主要负责备品室与危化品室的交互系列功能（主要为货物与单据的转移以及与仓库的关系）因为备品室里面不允许长时间存放危化品，所以在用户需要危化品的时候需要在备品室进行货物的交付，所以需要与危化品系统产生关联。

功能：

危化品申请单：

用户如果需要危化品需要先进行申请，也即是新建一个危化品申请单在SPM中，这里需要输入一些参数比如当前申请人的费用代码，部门等。

当输入料号的时候会进行物料推荐，这里推荐接口使用的是SpmMaterial接口，最开始的需求为，用户选择的推荐的料号需要进行库存比对，比对的是当前用户最顶层的部门的费用代码。在代码中需要沿着super\_dep\_id字段向上找知道此字段为0，

*public Integer* getTopDepId(*String depId*) {  
 *Integer* topDepId = *null*;  
 *Integer* curDepId = *Integer*.*parseInt*(*depId*);  
 *while* (*ObjectUtil*.*isNotEmpty*(curDepId)) {  
 *HCSDepDTO* depInfoByDepId = basicMapper.getDepInfoByDepId(curDepId);  
 *if* (*ObjectUtil*.*isEmpty*(depInfoByDepId)) {  
 *break*;  
 }  
 *Integer* superDep = depInfoByDepId.getSuperDep();  
 *if* (superDep.equals(0)) {  
 topDepId = curDepId;  
 *break*;  
 }  
 curDepId = superDep;  
 }  
 *return* topDepId;  
}

后续在选择料号时不需要进行库存的展示与判断（将库存的管理权交给HCS，解耦），在用户选择了一个料号之后该料的详细信息会自动展示，这个调用的是也是该接口(getEntryInfo废弃), 这里用户不允许修改带出来的物料信息只允许展示。之后就要输入申请数量，之后进行提交。

提交危化品申请单：

之前需要对用户的申请数量与库存进行校验，现在将控制权交给HCS，在每一个申请单建立的时候的开始状态都是开立，该单据共有5种状态，开立，提交，送货，驳回，完成。之后将前端传过来的参数入表即可。

状态的修改：

已经开立的单子可以进行提交，如果当前用户是备品室人员可以进行驳回，开立状态可以进行删除，编辑等操作。驳回状态可以正常提交，提交状态不允许删除与编辑，在驳回状态下点击编辑后会自动变成开立状态，

.set(updateVo.getDocStatusCode().equals("SPM\_HCS\_DISALLOWANCE"), SpmbuHCSApply::getDocStatusCode,  
 "SPM\_HCS\_BILL\_CREATED")  
.set(updateVo.getDocStatusCode().equals("SPM\_HCS\_DISALLOWANCE"), SpmbuHCSApply::getDocStatus,  
 spm\_hcs\_bill\_created)

与HCS交互部分：

单据汇总后抛转给HCS：

在一个单据是提交且不是被驳回状态下，该单据可以被抛转给HCS，抛转之后HCS会正常进行备货与运送。

*try* {  
 result = *HttpUtil*.*createPost*(hcsTransportationUrl)  
 .header("Content-Type", "application/json;charset=UTF-8")  
 .setConnectionTimeout(30 \* 1000)  
 .setReadTimeout(180 \* 1000)  
 .body(*JSONUtil*.*toJsonStr*(hcsApplyToHcsVOList))  
 .execute()  
 .body();  
 log.info("调用HCS接口：::requestJson={},url={},result={},current-time:{}",  
 *JSONUtil*.*toJsonStr*(hcsApplyToHcsVOList),  
 hcsTransportationUrl, result, *System*.*currentTimeMillis*());  
 wmsResponse = *JSON*.*parseObject*(result, *WmsResponse*.*class*);  
} *catch* (*Exception e*) {  
 log.error(">>>>>>>>>>>>>>>>调用HCS接口 ERROR:{}", *e*.getMessage(), *e*);  
 *throw new* CloudmesException(*SpmMouldResultCode*.FAILED\_TO\_CALL\_WMS.getCode(),  
 *MessageUtils*.*get*(*SpmMouldResultCode*.FAILED\_TO\_CALL\_WMS.getLocalCode()));  
}

抛转时将HCS需要且申请单里面有的字段抛转过去，比如申请数量，料号，载具等等，那边会在HCS库存中找到这个料并且进行备料扣库存等。

在HCS处理完之后，HCS会传回来一个需求单号，也就是docFromHCS，这个是SPM汇总之后，那边生成的一个单子所以也就会共用一个需求单号，同是修改spmbu\_hcs\_delivery\_apply表中的post\_hcs\_flag字段与post\_hcs\_message字段，若post\_hcs\_flag为1那么为抛转成功，则修改该单据的状态为送货中

送货单：

生成了一个送货单之后，HCS会不断的根据需求单号与来源单号与料号进行送货，会不断的将此次备货的数量等参数不断的抛转给SPM，SPM进行接受，因为一个申请可能不是一次性备货完成的，所以在该接口中要不断的累加目前已接收数量，同时在接口中要不断判断该次备货量是否大于库存（SPM系统中为了方便对应的HCS的库存表的备份），与该次备货量的累加是否已经大于了申请数量，若都满足则对当前已接收数量进行累加。同时减库存。

接收量只是目前在SPM中的危化品的数量，所以会有一个危化品领用列表，该列表会将本次需要送货的量计算并展示出来（目前已接收量 – 已送货量）currentDeliveriedQty,

*Integer* deliveredQty = *applyList*.getDeliveredQty();  
*Integer* aptQty = *applyList*.getAptQty();  
*Integer* remainTransportationNum = aptQty - deliveredQty;

因为这个接口是在确认领用的时候进行调用的，也可以理解为标识，并且领用为一次性全领用，所以将当前aptQty也即是接受量更新到当前送货量上，当危化品室又备了货也就是SPM的该单子的备货量又增加时，送货列表会自动计算出此次应送货量并展示此次应送货量，正常送货即可。（此列表只展示应该送货的（aptQty > deliverideQty）并且抛转HCS成功的且单据状态不为开立和驳回的单据）此接口分为App端和Web端，在Web端会进行输入工号来判断该用户是否存在，在App端会进行刷脸来判断。

完成：

当送货量等于备货量等于申请数量的时候将单据状态转成完成状态，证明该次交互完成。

治具实验板管理库存报表获取 + 导出

调用spmbu\_stock\_barcode表，加上一些搜索字段

使用EasyExcel进行导出，配置些多语言字段即可。isBorrow就是借出也即是未在库，导出的时候改下字段即可

行事历管理（HCS拋SPM接口）

在HCS系统中会发布行事历，发布的行事历会通过接口抛到SPM系统中，在SPM中判断当前时间是否大于该行事历的结束时间，如果大于那么

前端会展示一个该行事历的remark字段，通过isDisplay字段