

# 邮件模板（专有云）

加代码的，需要需要判断后，才进行修改

```
if (projectProperty.getEnviroment().equals(Constants.ENVIROMENT_PRIVATE_CLOUD))
```

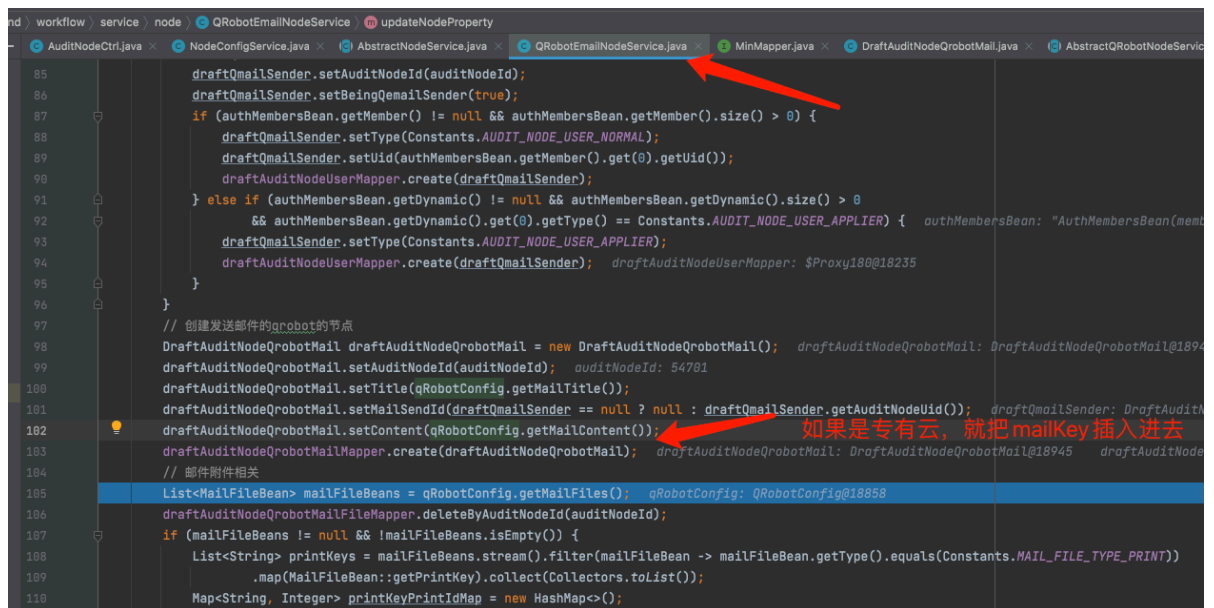
## Q-Robot内

1. 新增：Q-Robot：通过appkey去查对应的form，根据formId找到所有的邮件模板，选择对应的模板，用emailKey（在audit\_node\_qrobot\_mail新增）与之关联起来，在更新配置中，不需要修改
2. 查询：listEmailTemplates 根据appKey获取所有邮件模板，复用即可
3. 删除：Q-Robot节点：直接根据audit\_node\_id进行删除，直接把 audit\_node\_qrobot\_mail 节点中对应模板中的内容删除掉了，这部分不需要修改
4. 修改（更新配置）：传入新的mailKey进行重新绑定

流程内q-robot调用 /app/{appKey}/auditNode 接口，传递的参数 AuditNodeDetail 中 qRobotConfig 添加一个字段 mailKey

### 1. 草稿

1. 逻辑是删除草稿后，再插入进去。那么只要是专有云的，保存的时候就把mailKey设置到进去，存到表中



```
85     draftQmailSender.setAuditNodeId(auditNodeId);
86     draftQmailSender.setBeingQmailSender(true);
87     if (authMembersBean.getMember() != null && authMembersBean.getMember().size() > 0) {
88         draftQmailSender.setType(Constants.AUDIT_NODE_USER_NORMAL);
89         draftQmailSender.setUid(authMembersBean.getMember().get(0).getUid());
90         draftAuditNodeUserMapper.create(draftQmailSender);
91     } else if (authMembersBean.getDynamic() != null && authMembersBean.getDynamic().size() > 0
92         && authMembersBean.getDynamic().get(0).getType() == Constants.AUDIT_NODE_USER_APPLIER) {
93         draftQmailSender.setType(Constants.AUDIT_NODE_USER_APPLIER);
94         draftAuditNodeUserMapper.create(draftQmailSender);
95     }
96 }
97 // 创建发送邮件的qrobot节点
98 DraftAuditNodeQrobotMail draftAuditNodeQrobotMail = new DraftAuditNodeQrobotMail();
99 draftAuditNodeQrobotMail.setAuditNodeId(auditNodeId);
100 draftAuditNodeQrobotMail.setTitle(qRobotConfig.getMailTitle());
101 draftAuditNodeQrobotMail.setMailSendId(draftQmailSender == null ? null : draftQmailSender.getAuditNodeUid());
102 draftAuditNodeQrobotMail.setContent(qRobotConfig.getMailContent());
103 draftAuditNodeQrobotMailMapper.create(draftAuditNodeQrobotMail);
104 // 邮件附件相关
105 List<MailFileBean> mailFileBeans = qRobotConfig.getMailFiles();
106 draftAuditNodeQrobotMailMapper.deleteByAuditNodeId(auditNodeId);
107 if (mailFileBeans != null && !mailFileBeans.isEmpty()) {
108     List<String> printKeys = mailFileBeans.stream().filter(mailFileBean -> mailFileBean.getType().equals(Constants.MAIL_FILE_TYPE_PRINT))
109         .map(MailFileBean::getPrintKey).collect(Collectors.toList());
110     Map<String, Integer> printKeyPrintIdMap = new HashMap<>();
```

- 2.
3. 模板只能选择一个，附件有多个，用mailKey去 email\_template\_file 找对应的附件（多个），放入到mailFileBeans中

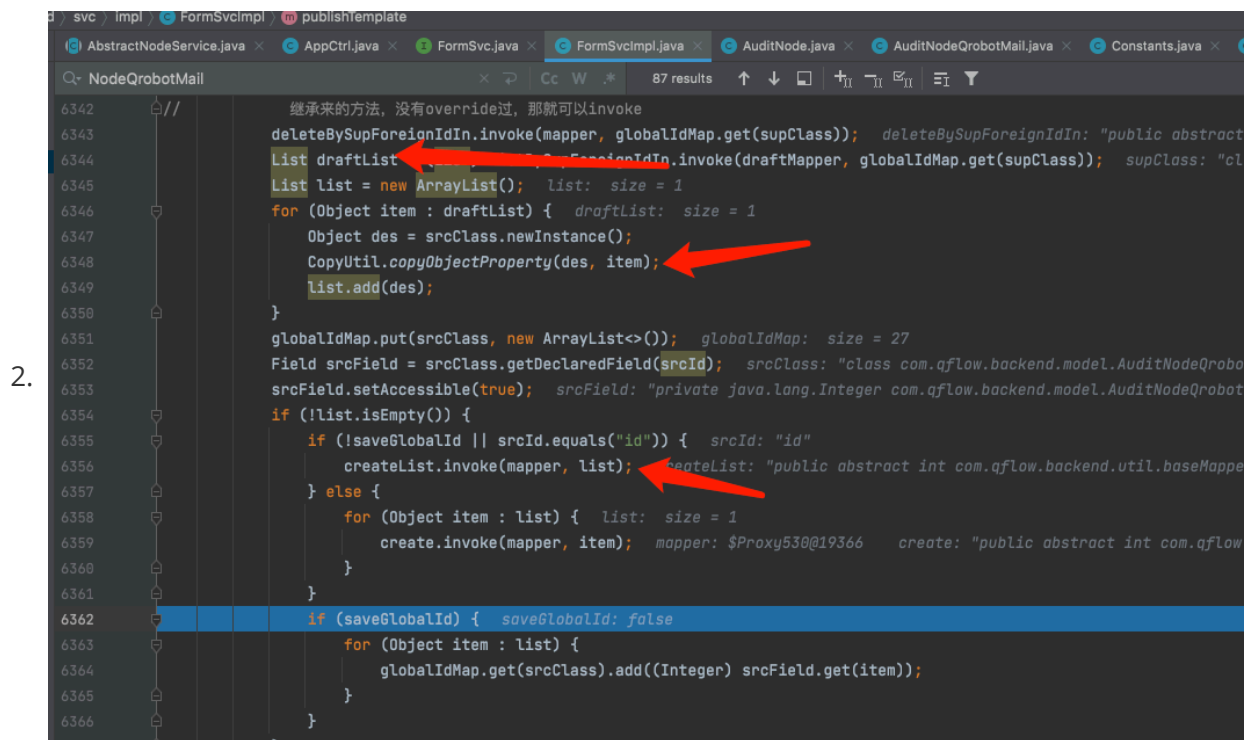
```
QRobotExecuteSvcImpl.java x QRobotEmailNodeService.java x DraftAuditNodeQrobotMail.java x AuditNodeCtrl.java x AbstractNodeService.java x QrobotSvc.java
92    && authMembersBean.getDynamic().get(0).getType() == Constants.AUDIT_NODE_USER_APPLIER) { authMembersBean
93        draftQmailSender.setType(Constants.AUDIT_NODE_USER_APPLIER);
94        draftAuditNodeUserMapper.create(draftQmailSender); draftAuditNodeUserMapper: $Proxy18@18235
95    }
96    }
97    // 创建发送邮件的qrobot的节点
98    DraftAuditNodeQrobotMail draftAuditNodeQrobotMail = new DraftAuditNodeQrobotMail(); draftAuditNodeQrobotMail: Draft
99    draftAuditNodeQrobotMail.setAuditNodeId(auditNodeId);
100    draftAuditNodeQrobotMail.setTitle(qRobotConfig.getMailTitle());
101    draftAuditNodeQrobotMail.setMailSendId(draftQmailSender == null ? null : draftQmailSender.getAuditNodeUid()); draft
102    draftAuditNodeQrobotMail.setContent(qRobotConfig.getMailContent());
103    draftAuditNodeQrobotMailMapper.create(draftAuditNodeQrobotMail); draftAuditNodeQrobotMail: DraftAuditNodeQrobotMail
104    // 邮件附件相关
105    List<MailFileBean> mailFileBeans = qRobotConfig.getMailFiles(); qRobotConfig: QRobotConfig@32372 mailFileBeans:
106    draftAuditNodeQrobotMailFileMapper.deleteByAuditNodeId(auditNodeId);
107    if (mailFileBeans != null && !mailFileBeans.isEmpty()) {
108        List<S + ($Proxy318@18233) lFileBeans.stream().filter(mailFileBean -> mailFileBean.getType().equals(Constants.MA
109        .map(MailFileBean::getPrintKey).collect(Collectors.toList());
110        Map<String, Integer> printKeyPrintIdMap = new HashMap<>(); printKeyPrintIdMap: size = 0
111        if (!printKeys.isEmpty()) {
112            List<DraftPrintTemplate> printTemplates = draftPrintTemplateMapper.listByPrintKeyIn(printKeys); printKeys:
113            if (printTemplates != null && !printTemplates.isEmpty()) {
```

4. DraftAuditNodeQrobotMail 中加上mailKey, 从 qRobotConfig 中获取, 执行create就可以

1. 同样, 附件中 DraftAuditNodeQrobotMailFile 也要加上mailKey

2. 非草稿

```
svc > impl > FormSvcImpl > publish
AbstractNodeService.java x AppCtrl.java x FormSvc.java x FormSvcImpl.java x AuditNode.java x AuditNodeQrobotMail.java x Constants.java
NodeQrobotMail
8844    * 复制远程查询相关内容
8845    */
8846    publishTemplate(QuestionRemoteLookUp.class, Form.class, globalIdMap);
8847    publishTemplate(RemoteLookHeaderBody.class, Form.class, globalIdMap);
8848    publishTemplate(QuestionRemoteLookUpVariables.class, Form.class, globalIdMap);
8849    publishTemplate(QuestionRemoteLookUpAlias.class, Form.class, globalIdMap);
8850    publishTemplate(AuditNodeQuestionPermission.class, AuditNode.class, globalIdMap);
8851    publishTemplate(AuditNodeSubmitRecalcQuestion.class, AuditNode.class, globalIdMap);
8852    publishTemplate(AuditNodeQueRelation.class, AuditNode.class, globalIdMap);
8853    publishTemplate(AuditNodeQueFilter.class, AuditNode.class, globalIdMap);
8854    publishTemplate(AutoJudge.class, AuditNode.class, globalIdMap);
8855    publishTemplate(AuditNodeQueCheck.class, AuditNode.class, globalIdMap);
8856    publishTemplate(AuditNodeQrobotMail.class, AuditNode.class, globalIdMap);
8857    publishTemplate(AuditNodeQrobotMailFile.class, AuditNode.class, globalIdMap); globalIdMap: size = 27
8858    publishTemplate(AuditNodePrintTemplateRet.class, AuditNode.class, globalIdMap);
8859    publishTemplate(AuditNodeTimeoutNotifier.class, AuditNode.class, globalIdMap);
8860    /** 复制web_hook相关内容 */
8861    publi + {Class@11770} "class com.qflow.backend.model.AuditNodeTimeoutNotifier"... Navigate
8862    publishTemplate(WebhookHeaderBody.class, AuditNode.class, globalIdMap);
8863    publishTemplate(WebhookParse.class, AuditNode.class, globalIdMap);
8864    publishTemplate(WebhookParseValue.class, AuditNode.class, globalIdMap);
8865    publishTemplate(WebhookGlobalConfigAliasBinding.class, AuditNode.class, globalIdMap);
```

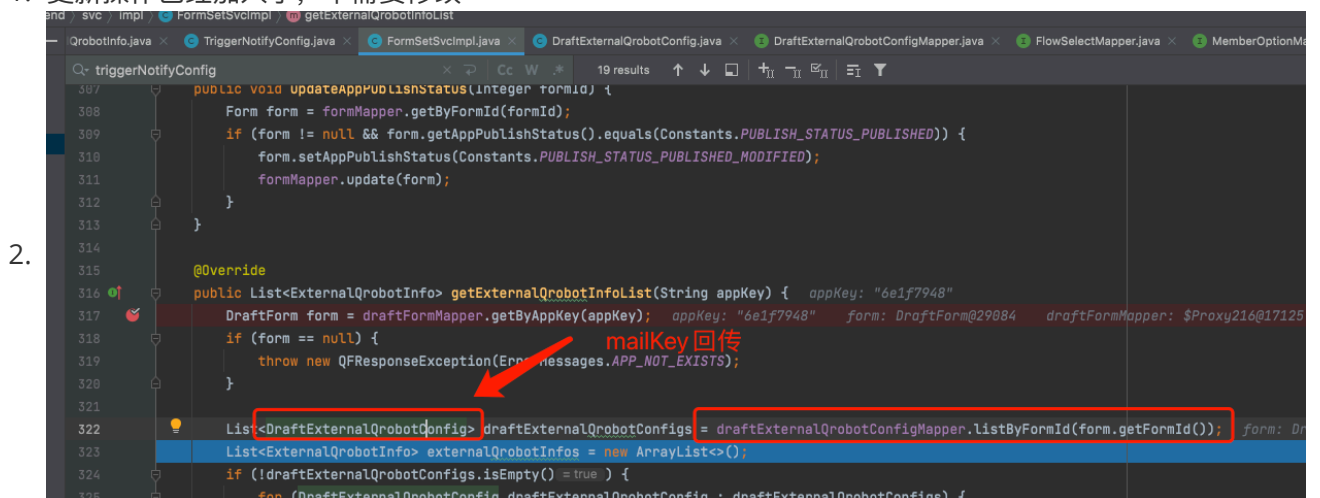


3. 直接复制，不需要修改

## Q-Robot外

1. 获取：

1. 更新操作已经加入了，不需要修改



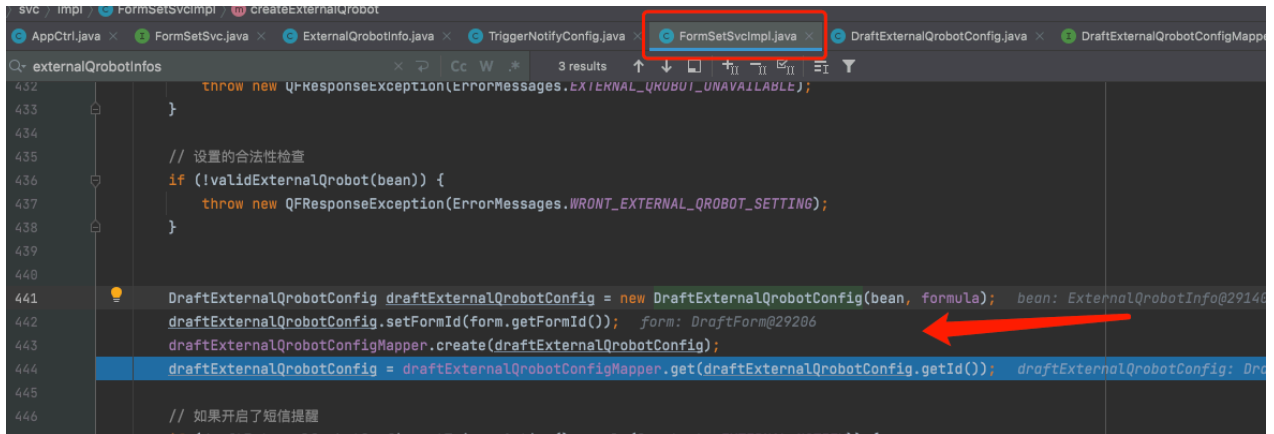
2. 新增

1. 与更新做的操作相同

3. 更新

1. /app/{appKey}/externalQrobot 接口，传递的参数 ExternalQrobotInfo 中 triggerNotifyConfig 中添加一个字段 mailKey，用于保存邮件模板的mailKey
2. DraftExternalQrobotConfig.java 中需要添加上mailKey，更新的时候存入数据库中

3.



是公有云的时候，mailKey是否需要置空？还是说一起保存上去

#### 4. 删除

1. 根据id整行数据删除，模板不动，不做修改

## 流程内外发送邮件

### Q-Robot内发送邮件

- 使用邮件模板发送邮件后，邮件发送部分代码需要改动
- Q-Robot内发送邮件代码所在地：
  - QEmailSendSvc.java sendEmail
  - QrobotSvcImpl.java.qRobotSendEmail qRobot发送邮件

1. AuditNodeQrobotMail 中加上mailKey
2. 如果查询查不到，就在auditRecord的qrobotErrorType加上“没有邮件模板 = 18”，并且直接返回（qr发送邮件执行失败）

```
auditRecord.setMsgCount(0);
auditRecord.setQrobotErrorType(没有邮件模板 = 18); //QROBOT_ERROR_TYPE_NO_EMAIL_TEMPLATE = 18
auditRecord.setExecuteStatus(Constants.EXECUTE_STATUS_FAIL);

checkResult.setAuditRecord(auditRecord);
checkResult.setqRobotOpResult(new QRobotOpResult(""));
checkResult.setBeingProcessFail(true);
return checkResult;
```

```
svc \impl \QrobotExecuteSvcImpl \checkQMailStatus
QrobotExecuteSvcImpl.java ThinSelectMapper.java PermissionChecker.java PermissionCheckerImpl.java AuditRecord.java AuditNodeQrobotMail.java Apply.java QrobotSvcImpl.java QRobot
211 List<String> realSenderEmail = senderEmailCandidate.stream().filter(email -> emailValidator.isLegalEmailAddress(email)).collect(Collectors.toList());
212 if (realSenderEmail.isEmpty()) {
213     auditRecord.setMailCount(0);
214     auditRecord.setQrobotErrorType(Constants.QROBOT_ERROR_TYPE_EMAIL_NO_SENDER);
215     auditRecord.setExecuteStatus(Constants.EXECUTE_STATUS_FAIL);
216     customNotifySvc.processQrobotFail(apply);
217     checkResult.setAuditRecord(auditRecord); auditRecord: "AuditRecord(auditRcdId=null, auditNodeId=54701, applyNodeId=null, applyId=1803692, ticket=null, em
218     checkResult.setQrobotOpResult(new QRobotOpResult( receivers: ""));
219     checkResult.setBeingProcessFail(true);
220     return checkResult;
221 } else {
222     senderEmail = realSenderEmail.get(0); realSenderEmail: size = 1
223     checkResult.setSenderEmail(senderEmail); checkResult: QMailCheckResult@30278 senderEmail: "897988813@qq.com"
224
225     // 替换邮箱content和title中的queId
226     String mailContent = CustomPushContentConverter.replaceQueIdInPushContent(auditNodeQrobotMail.getContent(), applyDetailInfo.getAnswers()); auditNodeQrobotMail
227     String mailTitle = CustomPushContentConverter.replaceQueIdInPushContent(auditNodeQrobotMail.getTitle(), applyDetailInfo.getAnswers());
228     String fromName = senderEmail.split( regex: "@" )[0];
229     checkResult.setMailContent(mailContent);
230     checkResult.setMailTitle(mailTitle);
231     checkResult.setFromName(fromName);
232     // 获取附件内容
233     // 上传文件字段附件
234     Map<String, File> nameFileMap = new HashMap<>();
235     int fileSize = 0;
236     boolean overLimit = false;
```

流程内中，私有云的话，那么用 mailKey 到模板中去查到 content 和 title，名字进行替换



3.

```
svc \impl \QrobotExecuteSvcImpl \checkQMailStatus
QrobotExecuteSvcImpl.java Constants.java ThinSelectMapper.java PermissionChecker.java PermissionCheckerImpl.java AuditRecord.java AuditNodeQrobotMail.java
33 // 上传文件字段附件
34 Map<String, File> nameFileMap = new HashMap<>(); nameFileMap: size = 0
35 int fileSize = 0; fileSize: 0
36 boolean overLimit = false; overLimit: false
37 List<AuditNodeQrobotMailFile> mailFiles = auditNodeQrobotMailFileMapper.listByAuditNodeId(auditNodeId); auditNodeId: 54701 mailFi
38 if (mailFiles != null && !mailFiles.isEmpty()) {
39     Map<Integer, List<AuditNodeQrobotMailFile>> typeMailFileMap = mailFiles.stream().collect(Collectors.groupingBy(AuditNodeQrobotMail
40     List<AuditNodeQrobotMailFile> fileMailFiles = typeMailFileMap.get(Constants.MAIL_FILE_TYPE_QUE); fileMailFiles: size = 1
41     List<AuditNodeQrobotMailFile> printMailFiles = typeMailFileMap.get(Constants.MAIL_FILE_TYPE_PRINT); typeMailFileMap: size = 1
42     List<Integer> queIds = fileMailFiles != null ? fileMailFiles.stream().map(AuditNodeQrobotMailFile::getQueId).collect(Collectors.to
43     List<Integer> printIds = printMailFiles != null ? printMailFiles.stream().map(AuditNodeQrobotMailFile::getPrintId).collect(Collectors.to
44     if (!queIds.isEmpty()) {
45         List<Question> questions = questionMapper.listByQueIdInAndBeingDeletedFalse(queIds); questions: size = 1 questionMapper:
46         List<Integer> tableQueIds;
47         if (questions != null && !questions.isEmpty()) {
48             queIds = questions.stream().map(Question::getQueId).collect(Collectors.toList()); queIds: size = 1
49             tableQueIds = questions.stream().map(Question::getSupId).filter(Objects::nonNull).collect(Collectors.toList()); question
50             if (!tableQueIds.isEmpty()) {
51                 tableQueIds = questionMapper.listByQueIdInAndBeingDeletedFalse(tableQueIds).stream().map(Question::getQueId).collect(
52             }
53     } else {
```

附件发送，在公有云中，只有模板 emailKey，所以根据 auditNodeId 在 audit\_node\_qrobot\_mail\_file 找到 emailKey，再根据 emailKey 在 email\_template\_file 中找到对应的 type, queId, printIds，重新塞到 mailFiles 中

## Q-Robot 外发送邮件

1. 这里会用动态 sql 查出 mailKey

```
svc \impl \ExternalQrobotSvcImpl \processNewData Lambda
ApplySvcImpl.java AuditSvcImpl.java ExternalQrobotSvc.java AuditSvc.java ApplyCtrl.java ApplySvc.java ExternalQrobotSvcImpl.java Ap
131 private QuartzTaskGateway quartzTaskGateway;
132
133 private final static Logger logger = LogManager.getLogger();
134
135 @Override
136 public void processNewData(Apply apply) {
137     List<ExternalQrobotConfig> externalQrobotConfigList = externalQrobotConfigMapper.listByFormId(apply.getFormId());
138     externalQrobotConfigList.forEach(externalQrobotConfig -> {
139         processNewDataWithCertainExternalQrobot(apply, externalQrobotConfig);
140     });
141 }
142
143 @Override
```

会查出 mailKey







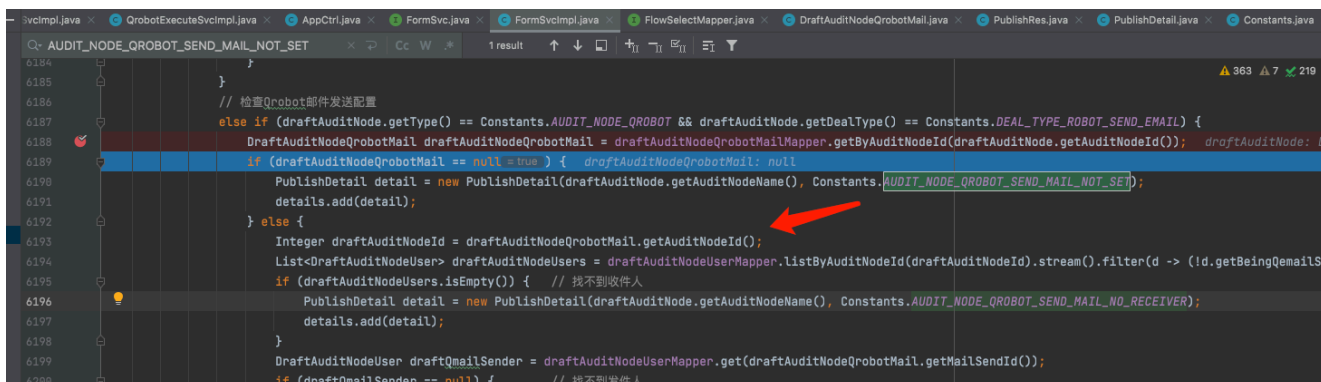
2. 私有云中，选择了模板，这个时候只有mailKey（公有云中还是会有content），所以需要通过mailKey来找到NotifyContent，然后传入pushContent中去

## 邮件模板删除：

- 删除后Q-Robot会失效，回传一个报错信息
  - /app/34fd7730/apply/1803573/auditRecord/3827755?role=1，加上qrobotErrorType：没有邮件模板 = 18
  - /app/34fd7730/publish，details里的加一个类型auditNodeName为：Q-Robot · 发送邮件  
'errorType = 16'
    - 没有设置邮件模板：AUDIT\_NODE\_QROBOT\_SEND\_MAIL\_NO\_TEMPLATE = 16
    - 没有邮件模板：QROBOT\_ERROR\_TYPE\_NO\_EMAIL\_TEMPLATE = 18

## 发布应用，模板删除后（流程内）

1. 在私有云中，如果没有mailKey，那么就是没有设置邮件模板



```
PublishDetail detail = new PublishDetail(draftAuditNode.getAuditNodeName(),  
Constants.没有模板 =18);  
//  
details.add(detail);
```

2. 如果模板删除掉，可以发布成功，但是mailKey对应的模板就找不到，这个时候需要在

app/34fd7730/apply/1803573/auditRecord/3827755?role=1 中设置提示，但是只是在auditRecord中取的，所有发送的时候没有找到模板，应该设置到 auditRecord中的 qrobotErrorType 去

所以中发送邮件的时候，需要把错误 没有模板 = 18 的code放进去(已在发送邮件的时候加上了)

## 发布应用，模板删除后（流程外）

- 没有提示，不做处理

## 对旧数据处理

### 流程内数据

1. `audit_node_qrobot_mail` 中添加一个逻辑外键 `email_id`（用UUID生成，[python的UUID生成方案](#)）
2. `select * from audit_node an where type = 3 and deal_type = 2`
3. 1. 找到所有节点，根据每一个 `formId` 找到 `audit_node_id`（可能存在多个QRobot节点）
4. `select * from audit_node_qrobot_mail anqm where anqm.audit_node_id in (54291,54405)`
5. 然后把标题和内容放到模版表中
  1. `audit_node_qrobot_mail` —> `email_template`, (`title` —> `title`, `content` —> `content`)
6. 模板附件：主要是与`audit_node_id`关联的
  1. 附件可能没有，查询条数等于0，那么就不把附件加入到模板附件表中

#### ## 流程内

```
select * from draft_audit_node an where type = 3 and deal_type = 2 and an.form_id = '20517'
select * from audit_node an where type = 3 and deal_type = 2 and an.form_id = '20517'
```

#### # Q-Robot节点上邮件的内容

```
select * from draft_audit_node_qrobot_mail danqm where danqm.audit_node_id in (54701,54703)
select * from audit_node_qrobot_mail anqm where anqm.audit_node_id in (54701,54703)
```

#### # Q-Robot节点上邮件的内容的附件

```
select * from draft_audit_node_qrobot_mail_file anqmf where anqmf.audit_node_id in (54701,54703)
select * from audit_node_qrobot_mail_file anqmf where anqmf.audit_node_id in (54701,54703)
```

#### # 邮件模板内容

```
select * from draft_email_template et where et.email_key = '49e34b1aa5834a979b8e031b43ae295a'
select * from email_template et where et.email_key = '49e34b1aa5834a979b8e031b43ae295a'
```

#### # 邮件模板附件内容

```
select * from draft_email_template_file etf where etf.email_key = '49e34b1aa5834a979b8e031b43ae295a'
select * from email_template_file etf where etf.email_key = '49e34b1aa5834a979b8e031b43ae295a'
```

## 修改表结构

```
1. draft_audit_node_qrobot_mail    #表中添加一个逻辑外键 email_id
2. audit_node_qrobot_mail          #表中添加一个逻辑外键 email_id
3. draft_audit_node_qrobot_mail_file #表中添加一个逻辑外键 email_id
4. audit_node_qrobot_mail_file      #表中添加一个逻辑外键 email_id
```

## 流程外数据

- 流程外邮件提醒设置旧数据放在 `external_qrobot_config` 表中的 `notify_content` 字段, 用 `email_status` 状态来区分是否开启了邮件提醒设置
- `external_qrobot_config` 表添加一个 `email_id` 来和表 `email_template` 进行联系, 再将 `notify_content` 字段迁移到 `email_template` 中的 `content` 字段

1. `external_qrobot_config` 表添加一个 `email_id`
2. `external_qrobot_config(notify_content) → email_template(content)`
3. 没有附件? 只有 `content`, 没有 `title`

```
## 流程外
# 流程外邮件内容
select * from draft_external_qrobot_config eqc where eqc.form_id = '20517'
select * from external_qrobot_config eqc where eqc.form_id = '20517'
```

## 修改表结构

```
1. draft_external_qrobot_config #表中添加一个逻辑外键 email_id
2. external_qrobot_config       #表中添加一个逻辑外键 email_id
```

## 注意事项

1. **draft** 和 **非draft** 表中都要做处理
2. 查询表 **不要用** `select *`
3. 备份数据处理的表, 找运维马

## 疑问 (已解决)

- 旧数据迁移, 模板名怎么规定 -----[参考这里](#)
- 所有都是新增模板? 用不用更新----- 都是新增模板
- 对原来数据流程内草稿中的邮件生成的模板是草稿, 还是非草稿的-----放到草稿模板中, 草稿放草稿、非草稿放非草稿表中
- `draft` 中都是为空的, 怎么处理? -----直接填充为空
  - 测试库中 `draft` 和 **非draft** 都有为空的
  - 新增流程内, 必须要有标题和正文, 但是可以没有附件
- 流程外的 Q-Robot 没有附件? 只有 `content`, 没有 `title`? -----`title` 直接为 “新数据提醒” 即可



- 选择了模板后，提醒内容是会展示出来吗，保存的时候数据也会传入过来吗？-----只会有模板名，内容不会展示，图中只是短信的

## 创建Q-Robot



当达到一定的条件时，自动触发动作，如发送提醒、添加数据等。 [了解更多](#)

### 触发动作 \*

发送提醒



1、触发时间为Q-robot执行失败时，提醒每5分钟内只会发送一次。

2、导入数据不会触发提醒

### 提醒方式 \*

☐ 微信提醒 需先关注微信服务号并绑定轻流账号 [如何绑定?](#)

☒ 邮件提醒 邮件提醒需要消耗邮件份额 [了解详情](#)

自定义邮件模板A



☐ 预览

☒ 短信提醒 仅可选择短信模板作为提醒内容，且会消耗短信份额 [了解详情](#)

请选择短信模版



### 提醒内容 \*

添加字段

请输入提醒内容



### 提醒谁 \*



上一步

确定