## 电子签章

### 原理和流程图

#### 原理描述

将本系统需要签署的合同pdf文件上传到e签宝平台，之后调用e签宝平台的流程签署相关api接口，完成文件签署，签署完成后下载已签署的文件信息到本系统中。

#### 流程图

图形用户界面, 应用程序, Teams

描述已自动生成

### 输入项描述

整体流程的输入项为：待签约的合同附件信息,客户信息

具体实现的输入项为：

第一步上传合同附件的输入项：ESignUtils. getESignFileId（String fileUrlPath,String fileName）,其中fileUrlPath为待签约合同附件的完整存放路径，fileName为待签约合同附件的文件完整名称,必须带上文件扩展名如：合同1.pdf（可以转换，但是建议只传pdf文件，避免转换格式错误或时间过长）

第二步发起签署流程的输入项：ESignUtils. beginSignContractFlow(String fileId,String fileName,String contractName, String notifyUrl,String noticeTypes,String psnAccount,String custName,String custCard),其中fileId为E签宝生成的附件id，fileName为E签宝生成的附件名称，contractName为签署流程主题，notifyUrl为回调接口地址, noticeTypes为通知方式（""默认 ""不通知 "1"短信通知 "2"邮件通知 "1,2"短信和邮件通知），psnAccount为客户联系方式，使用手机号或邮箱，custName为客户姓名，custCard为客户证件号

第三步获取待签署的合同文件链接地址的输入项ESignUtils. getSignContractUrl(String signFlowId,String linkInfo,String companyName)，其中signFlowId为e签宝平台流程id, linkInfo为客户联系方式，用客户手机号或客户邮箱地址，companyName为公司名称

第四步下载已签署文件的的输入项ESignUtils.getSignContractInfo（String signFlowId），其中signFlowId为e签宝平台生成的签署流程id

### 输出项描述

调用E签宝平台上传本地文件接口（ESignUtils. getESignFileId） 输出项为：E签宝生成的附件id

调用E签宝平台基于文件发起签署接口（ESignUtils. beginSignContractFlow） 输出项为：E签宝生成的签署流程id

调用E签宝平台获取签署页面链接接口（ESignUtils. getSignContractUrl） 输出项为：E签宝生成的签署文件地址

调用E签宝平台下载已签署文件及附属材料接口（ESignUtils. getSignContractInfo） 输出项为：E签宝存放的已签署文件的下载地址

### 过程描述

#### 用户点击合同管理列表中某一条合同的线上签约按钮之后会提交合同审批流程，流程审批通过后将本系统待签署的合同文件调用e签宝平台上传本地文件api接口，返回文件id，记录调用日志信息保存到三方平台调用日志表，返回的文件id保存到电子合同表

#### 调用e签宝平台基于文件发起签署的api接口，调用成功后返回签署流程id

#### 根据签署流程id获取平台签署地址，有短有效url和永久有效url

#### 通过推送任务以邮件的方式将签署地址发送给用户,用户点击链接跳转进行合同签署

#### 签署过程中e签宝平台将回调本系统通知签署状态变更以及签署结果，本系统收到通知后根据业务类型判断若当前流程状态为未结束状态，则更新电子合同表中的流程，若是已完成状态则调用下载已签署的文件api接口获取已签署文件的下载地址，并将其下载地址保存到电子合同表中

### 伪代码

#### 新增表结构

##### 电子合同表(合同线上签约)【预留】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表名** | **tb\_electronic\_contract** | **表中文名** | **电子合同表** | | |
| **字段名** | **字段描述** | **数据类型** | **字段长度** | **输入类型** | **备注** |
| Pkid | 主键 | varchar | 32 | 系统赋值 |  |
| contract\_pkid | 合同表pkid | varchar | 32 | 系统赋值 |  |
| init\_verion\_url | 初始化版本 | varchar | 256 | 用户录入 | 生成合同时产生，存储pdf文件地址 |
| stamp\_version\_url | 本系统盖章版本 | varchar | 256 |  | 本系统加盖电子签章产生，存储盖章后的pdf文件地址 |
| thirdstamp\_version\_url | 第三方系统盖章版本 | varchar | 256 | 系统赋值 | 本系统以及客户加盖电子签章产生，存储盖章后的pdf文件地址 |
| third\_file\_id | 第三方系统的文件id | varchar | 32 | 系统赋值 | 第三方系统中存储文件id |
| third\_file\_downloadurl; | 第三方系统的文件url | varchar | 256 | 系统赋值 | 第三方系统中存储文件id |
| third\_sign\_status | 第三方系统的签署状态 | varchar | 32 |  | 待签署、拒签、签署中、签署完成 |
| third\_sign\_time | 第三方系统的回调时间 | datetime | 16 |  | 第三方回调本系统接口的时间 |
| third\_flow\_id | 第三方系统的流程id | varchar | 32 |  | 第三方系统中存储流程id |

##### 三方平台调用日志表【预留】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表名** | **tb\_third\_api\_log** | **表中文名** | **三方平台调用日志表** | | |
| **字段名** | **字段描述** | **数据类型** | **字段长度** | **输入类型** | **备注** |
| pkid | 主键 | varchar | 32 | 系统赋值 |  |
| api\_platform\_type | 第三方平台类型 | varchar | 32 | 用户录入 | e签宝、短信平台 |
| api\_method | 请求方法 | varchar | 32 | 用户录入 | get、post、put、delete |
| api\_param\_format | 请求格式 | varchar | 32 | 用户录入 | xml、json |
| api\_request\_param | 请求入参 | text |  | 用户录入 |  |
| api\_respon\_param | 返回参数 | text |  | 系统赋值(可变更) |  |
| api\_url | 调用url路径 | varchar | 256 | 系统赋值 |  |

#### 公共工具类 ESignUtils.java

#### 对外怎么调用接口

ESignUtils.getSignContractInfo(String signFlowId);

EsignNotifyController e签宝平台回调本系统接口

#### 对内接口的实现逻辑

例如：获取合同解约链接方法ESignUtils.signflowRescissionUrl（String signFlowId）

//拼接api接口路径

String apiaddr= "/v3/sign-flow/"+ signFlowId + "/rescission-url";

//组装请求参数json格式的body部分

String jsonParm = "{}";

//配置请求方法

EsignRequestType requestType = EsignRequestType.POST;

//生成签名鉴权方式的的header

Map<String,String>header = EsignHttpHelper.signAndBuildSignAndJsonHeader(eSignAppId,eSignAppSecret,jsonParm,requestType.name(),apiaddr,true);

//发起接口请求

return EsignHttpHelper.doCommHttp(eSignHost, apiaddr,requestType , jsonParm, header,true);

#### 第一步 上传合同pdf附件到e签宝平台

|  |  |
| --- | --- |
| 调用说明 | 接口方法 |
| 上传合同pdf文件到e签宝平台 | //根据业务申请id查询出附件信息  HouseAppAttachEntity houseAppAttachEntity = houseAppAttachMapper.selectById(rentAppPkId);  String uFileId = ESignUtils.getESignFileId(houseAppAttachEntity.getFileUrl(),houseAppAttachEntity.getFileName());  //调用EsingUtils.java中的getESingFileId方法，将本系统待签署的合同附件上传到e签宝平台  public String getESignFileId(String fileUrlPath,String fileName) throws EsignException {  String rootFolder = new File(fileUrlPath).getAbsolutePath();  //获取文件id以及文件上传地址  String filePath = rootFolder + File.separator + fileName;  EsignHttpResponse getUploadUrl = getUploadUrl(filePath);  Gson gson1 = new Gson();  JsonObject getUploadUrlJsonObject = gson1.fromJson(getUploadUrl.getBody(), JsonObject.class);  String code = getUploadUrlJsonObject.get("code").getAsString();  String fileId = "";  if (successCode.equals(code)) {  JsonObject data = getUploadUrlJsonObject.getAsJsonObject("data");  //文件id后续发起签署使用  fileId = data.get("fileId").getAsString();  String fileUploadUrl = data.get("fileUploadUrl").getAsString();  System.out.println("获取文件id以及文件上传地址成功，文件id:"+fileId);  System.out.println("上传链接"+fileUploadUrl);  //文件上传  EsignHttpResponse uploadFileResponse = uploadFile(fileUploadUrl,filePath);  JsonObject uploadFileResponseJsonObject = gson1.fromJson(uploadFileResponse.getBody(), JsonObject.class);  String errCode = uploadFileResponseJsonObject.get("errCode").getAsString();  System.out.println("文件上传成功，状态码:"+errCode);  if (successCode.equals(errCode)) {  return fileId;  }  }  return fileId;  }  }  //将调用e签宝平台接口的过程写入第三方平台调用日志记录表  ThirdApiLogEntity thirdApiLogEntity = new ThirdApiLogEntity();  thirdApiLogEntity.setUfileUrl(uFileId);  thirdApiLogMapper.insert(thirdApiLogEntity); |

#### 第二步 发起签署流程

|  |  |
| --- | --- |
| 调用说明 | 接口方法 |
| 发起签署流程 | public String beginSignContractFlow(String fileId,String fileName,String contractName) throws EsignException {  Gson gson = new Gson();  /\* 发起签署\*/  EsignHttpResponse createByFile = createByFile(fileId,fileName,contractName);  //JSONObject getPsnAuthUrlObject = JSONObject.parseObject(getPsnAuthUrl.getBody());  JsonObject createByFileJsonObject = gson.fromJson(createByFile.getBody(), JsonObject.class);  JsonObject createByFileData = createByFileJsonObject.getAsJsonObject("data");  String signFlowId = "";  String code = createByFileData.get("code").getAsString();  if (successCode.equals(code)) {  signFlowId = createByFileData.get("signFlowId").getAsString();  }  System.err.println("流程id:"+signFlowId);  return signFlowId;  }  public EsignHttpResponse createByFile(String fileId,String fileName,String contractName) throws EsignException {  String apiaddr = "/v3/sign-flow/create-by-file";  //请求参数body体,json格式。get或者delete请求时jsonString传空json:"{}"或者null  String jsonParm = "{\n" +  " \"docs\": [\n" +  " {\n" +  " \"fileId\": \""+fileId+"\",\n" +  " \"fileName\": \""+fileName+"\"\n" +  " }\n" +  " ],\n" +  " \"signFlowConfig\": {\n" +  " \"signFlowTitle\": \""+contractName+"\",\n" +  " \"chargeConfig\": {\n" +  " \"chargeMode\": 0\n" +  " }\n" +  " },\n" +  " \"signers\": [\n" +  " {\n" +  " \"noticeConfig\": {\n" +  " \"noticeTypes\": \"1\"\n" +  " },\n" +  " \"orgSignerInfo\": {\n" +  " \"orgId\": \"\",\n" +  " \"orgInfo\": {\n" +  " \"legalRepIDCardNum\": \"\",\n" +  " \"legalRepIDCardType\": \"\",\n" +  " \"legalRepName\": \"孙xx\",\n" +  " \"orgIDCardNum\": \"915\*\*\*\*\*\*52355\",\n" +  " \"orgIDCardType\": \"CRED\_ORG\_USCC\"\n" +  " },\n" +  " \"orgName\": \"\*\*科技信息有限公司\",\n" +  " \"transactorInfo\": {\n" +  " \"psnAccount\": \"18\*\*\*\*\*\*\*\*79\",\n" +  " \"psnId\": \"\",\n" +  " \"psnInfo\": {\n" +  " \"bankCardNum\": \"\",\n" +  " \"psnIDCardNum\": \"\",\n" +  " \"psnIDCardType\": \"\",\n" +  " \"psnName\": \"\"\n" +  " }\n" +  " }\n" +  " },\n" +  " \"signConfig\": {\n" +  " \"signOrder\": 1\n" +  " },\n" +  " \"signFields\": [\n" +  " {\n" +  " \"customBizNum\": \"62453252523\",\n" +  " \"fileId\": \""+fileId+"\",\n" +  " \"normalSignFieldConfig\": {\n" +  " \"assignedSealId\": \"\",\n" +  " \"autoSign\": false,\n" +  " \"freeMode\": true,\n" +  " \"movableSignField\": true,\n" +  " \"orgSealBizTypes\": \"\",\n" +  " \"psnSealStyles\": \"\",\n" +  " \"signFieldPosition\": {\n" +  " \"positionPage\": \"1\",\n" +  " \"positionX\": 120,\n" +  " \"positionY\": 460\n" +  " },\n" +  " \"signFieldStyle\": 1\n" +  " },\n" +  " \"signFieldType\": 0\n" +  " }\n" +  " ],\n" +  " \"signerType\": 1\n" +  " }\n" +  " ]\n" +  "}";  //请求方法  EsignRequestType requestType = EsignRequestType.POST;  //生成请求签名鉴权方式的Header  Map<String, String> header = EsignHttpHelper.signAndBuildSignAndJsonHeader(eSignAppId, eSignAppSecret, jsonParm, requestType.name(), apiaddr, true);  //发起接口请求  return EsignHttpHelper.doCommHttp(eSignHost, apiaddr, requestType, jsonParm, header, true);  } |

#### 第三步 根据流程id获取签署链接地址

|  |  |
| --- | --- |
| 调用说明 | 接口方法 |
| 根据流程id获取签署链接地址 | public String getSignContractUrl(String signFlowId,String linkInfo,String companyName) throws EsignException {  Gson gson = new Gson();  //获取合同文件签署链接  EsignHttpResponse signUrl = signUrl(signFlowId,linkInfo,companyName);  JsonObject signUrlJsonObject = gson.fromJson(signUrl.getBody(), JsonObject.class);  JsonObject signUrlData = signUrlJsonObject.getAsJsonObject("data");  String shortUrl = signUrlData.get("shortUrl").getAsString();  String url = signUrlData.get("url").getAsString();  System.out.println("签署长链接:" +shortUrl);  System.out.println("签署短链接:"+url);  return url;  } |

#### 第四步 用户点击链接进行合同文件签署,暂无代码

#### 第五步 处理e签宝平台流程通知回调，流程结束后将流程中签署的文件信息下载到本系统中进行保存

|  |  |
| --- | --- |
| 调用说明 | 接口方法 |
| e签宝平台回调本系统接口，通知签署流程状态变更以及签署结果 | //接收e签宝平台接口回调  @PostMapping("/noticeFlowResult")  public ResponseMsg noticeFlowResult(HttpServletRequest request) {  String action = request.getHeader("action");  String signResult = request.getHeader("signResult");  if (verifySignSucc(request,eSignAppSecret)) {  //更新流程表中状态为已读  if (EsignActionEnum.OPERATOR\_READ.equals(action)) {  }  //更新流程表中状态为签署中  if (EsignActionEnum.SIGN\_FLOW\_INITIATED.equals(action)) {  }  //更新流程表中状态为已完成或拒签  if (EsignActionEnum.SIGN\_FLOW\_COMPLETE.equals(action)) {  if ("2".equals(signResult)) {  //更新流程表中状态为已完成  } else {  //更新流程表中状态为拒签  }  }  //更新流程表中状态为已完成  if (EsignActionEnum.SIGN\_MISSON\_COMPLETE.equals(action)) {  }  return ResponseMsg.succMsg();  } else {  return ResponseMsg.errorMsg("验证签名失败");  }  //下载已签署文件及附属材料  public String getSignContractInfo(String signFlowId) throws EsignException {  Gson gson = new Gson();  EsignHttpResponse fileDownloadUrl = fileDownloadUrl(signFlowId);  JsonObject fileDownloadUrlJsonObject = gson.fromJson(fileDownloadUrl.getBody(),JsonObject.class);  String code = fileDownloadUrlJsonObject.get("code").getAsString();  String downloadUrl = "";  if (successCode.equals(code)) {  JsonObject fileDownloadUrlArray = fileDownloadUrlJsonObject.getAsJsonObject("data");  JsonObject fileArray = fileDownloadUrlArray.getAsJsonObject("files");  downloadUrl = fileArray.toString();  System.out.println(fileDownloadUrlArray);  }  return downloadUrl;  } |

### E签宝平台调用的接口梳理

#### 上传本地文件

#### 基于文件发起签署

#### 获取签署页面链接

#### 下载已签署文件及附属材料