20200921-20200930：任务汇总：

【1】[开发]用于从rptdesign文件获取报告参数的API实现，请参阅“从报告文件获取参数的

数据处理”

从rptdesign文件中读取参数【2d】

【2】[开发]用于生成报告并转发给收件人的API实现，请参阅“生成报告输出文件并转发给收件人的数据处理”；【3d】

1.将动态参数填充到rptdesign文件中

2.集成通知服务转发输出文件

3.迁移重用代码

4.渲染rptdesign文件以输出PDF，WORD文件

20200921-22

今日任务：从rptdesign读取参数（待定）

20200921：

今日任务：~~封装相应的reportRequest以及response，使用枚举策略的方法~~，

20200922：

今日任务：~~pause~~与~~resume~~接口的实现;难点：~~quartz中jobdatamap的再次放入，直接使用sql进行存放~~

：改jobstatus为P，并且插入新的记录（job以及动态参数）：方法一：一个originalId

20200923：上午code-review

今日任務：

~~1、RequestEnum使用策略模式：Istratergy以及繼承抽象 （mid）（困难：如何根据type选择相应的策略方法）（待定）~~

~~2、验证：JsonSchema放入lion中，针对不同的schema抽象一个方法，可以扩展一个属性，通过属性判断相应的key，然后得到相应的schema判断 （hard）~~

3、~~邮箱验证~~，~~内部userId验证 （userId是一个字符串，没有规范，需要调用接口来判断其是否存在对应的user，如果user的邮箱没有填写，那么也不会发送；建议在发送的时候在进行逻辑校验） （待定）~~ （mid）

~~4、Cron-》CronTabExpression ，并转移 easy）~~

~~5、CommonUtil 改成静态，外部不可实例化；改成单例（枚举单例）~~

~~6、Result使用ResultUtil.Ok; （easy）~~

~~8、model中的Frequency改为DTO （easy）~~

~~9、StringUtils.equalsIg(\*\*\*\*, \*\*\*) （easy）~~

~~10、eventLog增加details （mid） 15.14~~

~~11、转移DyTimeEnum到2~~

~~12、删除SubscriptionType~~

13、enum修改（使用循环获得相应的enum，在执行，方法可行性待定） （mid）

16.53

Enum书写

@AllArgsConstructor  
    public enum FILE\_TYPE {  
        RPTDESIGN("R"), RPTLIBRARY("L"), OUTPUT("O");  
        @Getter  
        public String fileType;  
  
        public static FILE\_TYPE fromValue(String type) {  
            return Stream.of(FILE\_TYPE.values()).filter(fileType ->  
                    StringUtils.equals(fileType.getFileType(), type)  
            ).findFirst().get();  
        }  
    }

使用策略模式中的bean的获取方法

@Component("report")  
public class AlarmStrategry  extends BaseStrategry implements IRunStrategry   {  
    @Override  
    public void callDownstream(List<>,) {  
        IRunStrategry runStrategry =  SpringContextHolder.getBean("report");  
        runStrategry.callDownstream();  
    }  
}

20200923

今日任务：将动态参数填充到rptdesign文件中

20200924

今日任务：渲染rptdesign文件以输出PDF，WORD文件

20200927

今日任务： 集成通知服务转发输出文件

20200928

今日任务：迁移重用代码