**契约锁电子签章平台接入方案**

# 1引言

## 编写说明

本文档根据契约锁对外接口总体规划编制。本文档提供了详细的接口调用说明，用于帮助对接方开发人员对接契约锁系统。

## 背景

本文档适用于契约锁电子签章平台与接入平台间的行为、数据和事件的交互与传递。

# 接口调用流程

## 通用签署流程

E:\桌面\开放平台\V2记录\通用签署流程.png

* 创建合同草稿  
  初始化合同基本信息，指定合同签署方。
* 添加合同文档  
  直接上传文件生成合同文档，并绑定到合同。  
  如果已配置合同模板，传入模板参数生成合同文档，并绑定合同。
* 发起合同  
  发起合同，同时支持指定签署位置（固定用户的签名位置）。
* 接收方签署合同，有如下几种签署方式  
  （1）对接方获取签署页面，用户在签署页面上认证并签署。  
  （2）用户接收到签署通知消息，打开消息中的签署链接，登录契约锁签署合同。  
  （3）对接方平台集成Js sdk，用户在集成的页面中签署，适用于不能集成第三方页面的APP（小程序、钉钉等）。
* 对接方签署合同，有如下几种签署方式  
  （1）调用接口直接签署。  
  （2）登录契约锁系统签署。  
  （3）获取签署页面，在签署页面签署。



## 使用业务分类签署流程

E:\桌面\开放平台\V2记录\使用业务分类签署流程.png

* 创建合同草稿  
  对接方事先在业务分类中指定签署方、签署要求、签署位置、印章、模板等信息，创建合同草稿时只需传入基本的合同信息及签署信息即可发起合同。  
  接收方签署合同，有如下几种签署方式  
  （1）对接方获取签署页面，用户在签署页面上认证并签署。  
  （2）用户接收到签署通知消息，打开消息中的签署链接，登录契约锁签署合同。  
  （3）对接方平台集成Js sdk，用户在集成的页面中签署，适用于不能集成第三方页面的APP（小程序、钉钉等）。
* 对接方签署合同，有如下几种签署方式  
  （1）调用接口直接签署。  
  （2）登录契约锁系统签署。  
  （3）获取签署页面，在签署页面签署。



# 接口对接注意（重要！！！）

* 契约锁测试环境与生产环境数据不同步。
* 接口服务器域名：  
  测试环境：<https://openapi.qiyuesuo.cn> 。  
  生产环境：<https://openapi.qiyuesuo.com> 。
* 接口示例中的accessToken和acessKey 均指开通开放平台后获取的 App Token。
* sdk接口示例中的accessSecret是开通开放平台后获取的App Secret。
* http接口示例中的Timestamp指当前时间戳，13位数字，如1566489571857。
* http接口示例中的Signature是32位小写md5值，生成方式如下：  
  MD5(accessToken + accessSecret + timestamp)。

# 接口清单

## 合同接口

## 创建合同

### 创建合同草稿

**描述：**根据业务分类配置创建合同草稿，业务分类在契约锁公有云中配置。

**业务分类配置介绍：**

**签署方：**

（1）签署方预设时，用户传入的签署方与业务分类配置的签署方必须完全一致（数量、类型、顺序均匹配），此时会使用业务分类配置的签署流程、签署位置、印章等（发起方除外，发起方的签署流程下面介绍）。

（2）签署方非预设时，以参数中传入的签署方为准（发起方除外）。

**发起方签署流程：**

（1）发起方流程预设时，发起方使用预设的签署流程、签署位置、印章等。

（2）发起方流程非预设时，以参数中传入的签署方为准。如果此时发起方签署公章节点未设置公章或签署人，将会使用业务分类中配置的公章和签署人。

**文件模板：**

发起合同时，必须保证合文件模板中发起方的必填参数均已填写完成。

**请求地址：**/v2/contract/draft

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

Contract（合同）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| id | String | 否 | 合同ID（接口返回值） |
| bizId | String | 否 | 业务ID； 一个合同对应一个bizId，不能重复 |
| subject | String | 是 | 合同主题（合同名称） |
| description | String | 否 | 合同描述 |
| sn | String | 否 | 合同编号；可由用户传入， 也可由契约锁自动生成 |
| expireTime | String | 否 | 合同过期时间； 格式为yyyy-MM-dd HH:mm:ss， 默认过期时间为业务分类中配置的时间 |
| ordinal | Boolean | 否 | 是否顺序签署；默认为true |
| send | Boolean | 否 | 是否发起合同； 发起合同后不能再进行添加文档、 添加签署方等操作 |
| category | Category | 否 | 业务分类；默认为“默认业务分类” |
| creator | User | 否 | 创建人；默认为虚拟用户， 创建人必须已经加入对接方的公司 |
| status | String | 否 | 合同状态（接口返回值）：DRAFT（草稿） RECALLED（已撤回），SIGNING（签署中），REJECTED（已退回），COMPLETE（已完成），EXPIRED（已过期），FILLING（拟定中），INVALIDING（作废中），INVALIDED（已作废） |
| signatories | List<Signatory> | 否 | 签署方；签署合同的公司/个人； 发起合同时必传； |
| templateParams | List<TemplateParam> | 否 | 模板参数，用于文件模板的填参 |
| documents | List<Document> | 否 | 合同文档（接口返回值） |

Category（业务分类）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| id | String | 否 | 业务分类ID |
| name | String | 否 | 业务分类名称； 如果id为空时，使用name来确定业务分类， 需要保证name对应的业务分类唯一 |

User（个人用户）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系类型：MOBILE（手机号），EMAIL（邮箱），EMPLOYEEID（员工ID），NUMBER（员工编号） |

Signatory（签署方）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| id | String | 否 | 签署方ID（接口返回值） |
| tenantType | String | 是 | 签署方类型：COMPANY（公司）， PERSONAL（个人） |
| status | String | 否 | 签署方状态（接口返回值）：DRAFT（草稿） RECALLED（已撤回），SIGNING（签署中），REJECTED（已退回），SIGNED（已完成），EXPIRED（已过期），FILLING（拟定中），WAITING（待签署），INVALIDING（作废中），INVALIDED（已作废） |
| tenantName | String | 是 | 签署方名称 |
| receiver | User | 是 | 接收人 |
| serialNo | Integer | 是 | 签署顺序 |
| actions | List<Action> | 否 | 签署动作（签署流程） |
| attachments | List<Attchment> | 否 | 附件要求；用于指定用户签署时上传附件 |
| userAuthInfo | UserAuthInfo | 否 | 指定签署方认证信息 |

Action（签署动作/签署流程）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| id | String | 否 | 签署节点ID（接口返回值） |
| type | String | 是 | 类型：COMPANY（签署公章），OPERATOR（经办人签署）  PERSONAL（个人签名），LP（法人签署），AUDIT（审批） |
| Status | String | 是 | 动作状态（接口返回值）：INIT（初始化）  START（已启动），FINISH（已完成），STOP（意外终止） |
| name | String | 否 | 名称 |
| serialNo | Integer | 是 | 执行顺序，从1开始 |
| sealId | String | 否 | 印章ID；用于指定签署时所有印章 |
| operators | List<User> | 否 | 操作人；用于指定公章签署人，法人签署和经办人签署时无需指定 |

Attchment（附件要求）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| id | String | 否 | 附件ID（接口返回值） |
| title | String | 是 | 名称 |
| required | Boolean | 否 | 是否必须；默认为false |
| needSign | Boolean | 否 | 是否需要签署；默认为false |

UserAuthInfo（签署方认证信息）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| idCardNo | String | 否 | 文档参数名称 |
| bankNo | String | 否 | 指定认证银行卡号，仅银行卡认证时生效 |
| bankMobile | String | 否 | 指定认证银行卡手机号，仅银行卡认证时生效 |

TemplateParam（合同文档参数）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| name | String | 文档参数名称 |
| value | String | 文档参数值 |
| required | Boolean | 是否为必填参数（接口返回值） |

Document（合同文档）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 附件ID |
| title | String | 名称 |
| pageCount | Integer | 文档页数 |
| createTime | String | 创建时间；格式为yyyy-MM-dd HH:mm:ss |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Object | 返回数据；格式与请求参数一致 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1102 | DUPLICATE SIGNATORY，签署方重复 |
| 1103 | ACTION REQUIRED，缺少签署节点；签署方为公司时，必须保证其下的签署流程 不为空，签署流程可以由参数传入，也可以使用业务分类中配置的签署流程。 |
| 1105 | ACTION OPERATORS REQUIRED，缺少操作人； 签署方为公司时，其下的“审批”、“签署”节点必须设置操作人。 |
| 1111 | OPERATOR REQUIRED，缺少经办人； 存在经办人节点时，必须设置经办人。 |
| 1112 | DUPLICATE BIZID，重复的bizId； |
| 1113 | SIGNATORY REQUIRED，缺少签署方 |
| 1201 | SEAL NOT FOUND，找不到印章 |
| 1203 | INVALID SEAL STATUS，印章不是正常状态 |
| 1301 | CATEGORY CONFIG NOT MATCH，业务分类配置的签署方数量/类型 与参数中的签署方数量/类型不匹配 |
| 1302 | CATEGORY CONFIG ERROR，业务分类配置数据错误； 可能原因有：业务分类中配置的操作人已离职，或其他错误。 |
| 1401 | DOCUMENT REQUIRED，缺少合同文档；发起合同时必须有合同文档 |

**请求示例：**

**http示例**

​ POST /v2/contract/draft HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"sn": "123321",

"subject": "V2合同",

"description": "合同描述",

"expireTime": "2019-08-25 00:00:00",

"ordinal": true,

"send": false,

"category": {

"name": "V2"

},

"creator": {

"contact": "18435186216",

"contactType": "MOBILE"

},

"signatories": [

{

"tenantType": "COMPANY",

"tenantName": "思晨教育",

"receiver": {

"contact": "10100000000",

"contactType": "MOBILE"

},

"serialNo": 1,

"actions": [

{

"type": "COMPANY",

"name": "公章签署动作",

"serialNo": 1,

"operators": [

{

"contact": "10100000000",

"contactType": "MOBILE"

}

]

},

{

"type": "AUDIT",

"name": "审批动作",

"serialNo": 3,

"operators": [

{

"contact": "10100000000",

"contactType": "MOBILE"

}

]

},

{

"type": "LP",

"name": "法人签署动作",

"serialNo": 4

}

]

},

{

"tenantType": "PERSONAL",

"tenantName": "吉哲",

"receiver": {

"contact": "10100000000",

"contactType": "MOBILE"

},

"serialNo": 2

}

]

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 合同基本参数

Contract contract = new Contract();

contract.setSubject("测试合同");

contract.setCategory(new Category("个人-平台-预设签署方-默认流程"));

contract.setBizId("");

contract.setSend(false);

// 个人

Signatory signatory1 = new Signatory();

signatory1.setTenantName("个人接收方姓名");

signatory1.setTenantType("PERSONAL");

signatory1.setReceiver(new User("个人接收方姓名", "10100000000", "MOBILE"));

signatory1.setSerialNo(1);

// 对接方

Signatory signatory2 = new Signatory();

signatory2.setTenantName("测试合同");

signatory2.setTenantType("COMPANY");

signatory2.setReceiver(new User("公司接收方名称", "10100000000", "MOBILE"));

signatory2.setSerialNo(2);

Action action = new Action("COMPANY", 1);

signatory2.addAction(action);

// 设置签署方

contract.addSignatory(signatory1);

contract.addSignatory(signatory2);

contract.addTemplateParam(new TemplateParam("param1", "value1"));

contract.addTemplateParam(new TemplateParam("param2", "value2"));

// 创建合同

ContractDraftRequest request = new ContractDraftRequest(contract);

String response = sdkClient.service(request);

SdkResponse<Contract> responseObj = JSONUtils.toQysResponse(response, Contract.class);

// 返回结果

if(responseObj.getCode() == 0) {

Contract result = responseObj.getResult();

logger.info("创建合同成功，合同ID:{}", result.getId());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**​​C#示例**

//初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

ContractDraftRequest request = new ContractDraftRequest();

Contract contract = new Contract();

contract.Subject = "测试合同";

//添加对接方签署方

Signatory platformSignatory = new Signatory("COMPANY", new User("对接方经办人姓名", "101000000000", "MOBILE"), 1);

platformSignatory.TenantName = "测试公司";//对接方公司名称

//添加对接方签署流程，可根据需要调整

//目前对接方签署流程为：（1）员工审批 （2）法人章签署（3）公章签署

// 审批流程，并设置审批操作人

SignAction aduitAction = new SignAction("AUDIT", 1);

aduitAction.AddOperators(new User("101000000000", "MOBILE"));

platformSignatory.AddAction(aduitAction);

// 公章签署流程，并设置签署公章ID

SignAction sealAction = new SignAction("COMPANY", 2);

sealAction.SealId = "2490828768980361630";

platformSignatory.AddAction(sealAction);

// 法人章签署流程

platformSignatory.AddAction(new SignAction("LP", 3));

contract.AddSignatory(platformSignatory);

//添加个人签署方，并设置个人签署方需要上传的附件内容

Signatory personalSignatory = new Signatory("PERSONAL", new User("个人接收方姓名", "101000000000", "MOBILE"), 2);

personalSignatory.TenantName = "个人接收方姓名";//接收方名称

personalSignatory.AddAttachment(new Attachment("附件", true, false));//添加上传附件要求，并设置为必须上传

contract.AddSignatory(personalSignatory);

//设置合同基本信息

contract.ExpireTime = "2019-07-28 23:59:59";//设置合同过期时间，合同过期时间需要晚于发起时间

contract.Send = false; //合同不发起

request.Contract = contract;

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception("创建合同草稿请求服务器失败,失败原因" + e.Message);

}

SdkResponse<Contract> sdkResponse = HttpJsonConvert.DeserializeResponse<Contract>(response);

if (!sdkResponse.Code.Equals(0))

{

throw new Exception("创建合同草稿失败，失败原因：" + sdkResponse.Message);

}

Console.WriteLine("合同草稿创建成功");

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

/\*\* 合同基本信息 \*\*/

$contract = new Contract();

// $contract->setId("");

$contract->setBizId("1111111");

$contract->setSubject("合同名称");

$contract->setDescription("合同描述");

$contract->setSn("123321");

$contract->setExpireTime("2019-07-25 00:00:00");

$contract->setOrdinal(true);

$contract->setSend(false);

/\*\*指定业务分类\*\*/

$category = new Category();

$category->setName("业务分类名称");

$contract->setCategory($category);

$creator = new User();

$creator->setContact("10100000000");

$creator->setContactType("MOBILE");

$contract->setCreator($creator);

/\*\*公司签署方\*\*/

$companySignatory = new Signatory();

$companySignatory->setTenantType("COMPANY");

$companySignatory->setTenantName("测试企业");

$companySignatory->setSerialNo(1);

/\*\*公司签署方接收人\*\*/

$receiver = new User();

$receiver->setContact("10100000000");

$receiver->setContactType("MOBILE");

$companySignatory->setReceiver($receiver);

/\*\*公章签署动作\*\*/

$companyAction = new Action();

$companyAction->setType("COMPANY");

$companyAction->setName("公章签署动作");

$companyAction->setSerialNo(1);

/\*\*公章签署操作人\*\*/

$companyOperator = new User();

$companyOperator->setContact("10100000000");

$companyOperator->setContactType("MOBILE");

$companyOperators = array();

array\_push($companyOperators, $companyOperator);

$companyAction->setOperators($companyOperators);

/\*\*经办人签字动作\*\*/

$operatorAction = new Action();

$operatorAction->setType("OPERATOR");

$operatorAction->setName("经办人签字动作");

$operatorAction->setSerialNo(2);

/\*\*经办人签字操作人\*\*/

$operatorOperator = new User();

$operatorOperator->setContact("10100000000");

$operatorOperator->setContactType("MOBILE");

$operatorOperators = array();

array\_push($operatorOperators, $operatorOperator);

$operatorAction->setOperators($operatorOperators);

/\*\*审批动作\*\*/

$auditAction = new Action();

$auditAction->setType("AUDIT");

$auditAction->setName("审批动作");

$auditAction->setSerialNo(3);

/\*\*审批操作人\*\*/

$auditOperator = new User();

$auditOperator->setContact("10100000000");

$auditOperator->setContactType("MOBILE");

$auditOperators = array();

array\_push($auditOperators, $auditOperator);

$auditAction->setOperators($auditOperators);

/\*\*法人签署动作\*\*/

$lpAction = new Action();

$lpAction->setType("LP");

$lpAction->setName("法人签署动作");

$lpAction->setSerialNo(4);

$actions = array();

/\*\*公司签署方设置公章签署动作\*\*/

array\_push($actions, $companyAction);

/\*\*公司签署方设置经办人签字动作\*\*/

array\_push($actions, $operatorAction);

/\*\*公司签署方设置审批动作\*\*/

array\_push($actions, $auditAction);

/\*\*公司签署方设置审批动作\*\*/

array\_push($actions, $lpAction);

$companySignatory->setActions($actions);

/\*\*个人签署方\*\*/

$personalSignatory = new Signatory();

$personalSignatory->setTenantType("PERSONAL");

$personalSignatory->setTenantName("个人签署方名称");

$personalSignatory->setSerialNo(2);

$receiver = new User();

$receiver->setContact("10100000000");

$receiver->setContactType("MOBILE");

$personalSignatory->setReceiver($receiver);

/\*\*所有签署方\*\*/

$signatories = array();

array\_push($signatories, $companySignatory);

array\_push($signatories, $personalSignatory);

/\*\*填写模板参数\*\*/

$templateParam1 = new TemplateParam();

$templateParam1->setName("参数1");

$templateParam1->setValue("值1");

$templateParams = array();

array\_push($templateParams, $templateParam1);

/\*\*合同文档\*\*/

$document1 = new Document();

$document1->setId("2589678690921123947");

$documents = array();

array\_push($documents, $document1);

/\*\*合同签署方、文档、文档参数\*\*/

$contract->setSignatories($signatories);

$contract->setTemplateParams($templateParams);

$contract->setDocuments($documents);

$baseRequest = new ContractDraftRequest($contract);

$result = $sdkClient->service($baseRequest);

print\_r($result);

**Python示例**

# 初始化sdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

draft\_contract = Contract()

draft\_contract.set\_subject("合同主题-人事合同")

# 设置签署方 - 对接方公司，tenantName更换为您公司的名称，接收方更正为您公司员工

company\_signatory = Signatory()

company\_signatory.set\_tenantType("COMPANY")

company\_signatory.set\_tenantName("测试公司")

company\_signatory.set\_serialNo(1)

company\_receiver = User()

company\_receiver.set\_contact("10100000000")

company\_receiver.set\_contactType("MOBILE")

company\_signatory.set\_receiver(company\_receiver)

# 设置对接方公司签署流程 - 审批流程，并设置审批人

audit\_action = Action()

audit\_action.set\_type("AUDIT")

audit\_action.set\_serialNo(1)

audit\_operator = User()

audit\_operator.set\_contact("10100000000")

audit\_operator.set\_contactType("MOBILE")

audit\_action.set\_operators([audit\_operator])

# 设置对接方公司签署流程 - 公章签署流程，并设置该流程应该签署的公章

seal\_action = Action()

seal\_action.set\_type("COMPANY")

seal\_action.set\_serialNo(2)

seal\_action.set\_sealId("2490828768980361630")

# 设置对接方公司签署流程 - 法人章签署流程

lp\_action = Action()

lp\_action.set\_type("LP")

lp\_action.set\_serialNo(3)

company\_signatory.set\_actions([audit\_action, seal\_action, lp\_action])

# 设置签署方 - 个人签署方，并设置附件上传要求

personal\_signatory = Signatory()

personal\_signatory.set\_tenantType("PERSONAL")

personal\_signatory.set\_serialNo(2)

personal\_signatory.set\_tenantName("个人接收方姓名")

personal\_receiver = User()

personal\_receiver.set\_contact("10100000000")

personal\_receiver.set\_contactType("MOBILE")

personal\_signatory.set\_receiver(personal\_receiver)

# 生成上传附件要求

personal\_attachment = Attachment()

personal\_attachment.set\_title("附件-身份证正面照")

personal\_attachment.set\_required(True)

personal\_signatory.set\_attachments([personal\_attachment])

draft\_contract.set\_signatories([company\_signatory, personal\_signatory])

# 设置合同过期时间

draft\_contract.set\_expireTime("2019-07-28 23:59:59")

# 不发起合同

draft\_contract.set\_send(False)

# 请求服务器

draft\_response = sdkClient.request(ContractDraftRequest(draft\_contract))

# 解析返回数据

draft\_mapper = json.loads(draft\_response)

if draft\_mapper['code'] != 0:

raise Exception("创建合同草稿失败，失败原因：", draft\_mapper['message'])

draft\_result = draft\_mapper['result']

draft\_contractid = draft\_result['id']

print('创建合同草稿成功，合同ID：', draft\_contractid)

### 添加合同文档

#### 用文件添加合同文档

**请求地址：**/v2/document/addbyfile

**请求方法：**POST

**请求格式：** multipart/form-data;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| title | String | 是 | 名称 |
| file | File | 是 | 合同文件 |
| fileSuffix | String | 是 | 文件类型（文件后缀）： doc，docx，pdf，jpeg，png，gif，tiff，html，htm |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Response | 返回数据 |

Response（返回数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| documentId | String | 合同文档ID |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1101 | INVALID CONTRACT STATUS，合同状态无效； 只能为“草稿”状态合同添加文档。 |
| 1403 | INVALID FILE，无效的合同文件； 合同文件转换失败，可能是由于文件损坏或者文件类型错误。 |

**请求示例：**

**http示例**

​ POST /v2/document/addbyfile HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="contractId"

2591540368898105360

------WebKitFormBoundary7MA4YWxkTrZu0gW--

Content-Disposition: form-data; name="title"

文件文档

------WebKitFormBoundary7MA4YWxkTrZu0gW--

Content-Disposition: form-data; name="file"; filename="/C:/Users/QYS/Desktop/Test/授权.pdf

------WebKitFormBoundary7MA4YWxkTrZu0gW--

Content-Disposition: form-data; name="fileSuffix"

pdf

------WebKitFormBoundary7MA4YWxkTrZu0gW—

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 添加合同文档

DocumentAddByFileRequest request = new DocumentAddByFileRequest(contractId,

new StreamFile(new FileInputStream("E:/test/NoSign.pdf")), "pdf", "文件一");

String response = sdkClient.service(request);

SdkResponse<DocumentAddResult> responseObj = JSONUtils.toQysResponse(response, DocumentAddResult.class);

if(responseObj.getCode() == 0) {

DocumentAddResult result = responseObj.getResult();

logger.info("添加合同文档成功，文档ID:{}", result.getDocumentId());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

//初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(serverUrl, accessKey, accessSecret);

string contractId = "2589012016299597907";

// 引入待添加文件

string path = "C:\Users\Richard Cheung\Documents\契约锁\测试\人事合同.pdf";

Stream file = new FileStream(path, FileMode.Open);

// 组装根据文件添加文档请求

DocumentAddByFileRequest request = new DocumentAddByFileRequest(contractId, "添加文件", file, "pdf");

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回结果

SdkResponse<DocumentAddResult> responseObject = HttpJsonConvert.DeserializeResponse<DocumentAddResult>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$documentAddByFileRequest = new DocumentAddByFileRequest();

$documentAddByFileRequest->setContractId('2590758386643734529');

$documentAddByFileRequest->setBizId("1111111");

$file\_path = "C:/Users/QYS/Desktop/Test/444.pdf";

$file\_path = iconv("UTF-8", "GBK", realpath($file\_path));

$file = new \CURLFile($file\_path);

$documentAddByFileRequest->setFile($file);

$documentAddByFileRequest->setFileSuffix('pdf');

$documentAddByFileRequest->setTitle('V2添加文档');

$result = $sdkClient->service($documentAddByFileRequest);

print\_r($result);

return $result;

**Python示例**

#初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

documentbyfile\_request = DocumentAddByFileRequest()

file = open("C:\Users\Richard Cheung\Documents\契约锁\测试\测试合同.pdf", "rb")

documentbyfile\_request.set\_file(file)

documentbyfile\_request.set\_contractId(draft\_contractid)

# 将fileSuffix替换为将上传文件正确的文件类型

documentbyfile\_request.set\_fileSuffix('pdf')

documentbyfile\_request.set\_title('本地文件上传文档')

# 请求服务器

documentbyfile\_response = sdkClient.request(documentbyfile\_request)

# 解析返回数据

documentbyfile\_mapper = json.loads(documentbyfile\_response)

if documentbyfile\_mapper['code'] != 0:

raise Exception('根据本地文件添加合同文档失败，失败原因：', documentbyfile\_mapper['message'])

documentbyfile\_result = documentbyfile\_mapper['result']

file\_documentId = documentbyfile\_result['documentId']

print('根据本地文件添加合同文档成功，文档ID：', file\_documentId)

#### 用模板添加合同文档

**请求地址：**/v2/document/addbytemplate

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| title | String | 是 | 名称 |
| templateId | String | 是 | 模板ID |
| templateParams | List<TemplateParam> | 否 | 模板参数；如果是参数模板，则必填 |

TemplateParam（模板参数）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| name | String | 是 | 参数名称 |
| value | String | 是 | 参数值 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Response | 返回数据 |

Response（返回数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| documentId | String | 合同文档ID |

**请求示例：**

**http示例**

​ POST /v2/document/addbytemplate HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2593815518218461185",

"bizId": null,

"title": "V2添加模板文档",

"templateId": "2560778409594396776",

"templateParams": [

{

"name": "乙方姓名",

"value": "张三"

}

]

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 添加合同文档

List<TemplateParam> params = new ArrayList<>();

params.add(new TemplateParam("param1", "val1"));

params.add(new TemplateParam("param2", "val2"));

DocumentAddByTemplateRequest request = new DocumentAddByTemplateRequest(contractId,

templateId, params, "文件二");

String response = sdkClient.service(request);

SdkResponse<DocumentAddResult> responseObj = JSONUtils.toQysResponse(response, DocumentAddResult.class);

if(responseObj.getCode() == 0) {

DocumentAddResult result = responseObj.getResult();

logger.info("添加合同文档成功，文档ID:{}", result.getDocumentId());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**​C#示例**

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

string contractId = "2589012016299597907";

string templateId = "2562841550577214123";

// 组装根据模板创建文档请求

DocumentAddByTemplateRequest request = new DocumentAddByTemplateRequest(contractId, "添加模板", templateId);

// 设置模板参数内容

request.AddTemplateParam(new TemplateParam("Sender", "契约锁"));

request.AddTemplateParam(new TemplateParam("Reciver1", "开放平台"));

request.AddTemplateParam(new TemplateParam("Reciver2", "对接方公司"));

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<DocumentAddResult> responseObject = HttpJsonConvert.DeserializeResponse<DocumentAddResult>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$documentAddByTemplateRequest = new DocumentAddByTemplateRequest();

$documentAddByTemplateRequest->setContractId('2590758386643734529');

$documentAddByTemplateRequest->setBizId("1111111");

$documentAddByTemplateRequest->setTitle('V2添加模板文档');

$documentAddByTemplateRequest->setTemplateId('2558597440364655396');

/\*\*填写模板参数\*\*/

$templateParam1 = new TemplateParam();

$templateParam1->setName("乙方姓名");

$templateParam1->setValue("张三");

$templateParams = array();

array\_push($templateParams, $templateParam1);

$documentAddByTemplateRequest->setTemplateParams($templateParams);

$result = $sdkClient->service($documentAddByTemplateRequest);

print\_r($result);

return $result;

**Python示例**

#初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

documentbytemplate\_request = DocumentAddByTemplateRequest()

documentbytemplate\_request.set\_title('模板上传文档')

documentbytemplate\_request.set\_contractId(draft\_contractid)

documentbytemplate\_request.set\_templateId('2492236993899110515')

# 若模板为参数模板，设置模板中的参数值

documentbytemplate\_request.set\_templateParams(

[TemplateParam('接收方1', '契约锁'), TemplateParam('接收方2', '电子合同')])

# 请求服务器

documentbytemplate\_response = sdkClient.request(documentbytemplate\_request)

# 解析返回数据

documentbytemplate\_mapper = json.loads(documentbytemplate\_response)

if documentbytemplate\_mapper['code'] != 0:

raise Exception('根据模板添加合同文档失败，失败原因：', documentbytemplate\_mapper['message'])

documentbytemplate\_result = documentbytemplate\_mapper['result']

template\_documentId = documentbytemplate\_result['documentId']

print('根据模板添加合同文档成功，文档ID：', template\_documentId)

### 发起合同

**描述：**调用此接口来发起“草稿”状态的合同。合同发起后，签署方将收到签署通知签署合同。支持在发起合同时指定签署位置。

**请求地址：**/v2/contract/send

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| stampers | List<Stamper> | 否 | 指定签署位置 |

Stamper（签署位置）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| actionId | String | 否 | 签署节点ID；公司的签署位置必须 |
| signatoryId | String | 否 | 签署方ID；个人的签署位置必传 |
| type | String | 是 | 签署类型：  COMPANY（公章），PERSONAL（个人签名），  LP（法人章），TIMESTAMP（时间戳），ACROSS\_PAGE（骑缝章） |
| documentId | String | 是 | 合同文档ID |
| sealId | String | 否 | 印章ID（暂未支持） |
| keyword | String | 否 | 关键字 |
| keywordIndex | Integer | 否 | 关键字索引：1代表第1个关键字，0代码所有关键字  ，-1代表倒数第1个关键字；默认为1 |
| page | Integer | 否 | 坐标页码，0代表所有 |
| offsetX | Decimal | 否 | 横坐标/关键字偏移量 |
| offsetY | Decimal | 否 | 纵坐标/关键字偏移量 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1101 | INVALID CONTRACT STATUS，合同状态无效； 只能发起“草稿”状态合同。 |
| 1108 | KEYWORD NOT FOUND，未找到关键字； 签署位置中关键字未在文档中找到。 |
| 1109 | SIGN PAGE BEYOND，签署页码超出文档页数； 签署位置中的页码超出文档页数。 |
| 1203 | INVALID SEAL STATUS，无效的印章状态 |

**请求示例：**

**Http示例**

​ POST /v2/contract/send HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": null,

"stampers": [{

"actionId": "2591541618892972178",

"type": "COMPANY",

"documentId": "2591542337725374704",

"sealId": "2566229349702860958",

"offsetX": "0.1",

"offsetY": "-0.1",

"keyword": "劳动",

"keywordIndex": "2"

}, {

"actionId": "2591541618892972178",

"type": "COMPANY",

"documentId": "2591542337725374704",

"sealId": "2566229349702860958",

"offsetX": "0.5",

"offsetY": "0.5",

"page": "1"

}, {

"signatoryId": "2591541619077521559",

"type": "PERSONAL",

"documentId": "2591542337725374704",

"offsetX": 0.9,

"offsetY": 0.9,

"page": 1

}]

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 发起时可以设置签署位置

Stamper stamper = new Stamper();

stamper.setActionId(2589310177048080947L);

stamper.setDocumentId(2589310172379820580L);

stamper.setType("COMPANY");

stamper.setPage(1);

stamper.setOffsetX(0.1);

stamper.setOffsetY(0.1);

Stamper stamper2 = new Stamper();

stamper2.setActionId(2589310177048080947L);

stamper2.setDocumentId(2589310172379820580L);

stamper2.setType("COMPANY");

stamper2.setPage(1);

stamper2.setOffsetX(0.2);

stamper2.setOffsetY(0.1);

List<Stamper> stampers = new ArrayList<>();

stampers.add(stamper);

stampers.add(stamper2);

// 发起合同

ContractSendRequest request = new ContractSendRequest(2589310172899914283L, stampers);

String response = sdkClient.service(request);

SdkResponse<Object> responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("合同发起成功");

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

​**C#示例**

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

ContractSendRequest request = new ContractSendRequest();

//指定签署位置

//发起方（公司）：印章签署位置、法人章签署位置、时间戳位置

Stamper sealStamper = new Stamper();

sealStamper.DocumentId = documentId1;

sealStamper.ActionId = companyActionId;

sealStamper.Type = "COMPANY";

// 绝对位置定位：页数、X、Y定位

sealStamper.Page = 1;

sealStamper.OffsetX = 0.1;

sealStamper.OffsetY = 0.2;

// 关键字位置定位：keyword,keywordIndex,X,Y

//sealStamper.Keyword = "甲方签字"; //确保文件中包含该关键字

//sealStamper.KeywordIndex = 1;

//sealStamper.OffsetX = 0.0;

//sealStamper.OffsetY = 0.0;

Stamper lpStamper = new Stamper();

lpStamper.DocumentId = documentId1;

lpStamper.ActionId = lpActionId;

lpStamper.Type = "LP";

lpStamper.Page = 1;

lpStamper.OffsetX = 0.4;

lpStamper.OffsetY = 0.6;

Stamper timeStamper = new Stamper();

timeStamper.DocumentId = documentId1;

timeStamper.ActionId = companyActionId;

timeStamper.Type = "TIMESTAMP";

timeStamper.Page = 1;

timeStamper.OffsetX = 0.3;

timeStamper.OffsetY = 0.5;

//个人签署方 签署位置

Stamper personalStamper = new Stamper();

personalStamper.DocumentId = documentId1;

personalStamper.SignatoryId = personalSignatoryId;

personalStamper.Type = "PERSONAL";

personalStamper.Page = 1;

personalStamper.OffsetX = 0.7;

personalStamper.OffsetY = 0.5;

request.AddStamper(sealStamper);

request.AddStamper(lpStamper);

request.AddStamper(timeStamper);

request.AddStamper(personalStamper);

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception("发起合同请求服务器失败，失败原因：" + e.Message);

}

SdkResponse<Object> sdkResponse = HttpJsonConvert.DeserializeResponse<Object>(response);

if (!sdkResponse.Code.Equals(0))

{

throw new Exception("发起合同失败，失败原因：" + sdkResponse.Message);

}

Console.WriteLine("合同发起成功");

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractSendRequest = new ContractSendRequest();

$contractSendRequest->setContractId('2590758386643734529');

$contractSendRequest->setBizId("1111111");

/\*\*个指定签署位置：关键字定位\*\*/

$stamper1 = new Stamper();

$stamper1->setType('COMPANY');

$stamper1->setActionId('2590758390800289796');

$stamper1->setDocumentId('2590764888637018192');

$stamper1->setSealId('2555244623418466517');

$stamper1->setKeyword('劳动');

$stamper1->setKeywordIndex('2');

$stamper1->setOffsetX('0.1');

$stamper1->setOffsetY('-0.1');

/\*\*个指定签署位置：坐标定位\*\*/

$stamper2 = new Stamper();

$stamper2->setType('COMPANY');

$stamper2->setActionId('2590758390800289796');

$stamper2->setDocumentId('2590762507098624044');

$stamper2->setSealId('2555244623418466517');

$stamper2->setPage('1');

$stamper2->setOffsetX('0.5');

$stamper2->setOffsetY('0.5');

$stampers = array();

array\_push($stampers, $stamper1);

array\_push($stampers, $stamper2);

$contractSendRequest->setStampers($stampers);

$result = $sdkClient->service($contractSendRequest);

print\_r($result);

return $result;

**Python示例**

#初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 设置签署位置 - 公章签署

seal\_stamper = Stamper()

seal\_stamper.set\_actionId(company\_actionId)

seal\_stamper.set\_documentId(file\_documentId)

seal\_stamper.set\_type('COMPANY')

seal\_stamper.set\_offsetX(0.3)

seal\_stamper.set\_offsetY(0.5)

seal\_stamper.set\_page(1)

# 设置签署位置 - 法人章签署

lp\_stamper = Stamper()

lp\_stamper.set\_actionId(lp\_actionId)

lp\_stamper.set\_documentId(file\_documentId)

lp\_stamper.set\_type('LP')

lp\_stamper.set\_offsetX(0.6)

lp\_stamper.set\_offsetY(0.4)

lp\_stamper.set\_page(1)

# 设置签署位置 - 时间戳签署

time\_stamper = Stamper()

time\_stamper.set\_actionId(company\_actionId)

time\_stamper.set\_documentId(file\_documentId)

time\_stamper.set\_type('TIMESTAMP')

time\_stamper.set\_offsetX(0.9)

time\_stamper.set\_offsetY(0.5)

time\_stamper.set\_page(1)

# 设置签署位置 - 个人接收方 - 个人签名

personal\_stamper = Stamper()

personal\_stamper.set\_signatoryId(personal\_signatoryId)

personal\_stamper.set\_documentId(file\_documentId)

personal\_stamper.set\_type('PERSONAL')

personal\_stamper.set\_offsetX(0.4)

personal\_stamper.set\_offsetY(0.7)

personal\_stamper.set\_page(1)

# 请求服务器

send\_response = sdkClient.request(

ContractSendRequest(draft\_contractid, [seal\_stamper, lp\_stamper, time\_stamper, personal\_stamper]))

# 解析返回数据

send\_mapper = json.loads(send\_response)

if send\_mapper['code'] != 0:

raise Exception('发起合同失败，失败原因：', send\_mapper['message'])

print('合同发起成功')

## 签署合同

#### 签署公章

**请求地址：**/v2/contract/companysign

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| sealId | String | 否 | 印章ID，若指定了签署位置，可直接在此传递印章ID，同时若下方签署位置参数中无印章ID，会从此处获取 |
| stampers | Stamper | 否 | 指定签署位置 |

Stamper（签署位置）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| type | String | 是 | 签署类型：COMPANY（公章），  TIMESTAMP（时间戳），ACROSS\_PAGE（骑缝章） |
| documentId | String | 是 | 合同文档ID |
| sealId | String | 否 | 印章ID |
| keyword | String | 否 | 关键字 |
| keywordIndex | Integer | 否 | 关键字索引：1代表第1个关键字，  0代表所有关键字，-1代表倒数第1个关键字；默认为1 |
| page | Integer | 否 | 坐标页码，0代表所有 |
| offsetX | Decimal | 否 | 横坐标/关键字偏移量 |
| offsetY | Decimal | 否 | 纵坐标/关键字偏移量 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1106 | INVALID\_STAMPER\_TYPE，无效的签章类型；签署公章支持的签章类型如下： COMPANY（公章），TIMESTAMP（时间戳），ACROSS\_PAGE（骑缝章）。 |
| 1107 | NOT SIGN STEP，未挨到签署； 未挨到公司签署公章或者没有签署公章节点。 |
| 1109 | SIGN PAGE BEYOND，签署页码超出文档页数； 签署位置中的页码超出文档页数。 |
| 1201 | SEAL NOT FOUND，找不到印章 |

**请求示例：**

**Http示例**

​ POST /v2/contract/companysign HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": null,

"stampers": [

{

"type": "COMPANY",

"documentId": "2591542337725374704",

"sealId": "2566229349702860958",

"offsetX": "0.2",

"offsetY": "-0.2",

"keyword": "劳动",

"keywordIndex": "2"

},

{

"type": "TIMESTAMP",

"documentId": "2591542337725374704",

"sealId": "2566229349702860958",

"offsetX": "0.5",

"offsetY": "0.5",

"page": "1"

},

{

"type": "ACROSS\_PAGE",

"documentId": "2591542337725374704",

"sealId": "2566229349702860958",

"offsetY": "0.5"

}

]

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 签署公章

SignParam param = new SignParam();

param.setContractId(2589394742435041330L);

ContractSignCompanyRequest request = new ContractSignCompanyRequest(param);

String response = sdkClient.service(request);

SdkResponse<Object> responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("公章签署成功");

} else {

logger.info("公章签署失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "http://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 组装签署请求

ContractSignCompanyRequest request = new ContractSignCompanyRequest();

request.ContractId = "2589012016299597907";

// 公章签署位置

Stamper stamper1 = new Stamper();

stamper1.Type = "COMPANY";

stamper1.DocumentId = "2589013696294510740";

stamper1.OffsetX = 0.0;

stamper1.OffsetY = 0.01;

stamper1.Keyword = "核对人";

stamper1.SealId = "2566547701649482059";

request.AddStampers(stamper1);

// 时间戳签署位置

Stamper stamper3 = new Stamper();

stamper3.Type = "TIMESTAMP";

stamper3.DocumentId = "2589013696294510740";

stamper3.OffsetX = 0.0;

stamper3.OffsetY = 0.01;

stamper3.Keyword = "核对单位盖章";

request.AddStampers(stamper3);

// 公章签署位置

Stamper stamper2 = new Stamper();

stamper2.Type = "COMPANY";

stamper2.DocumentId = "2589013696294510740";

stamper2.OffsetX = 0.2;

stamper2.OffsetY = 0.4;

stamper2.Page = 1;

stamper2.SealId = "2566547701649482059";

request.AddStampers(stamper2);

// 骑缝章签署位置

Stamper stamper4 = new Stamper();

stamper4.Type = "ACROSS\_PAGE";

stamper4.DocumentId = "2589013583094440075";

stamper4.OffsetY = 0.4;

stamper4.SealId = "2566547701649482059";

request.AddStampers(stamper4);

string response = null;

try

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractCompanySignRequest = new ContractCompanySignRequest();

$contractCompanySignRequest->setContractId('2590758386643734529');

$contractCompanySignRequest->setBizId("1111111");

/\*\*公章签署\*\*/

$stamper1 = new Stamper();

$stamper1->setType('COMPANY');

$stamper1->setDocumentId('2590764888637018192');

$stamper1->setSealId('2555244623418466517');

$stamper1->setKeyword('劳动');

$stamper1->setKeywordIndex('2');

$stamper1->setOffsetX('0.2');

$stamper1->setOffsetY('-0.2');

/\*\*时间戳签署\*\*/

$stamper2 = new Stamper();

$stamper2->setType('TIMESTAMP');

$stamper2->setDocumentId('2590764888637018192');

$stamper2->setPage('1');

$stamper2->setOffsetX('0.5');

$stamper2->setOffsetY('0.5');

/\*\*骑缝章签署\*\*/

$stamper3 = new Stamper();

$stamper3->setType('ACROSS\_PAGE');

$stamper3->setDocumentId('2590764888637018192');

$stamper3->setSealId('2555244623418466517');

$stamper3->setOffsetY('0.5');

$stampers = array();

array\_push($stampers, $stamper1);

array\_push($stampers, $stamper2);

$contractCompanySignRequest->setStampers($stampers);

$result = $sdkClient->service($contractCompanySignRequest);

print\_r($result);

return $result;

**Python示例**

​# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

seal\_signParam = SignParam()

seal\_signParam.set\_contractId(draft\_contractid)

# 指定签署位置 - 公章签署位置

seal\_companyStamper = Stamper()

seal\_companyStamper.set\_documentId(file\_documentId)

seal\_companyStamper.set\_sealId('2490828768980361630')

seal\_companyStamper.set\_type('COMPANY')

seal\_companyStamper.set\_offsetX(0.3)

seal\_companyStamper.set\_offsetY(0.5)

seal\_companyStamper.set\_page(1)

# 指定签署位置 - 时间戳签署位置

time\_companyStamper = Stamper()

time\_companyStamper.set\_documentId(file\_documentId)

time\_companyStamper.set\_type('TIMESTAMP')

time\_companyStamper.set\_offsetX(0.5)

time\_companyStamper.set\_offsetY(0.3)

time\_companyStamper.set\_page(1)

# 指定签署位置 - 骑缝章签署位置（文档页数大于1页才会生效）

acrosspage\_companyStamper = Stamper()

acrosspage\_companyStamper.set\_documentId(file\_documentId)

acrosspage\_companyStamper.set\_sealId('2490828768980361630')

acrosspage\_companyStamper.set\_type('ACROSS\_PAGE')

acrosspage\_companyStamper.set\_offsetY(0.7)

seal\_signParam.set\_stampers([seal\_companyStamper, time\_companyStamper, acrosspage\_companyStamper])

sealsign\_response = sdkClient.request(ContractSignCompanyRequest(seal\_signParam))

# 解析返回参数

sealsign\_mapper = json.loads(sealsign\_response)

if sealsign\_mapper['code'] != 0:

raise Exception('公章签署失败，失败原因：', sealsign\_mapper['message'])

print('公章签署成功')

#### 签署法人章

**请求地址：**/v2/contract/legalpersonsign

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| stampers | Stamper | 否 | 指定签署位置 |

Stamper（签署位置）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| type | String | 是 | 签署类型： LP（法人章），TIMESTAMP（时间戳） |
| documentId | String | 是 | 合同文档ID |
| keyword | String | 否 | 关键字 |
| keywordIndex | Integer | 否 | 关键字索引：1代表第1个关键字， 0代码所有关键字，-1代表倒数第1个关键字；默认为1 |
| page | Integer | 否 | 坐标页码，0代表所有 |
| offsetX | Decimal | 否 | 横坐标/关键字偏移量 |
| offsetY | Decimal | 否 | 纵坐标/关键字偏移量 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1106 | INVALID\_STAMPER\_TYPE，无效的签章类型；签署法人章支持的签章类型如下： LP（法人章），TIMESTAMP（时间戳）。 |
| 1107 | NOT SIGN STEP，未挨到签署； 未挨到公司签署法人章或者没有签署法人章节点。 |
| 1109 | SIGN PAGE BEYOND，签署页码超出文档页数； 签署位置中的页码超出文档页数。 |
| 1201 | SEAL NOT FOUND，找不到法人章； 请确保在契约锁已维护法人章。 |

**请求示例：**

**Http示例**

​ POST /v2/contract/legalpersonsign HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": null,

"stampers": [

{

"type": "LP",

"documentId": "2591542337725374704",

"offsetX": "0.1",

"offsetY": "-0.1",

"keyword": "劳动",

"keywordIndex": "2"

},

{

"type": "LP",

"documentId": "2591542337725374704",

"offsetX": "0.5",

"offsetY": "0.5",

"page": "1"

}

]

}

**Java示例**

// 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 签署法人章

SignParam param = new SignParam();

param.setContractId(2589394742435041330L);

ContractSignLpRequest request = new ContractSignLpRequest(param);

String response = sdkClient.service(request);

SdkResponse<Object> responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("法人章签署成功");

} else {

logger.info("法人章签署失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

ContractSignLPRequest request = new ContractSignLPRequest();

request.ContractId = "2589012016299597907";

//法人章签署位置

Stamper stamper1 = new Stamper();

stamper1.Type = "LP";

stamper1.DocumentId = "2589013696294510740";

stamper1.OffsetX = 0.06;

stamper1.OffsetY = 0.01;

stamper1.Keyword = "核对人";

request.AddStampers(stamper1);

//时间戳签署位置

Stamper stamper3 = new Stamper();

stamper3.Type = "TIMESTAMP";

stamper3.DocumentId = "2589013696294510740";

stamper3.OffsetX = 0.06;

stamper3.OffsetY = 0.01;

stamper3.Keyword = "核对单位盖章";

request.AddStampers(stamper3);

//法人章签署位置

Stamper stamper2 = new Stamper();

stamper2.Type = "LP";

stamper2.DocumentId = "2589013583094440075";

stamper2.OffsetX = 0.2;

stamper2.OffsetY = 0.4;

stamper2.Page = 1;

request.AddStampers(stamper2);

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractLpSignRequest = new ContractLpSignRequest();

$contractLpSignRequest->setContractId('2590804900506210559');

$contractLpSignRequest->setBizId("1111111");

/\*\*关键字定位\*\*/

$stamper1 = new Stamper();

$stamper1->setType('LP');

$stamper1->setDocumentId('2590805031263637783');

$stamper1->setKeyword('劳动');

$stamper1->setKeywordIndex('2');

$stamper1->setOffsetX('0.1');

$stamper1->setOffsetY('-0.1');

/\*\*坐标定位\*\*/

$stamper2 = new Stamper();

$stamper2->setType('LP');

$stamper2->setDocumentId('2590805031263637783');

$stamper2->setPage('1');

$stamper2->setOffsetX('0.5');

$stamper2->setOffsetY('0.5');

$stampers = array();

array\_push($stampers, $stamper1);

array\_push($stampers, $stamper2);

$contractLpSignRequest->setStampers($stampers);

$result = $sdkClient->service($contractLpSignRequest);

print\_r($result);

return $result;

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret

lp\_signParam = SignParam()

lp\_signParam.set\_contractId(draft\_contractid)

# 指定签署位置 - 法人章

seal\_lpStamper = Stamper()

seal\_lpStamper.set\_documentId(file\_documentId)

seal\_lpStamper.set\_type('LP')

seal\_lpStamper.set\_offsetX(0.6)

seal\_lpStamper.set\_offsetY(0.1)

seal\_lpStamper.set\_page(1)

# 指定签署位置 - 时间戳

time\_lpStamper = Stamper()

time\_lpStamper.set\_documentId(file\_documentId)

time\_lpStamper.set\_type('TIMESTAMP')

time\_lpStamper.set\_offsetY(0.9)

time\_lpStamper.set\_offsetX(0.8)

time\_lpStamper.set\_page(1)

lp\_signParam.set\_stampers([seal\_lpStamper, time\_lpStamper])

lp\_response = sdkClient.request(ContractSignLpRequest(lp\_signParam))

# 解析返回数据

lp\_mapper = json.loads(lp\_response)

if lp\_mapper['code'] != 0:

raise Exception('法人章签署失败，失败原因：', lp\_mapper['message'])

print('法人章签署完成')

#### 审批

**请求地址：**/v2/contract/employeeaudit

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| pass | Bolean | 是 | 审批是否通过 |
| comment | String | 否 | 审批意见 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1107 | NOT SIGN STEP，未挨到签署； 未挨到员工审批，或者没有员工审批节点。 |

**请求示例：**

**Http示例**

​ POST /v2/contract/employeeaudit HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": null,

"pass": true,

"comment": "同意"

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 审批

ContractAuditRequest request = new ContractAuditRequest(contractId, true, "审批通过");

String response = sdkClient.service(request);

SdkResponse<Object> responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("审批成功");

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 组装员工审批请求

ContractAuditRequest request = new ContractAuditRequest("2589012016299597907",true, "审批通过");

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("员工审批失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“员工审批成功”);

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractAuditRequest = new ContractAuditRequest();

$contractAuditRequest->setContractId('2590804900506210559');

$contractAuditRequest->setPass(true);

$contractAuditRequest->setComment('同意');

$result = $sdkClient->service($contractAuditRequest);

print\_r($result);

return $result;

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 合同审批

audit\_response = sdkClient.request(ContractAuditRequest(draft\_contractid, True, '符合要求，审核通过'))

# 解析返回数据

audit\_mapper = json.loads(audit\_response)

if audit\_mapper['code'] != 0:

raise Exception('审批合同失败，失败原因：', audit\_mapper['message'])

print('合同审批成功')

#### 合同催签

**请求地址：**/v2/contract/notice

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| signatoryId | String | 否 | 签署方ID |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**请求示例：**

**Http示例**

​ POST /v2/contract/notice HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": "123456789",

"signatoryId": "2591540371314024472"

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 合同催签

ContractNoticeRequest request = new ContractNoticeRequest(contractId);

String response = sdkClient.service(request);

SdkResponse responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("合同催签成功");

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(serverUrl, accessKey, accessSecret);

ContractNoticeRequest request = new ContractNoticeRequest();

request.ContractId = "2588924049895063561";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("合同催签失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“合同催签成功”);

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractNoticeRequest = new ContractNoticeRequest();

$contractNoticeRequest->setContractId('2590804900506210559');

$contractNoticeRequest->setBizId('123456789');

$contractNoticeRequest->setSignatoryId('2590804962716127509');

$result = $sdkClient->service($contractNoticeRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 签署通知

notice\_request = ContractNoticeRequest()

notice\_request.set\_contractId('2589643804158468448')

notice\_response = sdkClient.request(notice\_request)

print('发送签署通知成功：\n', str(notice\_response), '\n')

#### 撤回/作废合同

**描述：**调用此接口撤回/作废合同。

合同是“草稿”状态时，删除合同； 合同是“签署中”状态时，撤回合同，合同变为“已撤回”状态； 合同是“已完成”状态时，作废合同（作废合同需要所有签署方签署作废文件后，作废完成），同时在作废文件上签署发起方公章。

**请求地址：**/v2/contract/invalid

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |
| sealId | String | 否 | 印章ID，作废合同时使用 发起方签署作废文件时指定的印章 |
| reason | String | 否 | 撤回/作废原因 |
| deleteDoc | Boolean | 否 | 作废完成后是否删除合同文件，默认false |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1101 | INVALID CONTRACT STATUS，无效的合同状态；以下是支持的合同状态及对应操作：  DRAFT（草稿）：删除合同；  FILLING（填参中），SIGNING（签署中）：撤回合同；  COMPLETE（已完成）：发起作废、并签署发起方。 |

**请求示例：**

**Http示例**

​ POST /v2/contract/invalid HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": "123456789",

"sealId": "2589678690921123947",

"reason": "作废",

"deleteDoc": false

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// “签署中”状态下撤回合同

ContractInvalidRequest request = new ContractInvalidRequest(2589413512234848660L, "撤回合同");

String response = sdkClient.service(request);

SdkResponse responseObj = JSONUtils.toQysResponse(response);

if(responseObj.getCode() == 0) {

logger.info("合同撤回成功");

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

// “已完成”状态下请求作废合同，同时发起方签署作废合同

// ContractInvalidRequest request = new ContractInvalidRequest(2589382258757710050L, null, "请求作废合同", false);

// String response = sdkClient.service(request);

// SdkResponse responseObj = JSONUtils.toQysResponse(response);

// if(responseObj.getCode() == 0) {

// logger.info("合同作废请求成功");

// } else {

// logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

// }

// “草稿”状态下删除合同

// ContractInvalidRequest request = new ContractInvalidRequest(2589382258757710050L);

// String response = sdkClient.service(request);

// SdkResponse responseObj = JSONUtils.toQysResponse(response);

// if(responseObj.getCode() == 0) {

// logger.info("合同删除请求成功");

// } else {

// logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

// }

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

//“签署中”状态下撤回合同，“已完成”状态下请求作废合同，同时发起方签署作废合同

// “草稿”状态下删除合同

ContractInvalidRequest request = new ContractInvalidRequest();

request.ContractId = "2588924049895063561";

request.Reason = "不符合预期";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("合同撤回，作废失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“合同撤回，作废成功”);

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractInvalidRequest = new ContractInvalidRequest();

$contractInvalidRequest->setContractId('2590804900506210559');

$contractInvalidRequest->setBizId('123456789');

$contractInvalidRequest->setSealId('2589678690921123947');

$contractInvalidRequest->setReason('作废');

$contractInvalidRequest->setDeleteDoc(false);

$result = $sdkClient->service($contractInvalidRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# “签署中”状态下撤回合同，“已完成”状态下请求作废合同，同时发起方签署作废合同

# “草稿”状态下删除合同

invalid\_request = ContractInvalidRequest()

invalid\_request.set\_contractId('2589643804158468448')

invalid\_request.set\_reason('python invalid')

invalid\_response = sdkClient.request(invalid\_request)

print('作废成功：\n', str(invalid\_response), '\n')

## 合同页面

### 签署页面

**请求地址：**/v2/contract/pageurl

**请求方法：**POST

**请求格式：**application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 是 | 合同ID |
| user | User | 是 | 操作人 |
| callbackPage | String | 否 | 回调页面 |

User（个人用户）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系类型：MOBILE（手机号），  EMAIL（邮箱），EMPLOYEEID（员工ID），NUMBER（员工编号） |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 签署页面信息 |

Result（签署页面数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| pageUrl | String | 文件签署页面 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1110 | NO VIEW PERMISSION，没有查看权限； 用户没有权限查看合同。 |

**请求示例：**

**Http示例**

​ POST /v2/contract/pageurl HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contractId": "2591540368898105360",

"bizId": null,

"user": {

"contact": "10100000000",

"contactType": "MOBILE"

},

"callbackPage": null

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 合同页面

ContractPageRequest request = new ContractPageRequest(2589382258757710050L,

new User("10100000000", "MOBILE"), "");

String response = sdkClient.service(request);

SdkResponse<ContractPageResult> responseObj = JSONUtils.toQysResponse(response, ContractPageResult.class);

if(responseObj.getCode() == 0) {

ContractPageResult result = responseObj.getResult();

logger.info("合同页面地址为:{}", result.getPageUrl());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

ContractPageRequest request = new ContractPageRequest();

request.ContractId = contractId; //内嵌页面签署合同ID

request.User = new User("10100000000", "MOBILE"); //内嵌页面签署人信息

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception("生成签署链接请求服务器失败，失败原因：" + e.Message);

}

SdkResponse<PageUrlResult> sdkResponse = HttpJsonConvert.DeserializeResponse<PageUrlResult>(response);

if (!sdkResponse.Code.Equals(0))

{

throw new Exception("生成签署链接失败，失败原因：" + sdkResponse.Message);

}

Console.WriteLine(“生成合同页面成功”);

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.me";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractAuditRequest = new ContractAuditRequest();

$contractAuditRequest->setContractId('2590804900506210559');

$audutOperator = new User();

$audutOperator->setContact('10100000000');

$audutOperator->setContactType('MOBILE');

$contractAuditRequest->setEmployee($audutOperator);

$contractAuditRequest->setPass(true);

$contractAuditRequest->setComment('同意');

$result = $sdkClient->service($contractAuditRequest);

print\_r($result);

return $result;

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 组装合同页面请求

page\_request = ContractPageRequest(draft\_contractid)

sign\_user = User()

sign\_user.set\_contact("10100000000")

sign\_user.set\_contactType("MOBILE")

page\_request.set\_user(sign\_user)

response = sdkClient.request(page\_request)

page\_mapper = json.loads(response)

if page\_mapper['code'] != 0:

raise Exception('签署链接生成失败，失败原因：', page\_mapper['message'])

page\_result = page\_mapper['result']

print('签署链接签署成功，链接：', page\_result['pageUrl'])

### 浏览页面

**请求地址：**/v2/contract/viewurl

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID于业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID于业务ID二选一，不能同时为空 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 签署页面信息 |

Result（浏览页面数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| pageUrl | String | 文件签署页面 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1401 | DOCUMENT REQUIRED，合同不包含任何文件不允许查看 |

**请求示例：**

**Http示例**

GET /v2/contract/viewurl?contractId=2604642423930777609 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的TimeStamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

// 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

ContractViewPageRequest request=new ContractViewPageRequest(2604642423930777609L);

String response = sdkClient.service(request);

SdkResponse<ContractPageResult> responseObj = JSONUtils.toQysResponse(response, ContractPageResult.class);

if(responseObj.getCode() == 0) {

ContractPageResult result = responseObj.getResult();

logger.info("合同页面地址为:{}", result.getPageUrl());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

ContractViewPageRequest request = new ContractViewPageRequest();

request.ContractId = "2604642423930777609";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

SdkResponse<PageUrlResult> responseObject = HttpJsonConvert.DeserializeResponse<PageUrlResult>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$contractViewPageRequest = new ContractViewPageRequest();

$contractViewPageRequest->setContractId('2605319579798344136');

$contractViewPageRequest->setBizId('12345678');

$result = $sdkClient->service($contractViewPageRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

view\_page\_request = ContractViewPageRequest()

view\_page\_request.set\_contractId('2604642423930777609')

view\_page\_response = sdkClient.request(view\_page\_request)

print('合同浏览 \n', str(view\_page\_response), '\n')

## ****查询合同****

### 合同详情

**请求地址：**/v2/contract/detail

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID于业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID于业务ID二选一，不能同时为空 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Contract | 合同详情，参考创建合同的请求参数 |

**请求示例：**

**Http示例**

​ GET /v2/contract/detail?contractId=2591540368898105360 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 合同详情

ContractDetailRequest request = new ContractDetailRequest(contractId);

String response = sdkClient.service(request);

SdkResponse<Contract> responseObj = JSONUtils.toQysResponse(response, Contract.class);

if(responseObj.getCode() == 0) {

Contract contract = responseObj.getResult();

logger.info("合同详情查询，合同主题：{}", contract.getSubject());

} else {

logger.info("请求失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "http://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

string contractId = "2589012016299597907";

ContractDetailRequest request = new ContractDetailRequest();

request.ContractId = contractId;

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Contract> responseObject = HttpJsonConvert.DeserializeResponse<Contract> (response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取合同详情失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取合同详情成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.me";

const accessKey = "0VPomO**";**

**const accessSecret = "TckV2xbfKJU0iN37v2anFt22NH**";

}

$sdkClient = Util::getSDk();

$contractDetailRequest = new ContractDetailRequest();

$contractDetailRequest->setContractId('2590804900506210559');

$result = $sdkClient->service($contractDetailRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "http://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 合同详情

detail\_request = ContractDetailRequest()

detail\_request.set\_contracId('2589643804158468448')

detail\_response = sdkClient.request(detail\_request)

print('查询合同详情：\n', str(detail\_response), '\n')

## 下载合同

### 下载合同与签署日志

**请求地址：**/v2/contract/download

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |

**返回参数：**

合同文件和签署日志的压缩包（.zip）

**请求示例：**

**Http示例**

GET /v2/contract/download?contractId=2591540368898105360 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

// 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 下载合同

ContractDownloadRequest request = new ContractDownloadRequest(contractId);

FileOutputStream fos = new FileOutputStream("E:/test/contract.zip");

sdkClient.download(request, fos);

IOUtils.safeClose(fos);

logger.info("下载合同成功");

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

//下载合同

ContractDownloadRequest request = new ContractDownloadRequest();

request.ContractId = "2589012016299597907";

Stream stream = new FileStream("C:\Users\Richard Cheung\Documents\契约锁\合同压缩包.zip", FileMode.Create);

try

{

client.Download(request, ref stream);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$filePath = "D:/contract-php-test.zip";

$contractDownloadRequest = new ContractDownloadRequest();

$contractDownloadRequest->setContractId('2590804900506210559');

$contractDownloadRequest->setBizId('123456789');

$result = $sdkClient->downloadService($contractDownloadRequest, $filePath);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 下载所有合同文件

contractdownload\_request = ContractDownloadRequest()

contractdownload\_request.set\_contractId('2589643804158468448')

contract\_zip = open("C:\Users\Richard Cheung\Documents\契约锁\合同文档.zip", "wb")

contract\_zip.write(sdkClient.request(contractdownload\_request))

print('下载文件成功 \n')

### 下载合同文档

**请求地址：**/v2/document/download

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| documentId | String | 是 | 合同文档ID |

**返回参数：**

合同文件（.pdf）

**请求示例：**

**Http示例**

​GET /v2/document/download?documentId=2591542337725374704 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​// 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 下载合同文档

DocumentDownloadRequest request = new DocumentDownloadRequest(2586891626797203479L);

FileOutputStream fos = new FileOutputStream("E:/test/doc.pdf");

sdkClient.download(request, fos);

IOUtils.safeClose(fos);

logger.info("下载合同文档成功");

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

//下载合同

string documentId = "2589013696294510740";

DocumentDownloadRequest request = new DocumentDownloadRequest();

request.DocumentId = documentId;

Stream stream = new FileStream("C:\Users\Richard Cheung\Documents\契约锁\document.pdf", FileMode.Create);

try

{

client.Download(request, ref stream);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$filePath = "D:/php-test.pdf";

$documentDownloadRequest = new DocumentDownloadRequest('2590805031263637783');

$result = $sdkClient->downloadService($documentDownloadRequest, $filePath);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 下载单个文件

documentdownload\_request = DocumentDownloadRequest()

documentdownload\_request.set\_documentId('2589649754185539990')

document\_pdf = open("C:\Users\Richard Cheung\Documents\契约锁\py.pdf", "wb")

document\_pdf.write(sdkClient.request(documentdownload\_request))

print('下载单个文件成功 \n')

### 下载合同附件

**请求地址：**/v2/attachment/download

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contractId | String | 否 | 合同ID，合同ID与业务ID二选一，不能同时为空 |
| bizId | String | 否 | 业务ID，合同ID与业务ID二选一，不能同时为空 |

**返回参数：**

合同附件压缩包（.zip）

**请求示例：**

**Http示例**

​GET /v2/attachment/download?contractId=2591540368898105360 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​// 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 下载合同附件

AttachmentDownloadRequest request = new AttachmentDownloadRequest(contractId);

FileOutputStream fos = new FileOutputStream("E:/test/attachment.pdf");

sdkClient.download(request, fos);

IOUtils.safeClose(fos);

logger.info("下载附件成功");

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

//下载附件

AttachmentDownloadRequest request = new AttachmentDownloadRequest();

request.ContractId = "2589012016299597907";

Stream stream = new FileStream("C:\Users\Richard Cheung\Documents\契约锁\attachment.zip", FileMode.Create);

try

{

client.Download(request, ref stream);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$filePath = "D:/attachment-php-test.zip";

$attachmentDownloadRequest = new AttachmentDownloadRequest();

$attachmentDownloadRequest->setContractId('2590804900506210559');

$attachmentDownloadRequest->setBizId('123456789');

$result = $sdkClient->downloadService($attachmentDownloadRequest, $filePath);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 下载附件信息

attachmentdownload\_request = AttachmentDownloadRequest()

attachmentdownload\_request.set\_contractId('2589643804158468448')

attachment\_zip = open("C:\Users\Richard Cheung\Documents\契约锁\attachment.zip", "wb")

attachment\_zip.write(sdkClient.request(attachmentdownload\_request))

print('附件下载成功 \n')

## 公司接口

### 公司信息

**请求地址：**/v2/company/detail

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| companyName | String | 是 | 公司名称 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Company | 公司数据 |

Company（公司数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| status | String | 公司状态：UNREGISTERED（未注册），  CERTIFYING（认证中），AUTH\_SUCCESS（认证成功） |
| id | String | 公司ID |
| name | String | 公司名称 |
| registerNo | String | 公司代码（工商注册号/统一社会信用代码） |

**请求示例：**

**Http示例**

​ GET /v2/company/detail?companyName=测试公司 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 公司信息

CompanyDetailRequest request = new CompanyDetailRequest("测试公司");

String response = sdkClient.service(request);

SdkResponse<Company> responseObj = JSONUtils.toQysResponse(response, Company.class);

if(responseObj.getCode() == 0) {

Company result = responseObj.getResult();

logger.info("公司查询，状态:{}", result.getStatus());

} else {

logger.info("公司查询失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 获取公司信息

CompanyDetailRequsest request = new CompanyDetailRequsest();

request.CompanyName = "测试公司";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<CompanyDetailResult> responseObject = HttpJsonConvert.DeserializeResponse<CompanyDetailResult>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取公司信息失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取公司信息成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$companyDetailRequest = new CompanyDetailRequest("测试企业");

$result = $sdkClient->service($companyDetailRequest);

print\_r($result);

return $result;

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 公司信息

request = CompanyDetailRequest("测试公司")

response = sdkClient.request(request)

print('获取公司详情：\n', str(response), '\n')

### 对接方信息

**请求地址：**/company/platforminfo

**请求方法：**GET

**请求参数：**无

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Company | 公司数据 |

Company（公司数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| status | String | 公司状态：UNREGISTERED（未注册），  CERTIFYING（认证中），AUTH\_SUCCESS（认证成功） |
| id | String | 公司ID |
| name | String | 公司名称 |
| registerNo | String | 公司代码（工商注册号/统一社会信用代码） |

**请求示例：**

**Http示例**

​ GET /company/platforminfo HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 公司信息

PlatformDetailRequest request = new PlatformDetailRequest();

String response = sdkClient.service(request);

SdkResponse<Company> responseObj = JSONUtils.toQysResponse(response, Company.class);

if(responseObj.getCode() == 0) {

Company result = responseObj.getResult();

logger.info("平台信息查询，名称:{}", result.getName());

} else {

logger.info("平台信息查询失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

string response = null;

try

{

response = client.Service(new PlatformDetailRequest());

}

catch(Exception e)

{

throw new Exception(e.Message);

}

SdkResponse<CompanyDetailResult> responseObject = HttpJsonConvert.DeserializeResponse<CompanyDetailResult>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取平台信息失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取平台信息成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$platformInfoRequest = new PlatformInfoRequest();

$result = $sdkClient->service($platformInfoRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 公司信息

response = sdkClient.request(PlatformDetailRequest())

print('获取对接方公司详情：\n', str(response), '\n')

## 印章接口

### 印章列表

**请求地址：**/v2/seal/list

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| selectOffset | Integer | 否 | 查询起始位置，默认为0 |
| selectLimit | Integer | 否 | 查询列表大小，默认1000 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 印章数据 |

Result（印章数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| totalCount | Integer | 印章数量 |
| list | Array[Seal] | 印章列表 |

Seal（印章数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 印章ID |
| name | String | 印章名称 |
| createTime | String | 创建时间，格式为：yyyy-MM-dd HH:mm:ss |

**请求示例：**

**Http示例**

​ GET /v2/seal/list?selectOffset=0& selectLimit=2 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 印章列表查询

SealListRequest request = new SealListRequest();

String response = sdkClient.service(request);

SdkResponse<SealListResult> responseObj = JSONUtils.toQysResponse(response, SealListResult.class);

if(responseObj.getCode() == 0) {

SealListResult sealList = responseObj.getResult();

logger.info("印章列表查询，数量:{}", sealList.getTotalCount());

} else {

logger.info("印章列表失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 印章列表查询

SealListRequest request = new SealListRequest(new PageBean(0, 5));

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<PageListResult<Category>> responseObject = HttpJsonConvert.DeserializeResponse<PageListResult<Category>>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取印章列表失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取印章列表成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$selectOffset = 0;

$selectLimit = 1;

$baseRequest = new SealListRequest($selectOffset, $selectLimit);

$result = $sdkClient->service($baseRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 获取印章列表

list\_request = SealListRequest(0, 5)

list\_response = sdkClient.request(list\_request)

print('获取印章列表：\n', str(list\_response), '\n')

### 印章图片

**请求地址：**/v2/seal/image

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| sealId | String | 是 | 印章ID |

**返回参数：**

印章图片流

**请求示例：**

**Http示例**

​ GET /v2/seal/image?sealId=2566229349702860958 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signaure]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 印章图片下载

SealImageRequest request = new SealImageRequest(2391453670764376075L);

FileOutputStream fos = new FileOutputStream("E:/test/seal.png");

sdkClient.download(request, fos);

fos.close();

logger.info("下载印章成功");

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 下载印章图片

Stream stream = new FileStream("C:\Users\Richard Cheung\Documents\契约锁\印章图片.png", FileMode.Create);

SealImageRequest request = new SealImageRequest(2565830903727656983);

try

{

client.Download(request, ref stream);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$sealImageRequest = new SealImageRequest("2555244623418466517");

$path = "D:/cs2.png";

$result = $sdkClient->downloadService($sealImageRequest, $path);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 下载印章图片

image\_request = SealImageRequest('2565830903727656983')

image\_response = sdkClient.request(image\_request)

file = open("C:\Users\Richard Cheung\Documents\契约锁\印章图片.png", "wb")

file.write(image\_response)

file.close()

print('印章下载成功')

## 文件模板

### 模板列表

**请求地址：**/v2/template/list

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| selectOffset | Integer | 否 | 查询起始位置，默认为0 |
| selectLimit | Integer | 否 | 查询列表大小，默认1000 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 文件模板数据 |

Result（文件模板数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| totalCount | Integer | 模板数量 |
| list | Array[Template] | 模板列表 |

Template（文件模板）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 模板ID |
| name | String | 模板名称 |
| createTime | String | 创建时间，格式为：yyyy-MM-dd HH:mm:ss |
| parameters | Array[TemplateParam] | 模板参数 |

TemplateParam（模板参数）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| name | String | 参数名称 |
| required | Boolean | 是否必填 |
| type | String | 参数类型：text(文本), textarea(多行文本), date(时间), person(身份证号), radio(单选), checkbox(多选), seal(签章), signature(签名) |

**请求示例：**

**Http示例**

​ GET /v2/template/list?selectOffset=0& selectLimit=2 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 模板列表

TemplateListRequest request = new TemplateListRequest();

String response = sdkClient.service(request);

SdkResponse<TemplateListResult> responseObj = JSONUtils.toQysResponse(response, TemplateListResult.class);

if(responseObj.getCode() == 0) {

TemplateListResult templateList = responseObj.getResult();

logger.info("文件模板列表查询，数量:{}", templateList.getTotalCount());

} else {

logger.info("文件模板列表失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 查询模板列表

TemplateListRequest request = new TemplateListRequest(new PageBean());

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<PageListResult<Template>> responseObject = HttpJsonConvert.DeserializeResponse<PageListResult<Template>>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取模板列表失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取模板列表成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$selectOffset = 0;

$selectLimit = 1;

$baseRequest = new TemplateListRequest($selectOffset, $selectLimit);

$result = $sdkClient->service($baseRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 查询模板列表

listRequest = TemplateListRequest(0, 5)

response = sdkClient.request(listRequest)

print('查询模板列表：\n', str(response), '\n')

### 模板预览页面

**请求地址：**/v2/template/pageurl

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| templateId | String | 是 | 模板ID |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 模板页面数据 |

Result（模板页面数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| pageUrl | String | 模板预览页面 |

**请求示例：**

**Http示例**

​ GET /v2/template/pageurl?templateId=2427320111455567969 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 模板预览页面

TemplatePageRequest request = new TemplatePageRequest(2586924492742590568L);

String response = sdkClient.service(request);

SdkResponse<TemplatePageResult> responseObj = JSONUtils.toQysResponse(response, TemplatePageResult.class);

if(responseObj.getCode() == 0) {

TemplatePageResult result = responseObj.getResult();

logger.info("模板页面查询，页面链接:{}", result.getPageUrl());

} else {

logger.info("模板页面查询失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 模板预览页面

TemplatePageRequest request = new TemplatePageRequest();

request.TemplateId = "2474165699643592754";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<PageUrlResult> responseObject = HttpJsonConvert.DeserializeResponse<PageUrlResult>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取模板预览页失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取模板预览页成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$templatePageRequest = new TemplatePageRequest();

$templatePageRequest->setTemplateId('2558596891082797841');

$result = $sdkClient->service($templatePageRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 获取模板预览页面

pageRequest = TemplatePageRequest('2474165699643592754')

response = sdkClient.request(pageRequest)

print('模板预览页面：\n', str(response), '\n')

## 员工接口

### 员工列表

**请求地址：**/v2/employee/list

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| selectOffset | Integer | 否 | 查询起始位置，默认为0 |
| selectLimit | Integer | 否 | 查询列表大小，默认1000 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 员工数据 |

Result（员工数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| totalCount | Integer | 员工数量 |
| list | Array[Employee] | 员工列表 |

Employee（员工信息）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 员工ID |
| name | String | 员工姓名 |
| idCardNo | String | 身份证号 |
| mobile | String | 手机号 |
| email | String | 邮箱 |
| number | String | 员工编号 |
| createTime | String | 创建时间 |

**请求示例：**

**Http示例**

​ GET /v2/employee/list?selectOffset=0& selectLimit=2 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 员工列表

EmployeeListRequest request = new EmployeeListRequest();

String response = sdkClient.service(request);

SdkResponse<EmployeeListResult> responseObj = JSONUtils.toQysResponse(response, EmployeeListResult.class);

if(responseObj.getCode() == 0) {

EmployeeListResult result = responseObj.getResult();

logger.info("员工查询，员工数量：{}", result.getTotalCount());

} else {

logger.info("员工查询失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 员工列表

EmployeeListRequest request = new EmployeeListRequest(new PageBean());

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<PageListResult<Employee>> responseObject = HttpJsonConvert.DeserializeResponse<PageListResult<Employee>>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取员工列表失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取员工列表成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$selectOffset = 0;

$selectLimit = 10;

$employeeListRequest = new EmployeeListRequest($selectOffset, $selectLimit);

$result = $sdkClient->service($employeeListRequest);

print\_r($result);

**Python示例​**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 员工列表

list\_request = EmployeeListRequest()

list\_request.set\_selectOffset(0)

list\_request.set\_selectLimit(5)

list\_response = sdkClient.request(list\_request)

print('员工列表查询：\n', str(list\_response), '\n')

### 添加员工

**请求地址：**/v2/employee/create

**请求方法：**POST

**请求格式:** application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| user | User | 是 | 员工信息，用于确定员工身份 |
| number | String | 否 | 员工编号 |

User（个人用户）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| name | String | 是 | 姓名 |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系类型：MOBILE（手机号），EMAIL（邮箱），EMPLOYEEID（员工ID），NUMBER（员工编号） |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Employee | 员工数据 |

Employee（员工信息）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 员工ID |
| name | String | 员工姓名 |
| idCardNo | String | 身份证号 |
| mobile | String | 手机号 |
| email | String | 邮箱 |
| number | String | 员工编号 |
| createTime | String | 创建时间 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1703 | EMPLOYEE NAME NOT MATCH，员工的姓名和实际姓名不匹配； 员工在契约锁已认证时，传入的员工姓名必须与真实的员工姓名一致。 |

**请求示例：**

**Http示例**

​ POST /v2/employee/create HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"user": {

"name": "王五",

"contact": "15000000000",

"contactType": "MOBILE"

},

"number": "123456"

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 添加员工

User user = new User("宋三", "13636350224", "MOBILE");

EmployeeCreateRequest request = new EmployeeCreateRequest(user, "123127");

String response = sdkClient.service(request);

SdkResponse<Employee> responseObj = JSONUtils.toQysResponse(response, Employee.class);

if(responseObj.getCode() == 0) {

Employee result = responseObj.getResult();

logger.info("创建员工，员工ID：{}", result.getId());

} else {

logger.info("创建员工失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 添加员工

User employeeUser = new User("伍伶俐", "15317727840", "MOBILE");

EmployeeCreateRequest request = new EmployeeCreateRequest(employeeUser);

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<EmployeeAddResult> responseObject = HttpJsonConvert.DeserializeResponse<EmployeeAddResult>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("添加员工失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“添加员工成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$user = new User();

$user->setName("王五");

$user->setContact("15000000000");

$user->setContactType("MOBILE");

$employeeCreateRequest = new EmployeeCreateRequest();

$employeeCreateRequest->setUser($user);

$employeeCreateRequest->setNumber('123456');

$result = $sdkClient->service($employeeCreateRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 创建员工

user = User("操作人名称", "15317727840", "MOBILE")

crete\_request = EmployeeCreateRequest(user)

create\_response = sdkClient.request(crete\_request)

print('创建员工：\n', str(create\_response), '\n')

### 更新员工

**描述：**支持修改员工姓名和员工编号。

**请求地址：**/v2/employee/update

**请求方法：**POST

**请求格式:** application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| user | User | 是 | 员工信息，用于确定员工身份 |
| number | String | 否 | 员工编号 |

User（个人用户）：

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| name | String | 否 | 姓名 |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系类型：MOBILE（手机号），EMAIL（邮箱），EMPLOYEEID（员工ID），NUMBER（员工编号） |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1701 | EMPLOYEE NOT FOUND，找不到员工 |
| 1703 | EMPLOYEE NAME NOT MATCH，员工的姓名和实际姓名不匹配； 员工在契约锁已认证时，传入的员工姓名必须与真实的员工姓名一致。 |

**请求示例：**

**Http示例**

​ POST /v2/employee/update HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signaure]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"user": {

"name": "李四",

"contact": "15000000000",

"contactType": "MOBILE"

},

"number": 777777

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 更新员工

User user = new User("宋三", "13636350222", "MOBILE");

EmployeeUpdateRequest request = new EmployeeUpdateRequest(user, "123126");

String response = sdkClient.service(request);

SdkResponse<Employee> responseObj = JSONUtils.toQysResponse(response, Employee.class);

if(responseObj.getCode() == 0) {

logger.info("更新员工成功");

} else {

logger.info("更新失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 更新员工信息

User employeeUser = new User("伍伶俐", "15317727840", "MOBILE");

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.DeserializeResponse<Object>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$user = new User();

$user->setName("李四");

$user->setContact("15000000000");

$user->setContactType("MOBILE");

$employeeUpdateRequest = new EmployeeUpdateRequest();

$employeeUpdateRequest->setUser($user);

$employeeUpdateRequest->setNumber(777777);

$result = $sdkClient->service($employeeUpdateRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 修改员工数据

user = User("操作人名称", "15317727840", "MOBILE")

update\_request = EmployeeUpdateRequest(user, "55555")

update\_response = sdkClient.request(update\_request)

print('修改员工数据：\n', str(update\_response), '\n')

### 移除员工

**请求地址：**/v2/employee/remove

**请求方法：**POST

**请求格式:** application/json;charset=UTF-8

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系类型：MOBILE（手机号）， EMAIL（邮箱），EMPLOYEEID（员工ID） |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码，0为成功 |
| message | String | 响应消息 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1701 | EMPLOYEE NOT FOUND，找不到员工 |
| 1702 | EMPLOYEE DISMISSED，员工已移除 |

**请求示例：**

**Http示例**

​ POST /v2/employee/remove HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"contact": "15000000000",

"contactType": "MOBILE"

}

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 移除员工

User user = new User("宋三", "13636350224", "MOBILE");

EmployeeRemoveRequest request = new EmployeeRemoveRequest(user);

String response = sdkClient.service(request);

SdkResponse<Employee> responseObj = JSONUtils.toQysResponse(response, Employee.class);

if(responseObj.getCode() == 0) {

logger.info("移除员工成功");

} else {

logger.info("移除失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 移除员工

User removeUser = new User("删除员工名称", "删除员工联系方式", "MOBILE"); EmployeeRemoveRequest request = new EmployeeRemoveRequest(removeUser);

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

// 解析返回内容

SdkResponse<Object> responseObject = HttpJsonConvert.Deseria

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$employee = new User();

$employee->setContact("15000000000");

$employee->setContactType("MOBILE");

$employeeRemoveRequest = new EmployeeRemoveRequest();

$employeeRemoveRequest->setUser($employee);

$result = $sdkClient->service($employeeRemoveRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 删除员工数据

user = User("操作人名称", "删除员工联系方式", "MOBILE")

remove\_request = EmployeeRemoveRequest(user)

remove\_response = sdkClient.request(remove\_request)

print('删除员工数据：\n', str(remove\_response), '/n')

## 业务分类

### 业务分类列表

**请求地址：**/v2/category/list

**请求方法：**GET

**请求参数：**

| **参数** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| selectOffset | Integer | 否 | 查询起始位置，默认为0 |
| selectLimit | Integer | 否 | 查询列表大小，默认1000 |

**返回参数：**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | Result | 业务分类数据 |

Result（业务分类数据）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| totalCount | Integer | 业务分类数量 |
| list | Array[Category] | 业务分类列表 |

Category（业务分类）：

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| id | String | 业务分类ID |
| name | Array[Category] | 业务分类名称 |
| createTime | String | 创建时间，格式为：yyyy-MM-dd HH:mm:ss |

**请求示例：**

**Http示例**

​ GET /v2/category/list?selectOffset=0& selectLimit=2 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

​ // 初始化sdkClient

String serverUrl = "https://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

// 业务分类列表

CategoryListRequest request = new CategoryListRequest();

String response = sdkClient.service(request);

SdkResponse<CategoryListResult> responseObj = JSONUtils.toQysResponse(response, CategoryListResult.class);

if(responseObj.getCode() == 0) {

CategoryListResult categoryList = responseObj.getResult();

logger.info("业务分类列表查询，数量:{}", categoryList.getTotalCount());

} else {

logger.info("业务分类列表失败，错误码:{}，错误信息:{}", responseObj.getCode(), responseObj.getMessage());

}

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

// 查询业务分类列表

CategoryListRequest request = new CategoryListRequest(new PageBean());

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

SdkResponse<PageListResult<Category>> responseObject = HttpJsonConvert.DeserializeResponse<PageListResult<Category>>(response);

if (!responseObject.Code.Equals(0))

{

throw new Exception("获取业务分类列表失败，失败原因：" + responseObject.Message);

}

Console.WriteLine(“获取业务分类列表成功”);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

$selectOffset = 1;

$selectLimit = 1;

$baseRequest = new CategoryListRequest($selectOffset, $selectLimit);

$result = $sdkClient->service($baseRequest);

print\_r($result);

return $result;

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

# 查询业务分类列表

sdkClient = SdkClient(url, accessToken, accessSecret)

request = CategoryListRequest()

response = sdkClient.request(request)

print('查询业务分类列表：\n', str(response))

## 个人实名认证

### 获取个人认证链接

**请求地址：** /v2/personalauth

**请求方法：** POST

**请求参数：**

| **参数名称** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| mode | String | 否 | 实名认证模式，DEFAULT、FACE、BANK，默认为DEFAULT |
| user | UserInfo | 是 | 待认证用户信息 |
| username | String | 否 | 指定用户认证名称 |
| idCardNo | String | 否 | 指定用户身份证号 |
| bankNo | String | 否 | 指定用户银行卡号 |
| bankMobile | String | 否 | 指定用户银行卡预留手机号 |
| callbackUrl | String | 否 | 认证回调地址 |
| callbackPage | String | 否 | 跳转页面地址 |
| manualSwitch | Boolean | 否 | 是否允许使用人工审核通过认证，默认不开启 |

UserInfo :

| **名称** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系方式类型：MOBILE、EMAIL |

**返回参数：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | PersonAuthResponse | 认证信息 |

PersonAuthResponse：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| authUrl | String | 认证链接 |
| authId | String | 认证请求ID，用于查询认证结果 |

**请求示例：**

**Http示例**

POST /v2/personalauth HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"mode":"BANK",

"user":{

"contact":"10020033044",

"contactType":"MOBILE"

},

"username":"张三",

"callbackUrl":"http://www.qiyuesuo.com"

}

**Java示例**

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

UserAuthPageRequest request = new UserAuthPageRequest();

request.setUser(new User("10020033044", "MOBILE"));

String response = sdkClient.service(request);

SdkResponse < UserAuthPageResult > responseObj = JSONUtils.toQysResponse(response, UserAuthPageResult.class);

logger.info(responseObj.getResult().getAuthUrl());

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

string response = null;

UserAuthPageRequest request = new UserAuthPageRequest();

request.Mode = "DEFAULT";

request.User = new User("10020033044", "MOBILE");

string response = client.Service(request);

SdkResponse < UserAuthPageResult> responseObject = HttpJsonConvert.DeserializeResponse < UserAuthPageResult>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$sdkClient = Util::getSDk();

// 待认证的用户信息

$user = new User();

$user->setContactType("MOBILE");

$user->setContact("10020033044");

$userAuth = new UserAuth();

$userAuth->setMode("DEFAULT");

$userAuth->setUser($user);

$userAuthPageRequest = new UserAuthPageRequest();

$userAuthPageRequest->setUserAuth($userAuth);

$result = $sdkClient->service($userAuthPageRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

userAuthPageRequest = UserAuthPageRequest()

userAuthPageRequest.set\_user(User(contact='10020033044', contactType='MOBILE'))

response = sdkClient.request(userAuthPageRequest)

print(response)

### 查询个人认证状态

**请求地址：**/v2/personalauth/result

**请求方法：** GET

**请求参数：**

| **名称** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| contact | String | 否 | 认证用户联系方式，认证Id authId和用户信息 contact、contactType两者不能同时为空 |
| contactType | String | 否 | 认证用户联系方式类型，MOBILE("手机")、EMAIL("邮箱") |
| authId | String | 否 | 待查询的认证请求信息，认证Id authId和用户信息 contact、contactType两者不能同时为空 |

**返回参数：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Int | 响应码 |
| message | String | 响应消息 |
| result | PersonRealNameResponse | 认证信息 |

PersonRealNameResponse：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| realName | Boolean | 是否实名认证，是否认证成功 |
| user | UserRealNameInfo | 认证信息 |

UserRealNameInfo：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| name | String | 认证姓名 |
| contact | String | 认证联系方式 |
| idCardNo | String | 认证身份证号码 |
| bankNo | String | 认证银行卡号，仅传递authId，且完成对应银行卡认证后返回 |
| bankMobile | String | 认证银行卡预留手机号，仅传递authId，且完成对应银行卡认证后返回 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1604 | INVALID\_AUTHID，未匹配到认证信息 |

**请求示例：**

**Http示例**

GET /v2/personalauth/result?contact=10020033044& contactType=MOBILE HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

// 初始化sdkClient

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

UserAuthResultRequest request = new UserAuthResultRequest(new User("10020033044", "MOBILE"));

String response = sdkClient.service(request);

SdkResponse < UserAuthResult > responseObj = JSONUtils.toQysResponse(response, UserAuthResult.class);

logger.info(responseObj.getResult().getRealName().toString());

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

UserAuthResultRequest request = new UserAuthResultRequest();

request.User = new User("10020033044", "MOBILE");

string response = client.Service(request);

SdkResponse<UserAuthResult> responseObject = HttpJsonConvert.DeserializeResponse<UserAuthResult>(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

// 初始化$sdkClient

class Util {

const url = "https://openapi.qiyuesuo.cn";

const accessKey = "替换为您申请的开放平台App Token";

const accessSecret = "替换为您申请的开放平台App Secret";

}

$userAuthResultRequest = new UserAuthResultRequest();

$userAuthResultRequest->setContactType("MOBILE");

$userAuthResultRequest->setContact("10020033044");

$result = $sdkClient->service($userAuthResultRequest);

print\_r($result);

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

userAuthResultRequest = UserAuthResultRequest()

userAuthResultRequest.set\_user(User(contact='10020033044', contactType='MOBILE'))

response = sdkClient.request(userAuthResultRequest)

print(response)

### 回调说明

**描述：**接收回调请求的地址接口需要支持Content-Type: application/x-www-form-urlencoded 的post请求，每次回调不通会重复调用一次，如果调用不通，1分钟后再调用2次，一共调用4次。

**参数说明：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| mode | String | 认证模式：IVS：手机三要素认证、FACE：人脸认证、BANK：银行卡认证、MANUAL：人工审核 |
| conclusions | Conclusions | 认证信息 |
| status | String | 1:认证通过，2：认证不通过 |

Conclusions：

| **参数名称** | **类型** | **描述** |
| --- | --- | --- |
| authId | Stirng | 认证ID |
| username | String | 姓名 |
| mobile | String | 联系方式 |
| cardNo | String | 身份证号码 |
| bankNo | String | 银行卡号 |
| contact | String | 银行卡预留手机号 |
| realname | Boolean | 是否实名认证 |

接收回调请求接口的返回值： 接口返回值必须包含以下参数，且转换成json字符串返回。

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Int | 接收回调消息 响应码 |

返回值说明：

code为0表示接收回调消息成功： {"code":0} code不为0表示接收回调消息失败： {"code":1001}

## 企业认证

### 获取企业认证链接

**请求地址（H5）：**/companyauth/h5page

**请求地址（PC）：**/companyauth/pcpage

**请求方式：**POST

**请求参数：**

| **参数名称** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| companyName | String | 是 | 待认证公司名称 |
| applicant | UserInfo | 是 | 认证提交人，企业认证通过后，认证提交人会自动成为该企业的系统管理员 |
| registerNo | String | 否 | 待认证公司注册号 |
| legalPerson | String | 否 | 待认证公司法人姓名 |
| callbackUrl | String | 否 | 认证回调地址 |

UserInfo（提交人信息，企业认证通过后，提交人对应的账号会自动成为该公司的管理员） :

| **名称** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| name | String | 否 | 申请者姓名 |
| contact | String | 是 | 联系方式 |
| contactType | String | 是 | 联系方式类型：MOBILE、EMAIL |

**返回参数：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | CorpAuthResponse | 认证信息 |

CorpAuthResponse：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| authUrl | String | 认证链接 |
| requestId | String | 认证请求ID，用于查询认证结果 |

**响应码（全局响应码请查看文档末“全局响应码”）：**

| **响应码** | **描述** |
| --- | --- |
| 1605 | COMPANY ALREADY AUTHED，该公司已完成认证 |

**请求示例：**

**Http示例**

POST /companyauth/h5page HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

Content-Type: application/json

{

"companyName":"上海契约锁公司",

"applicant":{

"name":"张三",

"contact":"12312378900",

"contactType":"MOBILE"

},

"callbackUrl":"http://www.qiyuesuo.com"

}

**Java示例**

// 初始化sdkClient

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

User applicant = new User("张三", "12312378900", "MOBILE");

CompanyAuthH5PageRequest request = new CompanyAuthH5PageRequest("上海契约锁公司", applicant);

String response = sdkClient.service(request);

SdkResponse < CompanyAuthPageResult > pageResultResponse = JSONUtils.toQysResponse(response, CompanyAuthPageResult.class);

logger.info(pageResultResponse.getResult().getAuthUrl());

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

User applicant = new User("张三", "12312378900", "MOBILE");

CompanyAuthH5PageRequest request = new CompanyAuthH5PageRequest("上海契约锁公司", applicant);

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

SdkResponse <CompanyAuthPageResult > responseObject = HttpJsonConvert.DeserializeResponse <CompanyAuthPageResult >(response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

待支持

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

auth\_h5\_page\_request = CompanyAuthH5PageRequest()

auth\_h5\_page\_request.set\_companyName('上海契约锁公司')

applicant\_user = User();

applicant\_user.set\_name('张三')

applicant\_user.set\_contact('12312378900')

applicant\_user.set\_contactType('MOBILE')

auth\_h5\_page\_request.set\_applicant(applicant\_user)

auth\_h5\_page\_response = sdkClient.request(auth\_h5\_page\_request)

print(auth\_h5\_page\_response)

### 查询认证结果

**请求地址：**/companyauth/result

**请求方式：**GET

**请求参数：**

| **参数名称** | **类型** | **必须** | **描述** |
| --- | --- | --- | --- |
| companyName | String | 否 | 待查询的认证公司名称 |
| requestId | String | 否 | 认证请求ID |

**返回参数：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Integer | 响应码 |
| message | String | 响应消息 |
| result | CorpAuthInfo | 认证结果 |

CorpAuthInfo：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| status | Integer | 认证状态，-1（“无认证记录”）、1（“未提交认证申请”）、2（“认证通过”）、3（“认证不通过”）、4（“认证中”） |
| name | String | 认证企业名称 |
| registerNo | String | 认证企业的工商注册号 |
| legalPerson | String | 认证企业的法人代表姓名 |
| applicant | String | 申请者姓名 |
| applicantPhone | String | 申请者联系方式 |

**请求示例：**

**Http示例**

GET /companyauth/result?companyName=上海契约锁公司 HTTP/1.1

Host: [host]

x-qys-open-timestamp: [替换为请求头生成的Timestamp]

x-qys-open-signature: [替换为请求头生成的Signature]

x-qys-open-accesstoken: [替换为请求头生成的Token]

**Java示例**

// 初始化sdkClient

String serverUrl = "http://openapi.qiyuesuo.cn";

String accessKey = "替换为您申请的开放平台App Token";

String accessSecret = "替换为您申请的开放平台App Secret";

SdkClient sdkClient = new SdkClient(serverUrl, accessKey, accessSecret);

CompanyAuthResultRequest request = new CompanyAuthResultRequest();

request.setRequestId("f8967fcd-6bdc-42a2-a136-17dd6225d0dd");

String response = sdkClient.service(request);

SdkResponse < CompanyAuthResult > resultResponse = JSONUtils.toQysResponse(response,CompanyAuthResult.class);

logger.info(resultResponse.getResult().getStatus().toString());

**C#示例**

// 初始化sdkClient

string serverUrl = "https://openapi.qiyuesuo.cn";

string accessKey = "替换为您申请的开放平台App Token";

string accessSecret = "替换为您申请的开放平台App Secret";

SDKClient client = new SDKClient(accessKey, accessSecret, serverUrl);

CompanyAuthDetailRequest request = new CompanyAuthDetailRequest();

request.RequestId = "6110f2d0-4f65-4e1e-83c0-7a64ff747e68";

string response = null;

try

{

response = client.Service(request);

}

catch (Exception e)

{

throw new Exception(e.Message);

}

SdkResponse < CompanyAuthResult > responseObject = HttpJsonConvert.DeserializeResponse < CompanyAuthResult > (response);

Console.WriteLine(HttpJsonConvert.SerializeObject(responseObject));

**PHP示例**

待支持

**Python示例**

# 初始化SdkClient

url = "https://openapi.qiyuesuo.cn"

accessToken = '替换为您申请的开放平台App Token'

accessSecret = '替换为您申请的开放平台App Secret'

sdkClient = SdkClient(url, accessToken, accessSecret)

auth\_detail\_request = CompanyAuthDetailRequest()

auth\_detail\_request.set\_requestId('69cca0f6-a247-4d64-8de3-aac343690a25')

auth\_detail\_response = sdkClient.request(auth\_detail\_request)

print(auth\_detail\_response)

### 回调说明

**描述：**接收回调请求的地址接口需要支持Content-Type: application/x-www-form-urlencoded 的post请求，每次回调不通会重复调用一次，如果调用不通，1分钟后再调用2次，一共调用4次。

**参数说明：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| status | Integer | 认证结果状态：1（“认证成功”）、2（“认证失败”） |
| requestId | String | 认证请求Id |
| authInfo | CorpAuthDetail | 认证详情 |

CorpAuthDetail：

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| name | String | 认证公司名称 |
| registerNo | String | 认证公司工商注册号 |
| legalPerson | String | 认证公司法人代表姓名 |
| applicantName | String | 认证记录申请人姓名 |
| applicantPhone | String | 认证记录申请人联系方式 |

接收回调请求接口的返回值： 接口返回值必须包含以下参数，且转换成json字符串返回。

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| code | Int | 接收回调消息 响应码 |

返回值说明：

code为0表示接收回调消息成功： {"code":0} code不为0表示接收回调消息失败： {"code":1001}

## 全局响应码

| **响应码** | **描述** |
| --- | --- |
| 0 | SUCCESS，接口调用成功 |
| 1001 | ERROR，服务器内部错误 |
| 1002 | PERMISSION DENIED，没有接口调用权限 |
| 1005 | INVALID PARAM，无效的参数 |
| 1601 | INSUFFICIENT BALANCE，余额不足 |