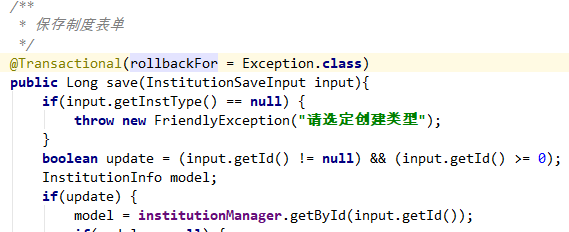
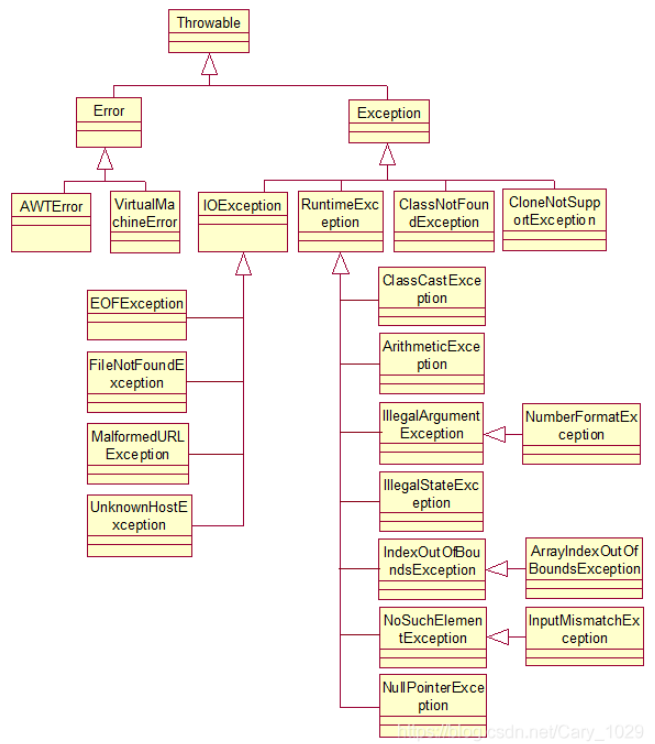
第五天

1. **问题：**
2. 【学习】关于注解@Transactional的rollbackFor属性

在一些方法的上面，我们一般通过@Transactional注解为其添加事务，在出现异常的时候，可以让数据进行回滚操作，如下图所示



Java中的异常



我们可以将异常分为两大类，一个是运行时异常（RuntimeExcepiton）和非运行时异常（Exception中除了RuntimeException及其子类以外的）

也可以将异常分为受控异常（checked exceptions）和不受控异常（unchecked exceptions）

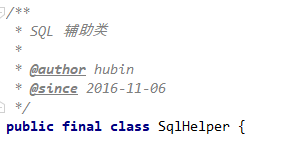
受控异常：就是非运行时异常，即Exception中除了RuntimeException及其子类以外的异常

不受控异常：RuntimeException和Error

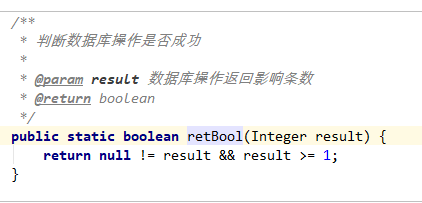
对于spring框架的事物管理默认地只在发送不受控异常（RuntimeException和Error）才会进行事务回滚，也就是说，当事务方法抛出受控异常（Exception中除了RuntimeException及其子类外的异常）时是不会进行事务的回滚的。Rollbackfor属性通过配置rollbackfor=Exception.class,当发生受控异常时，事务也能够进行回滚。

1. 【学习】sqslhelper

SqlHelper是一个sql辅助类，可以使代码更简洁，更有可读性。



比较常用的一个方法是retBool(Integer result)



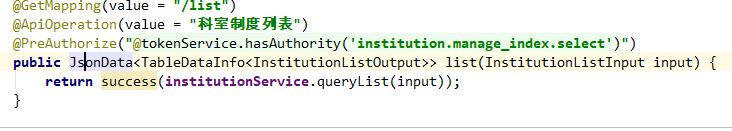
用于判断数据库的操作是否成功

如mp自带的service中的插入操作save中，mp中的insert返回的是受影响的记录条数，然后可以通过sqlhelper的retbool方法返回是否操作成功，如果成功返回1，如果不成果返回0



1. 驳回状态先删除老的审批看不懂
2. 【学习】分页查询的学习

分页在controller层的返回类型是JsonData<TableDataInfo<InstitutionListOutput>>



由于在controller层中我们添加了该注释



所有的数据再返回给前端的时候，都会转换成json格式，而这个JsonData是我们自定义的返回类型

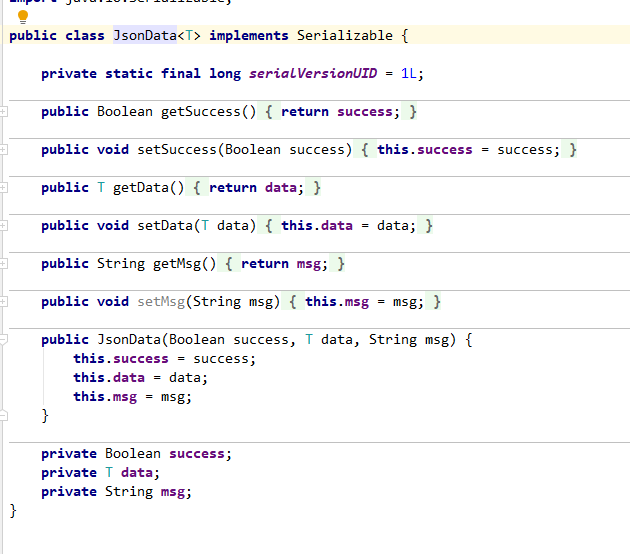
其有三个返回字段，分贝是success，data，msg，

**Success：**true表示调用接口成功 false:调用接口失败

**Data：**接口放回的数据都放在这里

**Msg：**接口返回的提示信息都放在这里

并且我们看到其实现了Serializable，说明其需要进行序列化，将该对象通过接口从后盾传到前端或是从前端传到后端



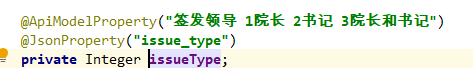
TableDataInfo：表格分页数据对象



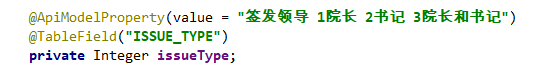
1. **需求解决：**
2. 新增/修订制度，新增选项“签发领导”：院长/书记/院长和书记，对应审批流中的院长或书记，多人的话是会签

【科室制度信息保存请求中的相关修改】

输入类的字段接口增加swagger注释



数据表对应实体类NFINSTI\_INFO科室制度信息中的签发领导字段的增加



【科室制度信息提交请求中的相关修改】

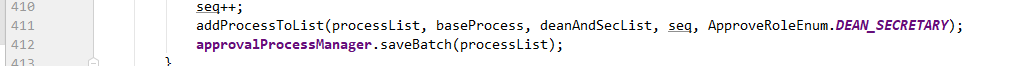
1. 院长
2. 书记
3. 院长和书记



书记和院长信息的获取



书记和院长信息的添加



2.选择修订制度时，制度说明修改为修订内容

前端解决