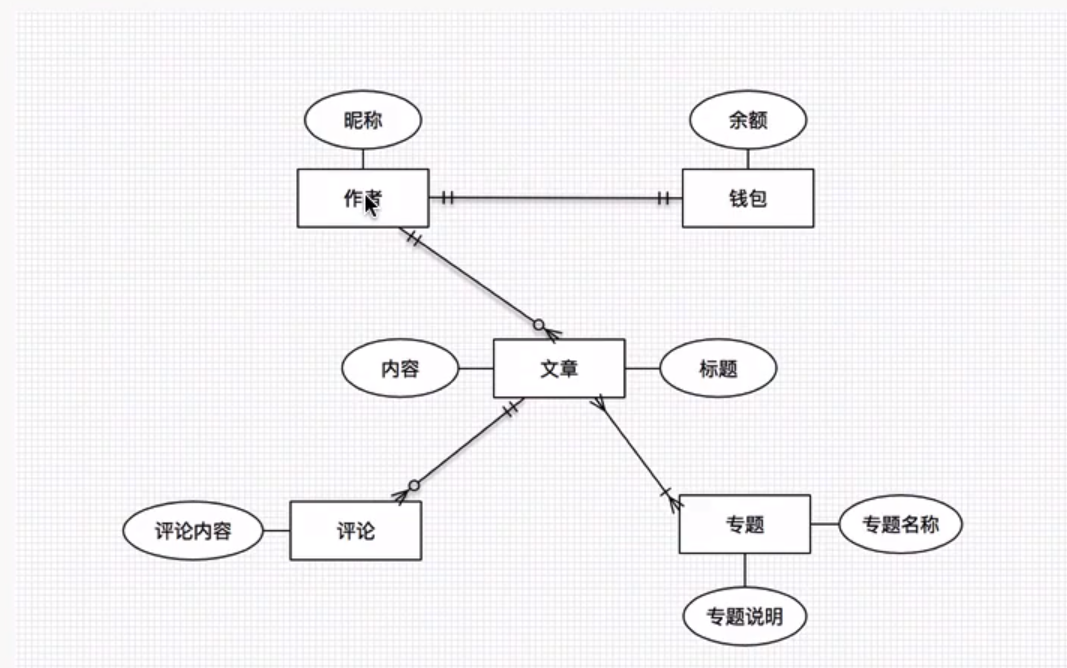
作者注册更新删除。

文章新增，更新，删除

评论文章

专题新增，更新，删除

专题收录



JPA使用步骤：

1. Spring boot框架构建
2. 引入JPA模块 （initializr引入，或者pom中加入）
3. 在properties,yml文件里配置JPA
4. 模型类？Domain包下，定义实体类
5. 实现Repository接口

Jpa.hibernate.ddl-auto参数说明？

Update:第一次加载hibernate时，会根据我们的model类在对应数据库里创建新表，以后再加载hibernate的时候，会根据model类来自动更新表结构。注意:即使实体模型发生了变化，表结构不会删除以前的列或者行，只会新增一个列。

还有create，create-drop，validate参数。

Create:第一次加载hibernate时，根据model类创建新表，每次加载hibernate都会创建新表。表数据会丢失。慎用此属性。

Create-drop:加载hibernate时，会根据model类创建新表，当sessionFactory销毁的时候，把对应的表删除掉。相当于临时表。

Validate：每次加载hibernate时候，根据model类来验证我们的表结构，不会创建新表，或者是更新表。

@Basic(fetch=LAZY)返回的数据库表对象默认不含该列属性，当调用class.getter时候，才会去数据库查找此属性。

@Transient不需要持久化，短暂使用。数据库表里不会映射此属性

@Temporal(Temporal.DATE)只存日期，不存小时 秒。。

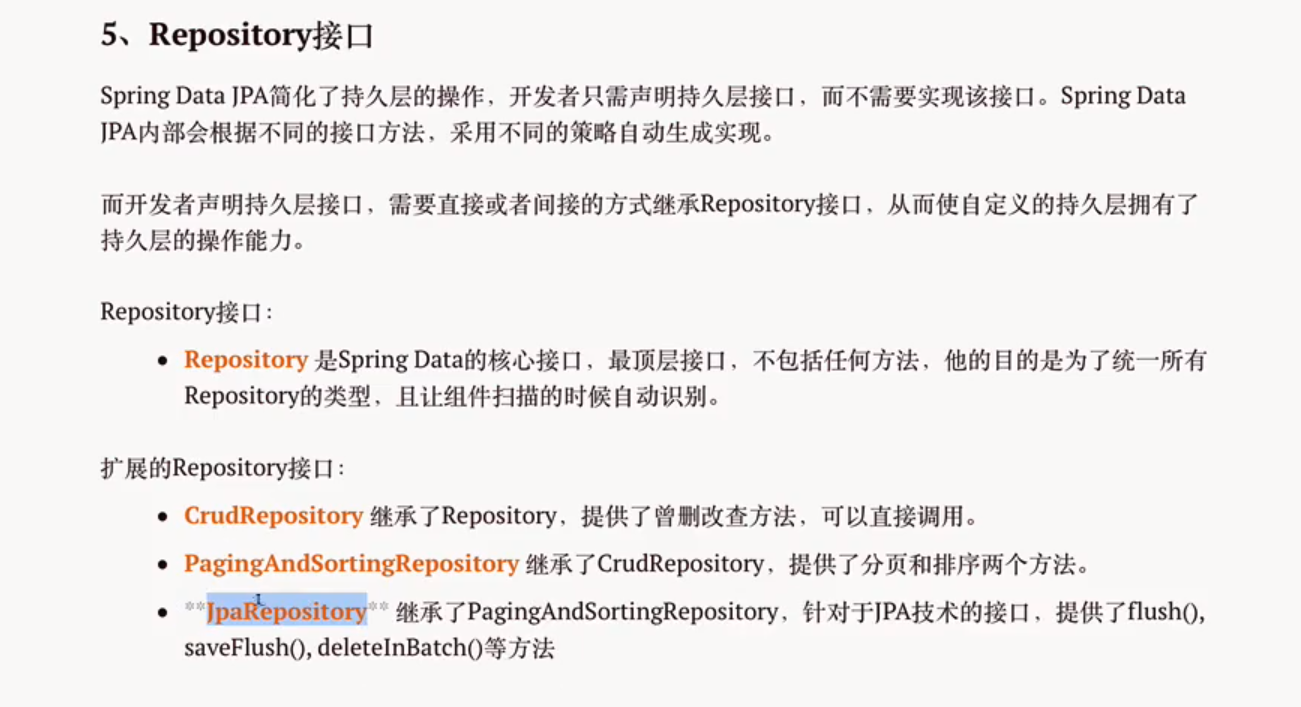
@Lob映射成BLOB或者CLOB，BLOB是图片，CLOB是文本

@Lob一般与延迟加载配合使用。

实体类内必须要加空的构造函数：

JPA对象时hibernate创建的，hibernate内部通过反射机制，反射机制需要用到无参数构造函数。

Repository接口？



方法名创建查询？

Find+全局修饰+By+实体的属性名称+限定词+连接词+。。。（其他实体属性）+OrderBy+排序属性+排序方向

Distinct是全局修饰，非必须

Nickname和phone是实体属性名

And是连接词

Ignorecase是限定词

Signdate是排序属性

Desc是排序方向

例如



JPQL语句自定义查询？？？

SQL查询，JPQL更新

使用@Query标记JPQL语句

@Query(**"select a from Author a where a.phone=?1"**)  
List<Author> findByPhone(String phone);

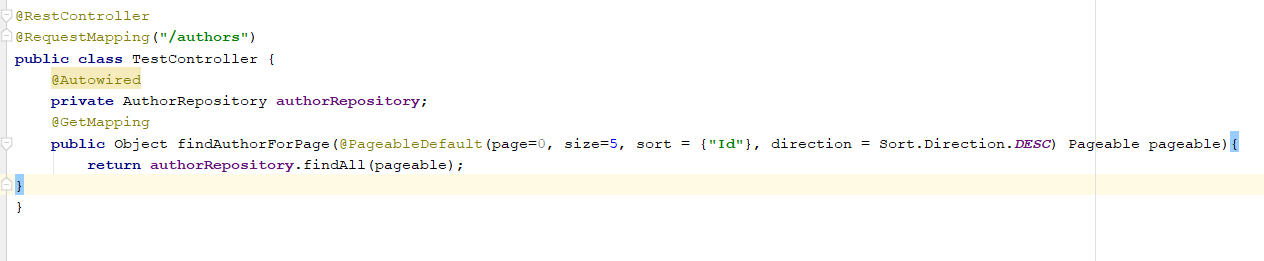
@Query(**"select a.nickName,length(a.nickName) from Author a where a.nickName like %?1%"**)  
List<Object[]> findArry(String nickName);

开启SQL语句查询？

@Query(value = **"select** *\** **from author where nickname like %?1%"**, nativeQuery = **true**)  
List<Author> findbySql(String nickName);

分页与排序和事务处理

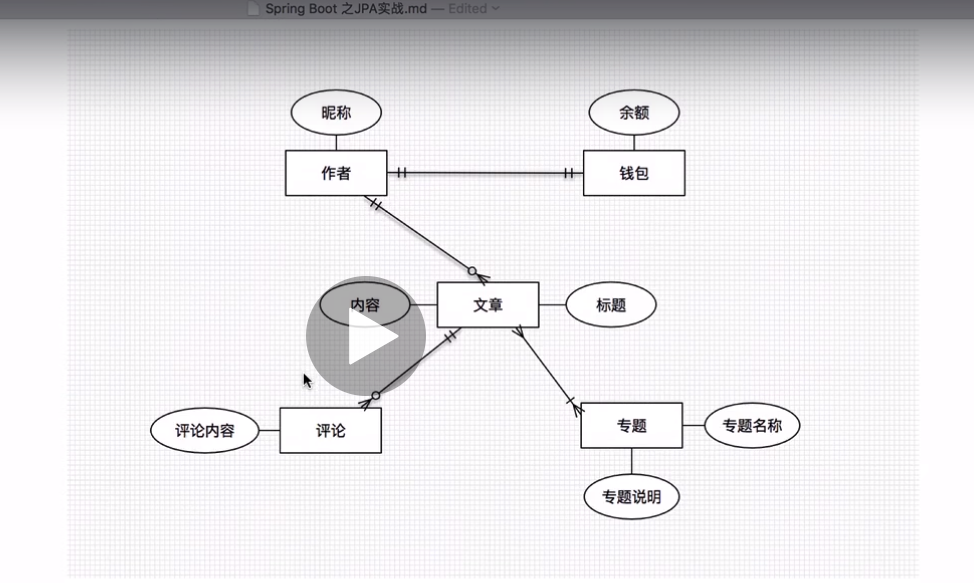
分页排序：



事务处理：

@Transactional

一对一关系？作者和钱包一对一

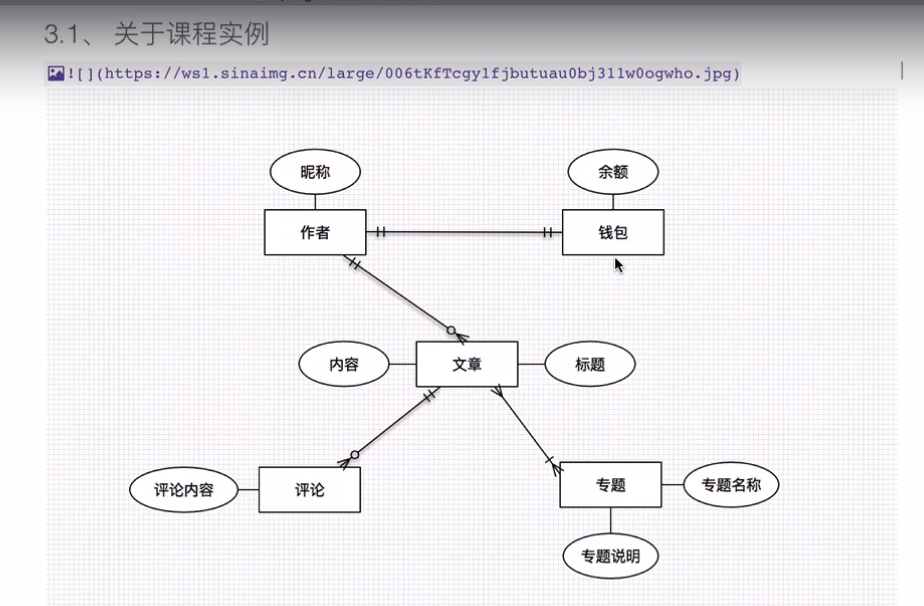


**public class** Author {  
 @Id  
 @GeneratedValue  
 **private** Long **id**;  
 **private** String **nickName**;  
 **private** String **phone**;  
 @Temporal(TemporalType.***DATE***)  
 **private** Date **signDate**;  
 *//级联保存persist,级联更新merge,级联查询不需要设置，只需要设置1对1关系* @OneToOne(cascade = {CascadeType.***PERSIST***, CascadeType.***MERGE***,CascadeType.***REMOVE***})  
 **private** Wallet **wallet**;  
  
 **public** Wallet getWallet() {  
 **return wallet**;  
 }  
  
 **public void** setWallet(Wallet wallet) {  
 **this**.**wallet** = wallet;  
 }  
  
  
  
 **public** Long getId() {  
 **return id**;  
 }  
  
 **public void** setId(Long id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getNickName() {  
 **return nickName**;  
 }  
  
 **public void** setNickName(String nickName) {  
 **this**.**nickName** = nickName;  
 }  
  
 **public** String getPhone() {  
 **return phone**;  
 }  
  
 **public void** setPhone(String phone) {  
 **this**.**phone** = phone;  
 }  
  
 **public** Date getSignDate() {  
 **return signDate**;  
 }  
  
 **public void** setSignDate(Date signDate) {  
 **this**.**signDate** = signDate;  
 }  
  
  
 **public** Author(){  
  
 }

@OneToMany注解

一对一和一对多

关系设置，属性设置，延迟加载？？？

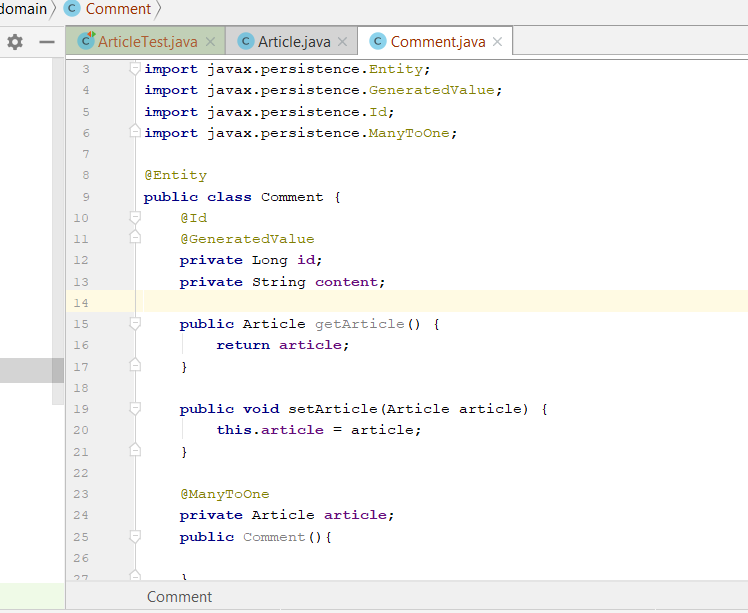


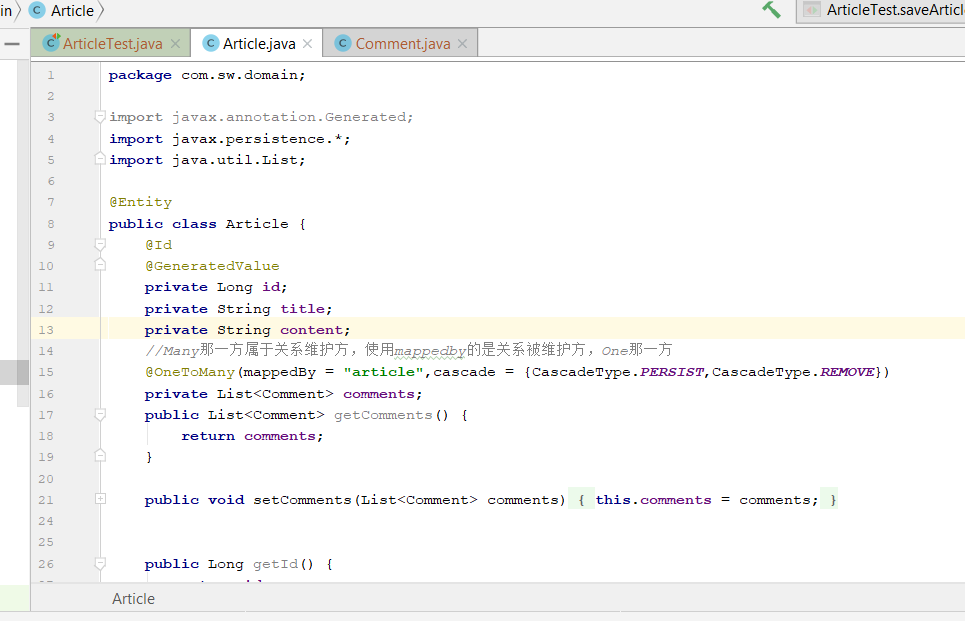
文章和评论 一对多的关系

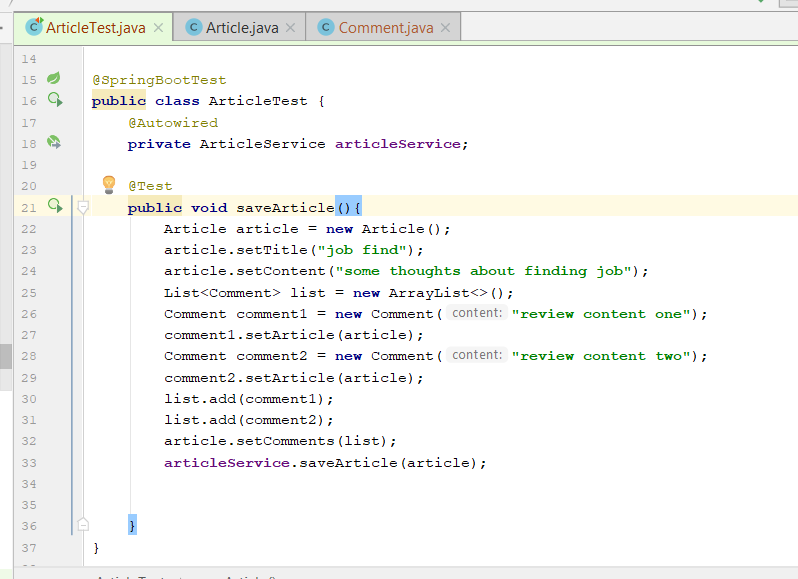
API？？

文章新增 更新 删除

评论文章，新增 删除。







}