

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

福昕 PDF Services API 快速上手指南

快速上手

福昕PDF Services API支持通过HTTP客户端调用基于云的REST API来处理PDF。以下是获取开发者凭据并使用API的步骤（更多示例详见[GitHub](#)）：

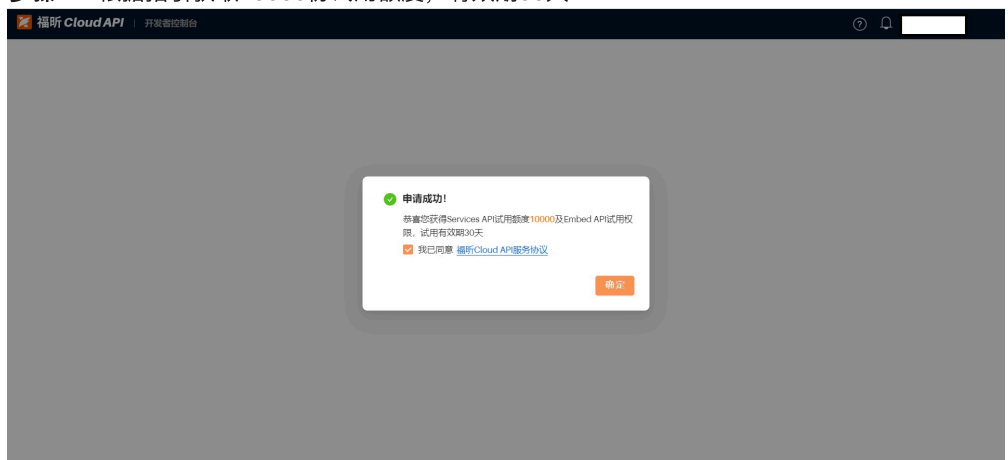
获取凭证

您需要一个客户端ID和密钥ID来使用福昕PDF Services API，具体步骤如下：

步骤1： 注册福昕账号并登录 福昕开发者控制台。



步骤2： 根据指引获取10000份试用额度，有效期30天



步骤3： 创建您的工程。

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

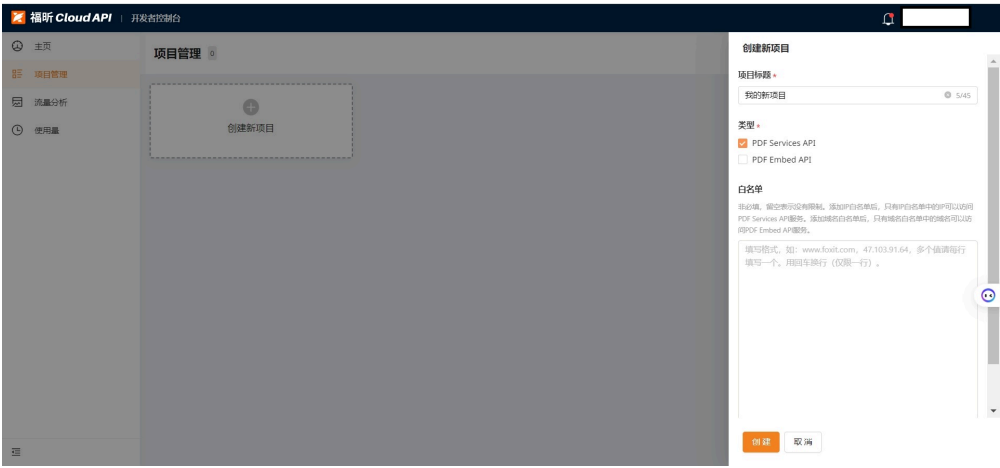
低代码平台

简介

概念介绍

快速使用指南

HTTP触发器



步骤4：拷贝客户端ID和密钥ID作为凭证。



调用APIs

福昕PDF Services API是通过使用福昕提供的凭据进行身份验证的REST API。在获得凭据后，您可以按照以下说明调用REST API。每个成功的文档处理需要进行以下三个REST API调用：

1. 调用文档处理API，例如：`/document/create`
2. 调用 `/task` API 并使用步骤1返回的任务ID(taskId)获取任务执行状态
3. 调用 `/download` API 并使用步骤2返回的文档ID(docId)下载生成的文件（PDF文档，图片，文本文件，压缩包等）。

备注：每个接口参数sn的生成参考签名(sn)生成算法。

1. 创建新任务

调用 `/document/create` (创建PDF文件) 上传输入文件并创建一个新的任务。

例如: POST `https://servicesapi.foxitsoftware.cn/api/document/create`

`/document/create` 允许用户通过微软Office文档(Word, Excel and PowerPoint)和其他格式的文件来创建PDF文件。

- 请求参数

查询参数显示在以下表格中，并附加到URL并传递到服务器。

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

参数	类型	是否必须项	描述
clientId	字符串	必须	客户端ID，开发者可以通过福昕开发者控制台生成。
sn	字符串	必须	参数的签名。

表单数据参数如下表所示，附加到HTTP主体并传递给服务器。

参数	类型	是否必须项	描述
inputDocument	文件	必须	需要被转换为PDF的文档
format	字符串	必须	被转换的文档格式。可选值为：word, excel, ppt, image, text.

返回值

/task 允许用户通过返回的 taskId 获取任务状态。如果任务已完成，我们可以获取 docId。

返回值	类型	描述
code	整型	错误值
msg	字符串	错误信息
data	对象	详细数据对象
taskInfo	对象	任务信息
taskId	字符串	任务ID
details	对象	详细错误信息
checkParams	数组	参数错误信息

例子

复制代码

```
<table>
  <tr>
    <th>Example Request</th>
    <td>
      <pre>
        <code>
          $curl -X POST --header 'Content-Type: multipart/form
        </code>
      </pre>
    </td>
  </tr>
  <tr>
    <th>Example Response</th>
    <td>
      <pre>
        <code>Success:
      </pre>
    </td>
  </tr>
</table>
```

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

福昕 Cloud API

```
{
  "code": 0,
  "msg": "Success",
  "data" {
    "taskInfo":
    {
      "taskId": "613881e8ba5e2a000152d520"
    }
  }
}
Failed:
{
  "code": 600000,
  "msg": "Error: Missing parameter",
  "data": []
}
```

```
</pre>
</td>
</tr>
</table>
```

2. 等待任务完成

GET <https://servicesapi.foxitsoftware.cn/api/task> 复制代码

/task 允许用户通过返回的 taskId 获取任务状态。如果任务已完成，我们可以获取 docId。

- 请求参数

参数	类型	是否必须项	描述
clientId	字符串	必须	客户端ID，开发者可以通过福昕开发者控制台生成。
sn	字符串	必须	参数的签名。
taskId	字符串	必须	新任务的任务ID。

返回值

返回值	类型	描述
code	整型	返回值
msg	字符串	返回信息
data	对象	Json类型数据
taskInfo	对象	任务信息
docid	字符串	输出文档ID
percentage	整型	任务执行的百分比
detail	字符串	详细错误信息

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

例子

复制代码

```
<table>
  <tr>
    <th>Example Request</th>
    <td>
      <pre>
        <code>
          $curl -X GET --header 'Accept: application/json' 'ht
        </code>
      </pre>
    </td>
  </tr>
  <tr>
    <th>Example Response</th>
    <td>
      <pre>
        <code>Success:
          {
            "code": 0,
            "msg": "Success",
            "data": {
              "taskInfo": {
                "percentage": 100,
                "docid": "6138a1f00afc7f000128cb01"
              }
            }
          }
        Failed:
          {
            "code": 600060,
            "msg": "Error: Server internal error",
            "data": {
              "detail": "10000|GENERIC_SERVER_FAILURE, more info:
            }
          }
        </code>
      </pre>
    </td>
  </tr>
</table>
```

3. 下载输出文件

下载输出文件。*/download* API 通过*/task* API 返回的文档ID作为输入参数来下载输出文件。GET <https://servicesapi.foxitsoftware.cn/api/download>

- 请求参数

参数	类型	是否必须项	描述
clientId	string	必须	客户端ID，开发者可以通过福昕开发者控制台生成。
sn	string	必须	参数的签名。
docId	string	必须	task接口返回的文档ID

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

返回值

输出文件的数据

Node.js代码实例

确保您已安装版本大于或等于12.0的Node.js 。 您可以使用任何您喜欢的HTTP客户端库， 这里我们使用流行的 "axios" 作为示例。

步骤1： 安装依赖

```
npm install axios form-data
```

复制代码

步骤2： 新建一个文件: test-api.js

```
const FormData = require('form-data');
const axios = require('axios');
const fs = require('fs');
const path = require('path');

// Get clientId and secretId form the json file.
const credentials_params = require('../foxit_cloud_api_credentials.json')
const clientId = credentials_params.client_credentials.client_id;
const secretId = credentials_params.client_credentials.secret_id;

// The signature of parameters will be calculated in the actual interface
const sn = 'testsn'

// TODO: replace with your own input doc path and output file path
const inputFilePath = '../input_files/AboutFoxit.doc'
const outputFilePath = '../output_files/create_pdf_from_others/AboutFoxi

// create axios instance and setup common request config
const request = axios.create({
  baseURL: 'https://servicesapi.foxitsoftware.cn/api/',
  timeout: 60 * 1000,

  // common params for every api call
  params: {
    sn,
    clientId
  },
});

//format can be "word", "excel", "ppt", "image", "text".
function createPDFTask(inputFilePath, format = 'word'){
  const querystring = require('querystring');
  const crypto = require('crypto');
  let queryParams = {
    'clientId': clientId,
    'format': format,
  };

  const sortedParams = Object.fromEntries(Object.entries(queryParams).so
  // Stringify the parameters
  const querystringified = querystring.stringify(sortedParams);
```

复制代码

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

```

const queryStringWithSecret = querystringified + '&sk=' + secretId;
// Generate signature using md5
request.defaults.params.sn = crypto.createHash('md5').update(queryStri
const readStream = fs.createReadStream(inputFilePath)
readStream.on('error',function (err) {
    console.log('read input file error');
    console.log(err.message)
    process.exit(1)
})

const formData = new FormData()
formData.append('inputDocument', readStream)
formData.append('format', format)
//Upload a file and create a new workflow task.
return request({
    method: 'post',
    url: '/document/create',
    headers: formData.getHeaders(),
    data: formData
}).then(function (res) {

    const resultData = res.data

    // Read the api doc about all result codes
    if(resultData.code === 0){
        return resultData.data.taskInfo.taskId
    }
}).catch(function (err) {
    console.log("Create pdf from others task error:", err.response.data)
    throw err
})
}

function getTaskInfo(taskId){
    const querystring = require('querystring');
    const crypto = require('crypto');
    let queryParams = {
        'clientId': clientId,
        'taskId': taskId,
    };

    const sortedParams = Object.fromEntries(Object.entries(queryParams).so
    // Stringify the parameters
    const querystringified = querystring.stringify(sortedParams);

    const queryStringWithSecret = querystringified + '&sk=' + secretId;
    // Generate signature using md5
    const sn = crypto.createHash('md5').update(queryStringWithSecret).dige

    return request({
        method: 'get',
        url: '/task',
        params: {sn, taskId}
    }).then(function (res) {
        const resultData = res.data
        if(resultData.code === 0){
            const taskInfo = resultData.data.taskInfo
            console.log("Task process is:", taskInfo.percentage)
            return taskInfo
        }
    })
}

```

Embed API

快速指南

开发者手册

Services API

快速指南

签名(sn)生成算法

API手册

低代码平台

简介

概念介绍

快速使用指南

HTTP触发器

```

    }else{
      throw new Error(resultData.code + resultData.message)
    }
  }).catch(function (err) {
    throw err
  })
}

function pollForDocId(taskId, intervalInMilliseconds = 2000){
  return new Promise(function(resolve, reject){
    // poll for task status and once task is completed resolve the promi
    let timeout
    function poll(){
      if(timeout){clearTimeout(timeout)}
      getTaskInfo(taskId).then(function(taskInfo){
        if(taskInfo.percentage === 100){
          console.log("Task completed.")
          resolve(taskInfo.docId)
        }else{
          setTimeout(poll, intervalInMilliseconds)
        }
      }).catch(function(err){
        // when task is running, the task api will return error
        // if task is running, try to get taskInfo later
        if(err.response.data.data.detail.indexOf('The task is running')
          console.log("Task is running, retry in ", intervalInMilliSecon
          setTimeout(poll, intervalInMilliseconds)
        }
      })
    }
    poll()
  })
}

function downloadFileByDocId(docId, outputFilePath){

  const writeStream = fs.createWriteStream(outputFilePath)

  const querystring = require('querystring');
  const crypto = require('crypto');
  let queryParams = {
    'clientId': clientId,
    'docId': docId,
  };

  const sortedParams = Object.fromEntries(Object.entries(queryParams).so
  // Stringify the parameters
  const querystringified = querystring.stringify(sortedParams);

  const queryStringWithSecret = querystringified + '&sk=' + secretId;
  // Generate signature using md5
  const sn = crypto.createHash('md5').update(queryStringWithSecret).dige

  writeStream.on('error',function (err) {
    console.log(err.message)
    process.exit(1)
  })
  // Download the convert streams.
  request({

```


Embed API

- 快速指南
- 开发者手册

Services API

- 快速指南
 - 签名(sn)生成算法
 - API手册
- 低代码平台
- 简介
 - 概念介绍
 - 快速使用指南
 - HTTP触发器

```
method: 'get',
url: '/download',
responseType: 'stream',
params: {sn, docId}
}).then(function (res) {
  res.data.pipe(writeStream)
}).catch(function (err) {
  console.log(err);
})
// Save the convert file.
return new Promise(function(resolve, reject){
  writeStream.on('finish', function(){
    console.log('Download stream finished');
    resolve()
  })
  writeStream.on('error',function (err) {
    console.log(err.message)
    reject(err)
  })
})
}

async function start(){
  outputPath = path.dirname(outputFilePath);
  if(!fs.existsSync(outputPath)){
    fs.mkdirSync(outputPath);
  }
  const taskId = await createPDFTask(inputFilePath)
  const docId = await pollForDocId(taskId)
  await downloadFileByDocId(docId, outputFilePath)
}

start().then(function(){
  console.log("Create PDF successfully!");
  process.exit(0)
}).catch(function(err){
  console.log("Create PDF failed: " + err);
  process.exit(1)
})
})
```

步骤3：执行 test-api.js

```
node test-api.js
```

复制代码



福昕在亚洲、美洲、欧洲和澳洲设有多家子公司，福昕直接用户已超过7.5亿，企业用户数达42.5万以上，遍布全球。

产品	应用场景	文档	支持
福昕 PDF SDK	数据安全	文档主页	联系我们
	文件处理		福昕软件服务协议
	法律实践		福昕Cloud API服务协议



QQ在线客服 (2852373735)
周一至周五9:00-18:00

点击进入



销售咨询
SALES CONSULTANT

010-50951668