# webSdk接口说明1.2

部分接口参数采用js的对象来表示，具体调用方式可以参考demo的使用

### 基本地址

> 地址：http://127.0.0.1:18999/WebScan

### 获取当前连接设备id

> 方法名：**/getVersionInfo**
>
> - 参数：**licence(授权码) pid (设备id，可以为空)**
> - 类型：**POST**
> - 返回值：**{ “code”: 200, “msg”: null, “data”: “设备id”}**

### 获取连接的设备名

> 方法名： **/getDevices**
>
> - 参数：**pid (设备id，可以为空)**
> - 类型：**GET**
> - 返回值：**{ “code”: 200, “msg”: null, “data”: [ “设备名” ]}**

### 获取扫描参数以及配置信息

> 方法名： **/getParams**
>
> - 参数：**pid (设备id)**
> - 类型：**GET**
> - 返回值：
>
> {
> “code”: 200,
> “msg”: null,(错误信息)
> “data”: {
> “device”: “Scanner”,//设备名称
> “autofeeder”: true,//自动进纸
> “pixel”: 1,//扫描模式 0：黑白 1：灰度 2：彩色
> “white”: false,//丢弃空白页
> ‘discardBlankThre’: 5,//跳过空白页阈值 1-100 默认值5

“single”: false,//单页扫描
“format”: “jpg”,//输出格式 jpg png bmp tiff pdf ofd
“resolution”: 200,//扫描分辨率 范围 100-600 默认值200
“orentation”:0,//图片旋转 0：原图 90：度 180：旋转180度 270：旋转270度
“paperType”:”Auto”,//扫描幅面 A3：A3幅面 Auto:自适应幅面 A4：A4幅面 A4R:A4横向
“‘splitImage”: 0,//图像分割 0：disable 1：垂直分割 2： 水平分割
“noiseDetachEnable”: true,//去除噪点
“noiseDetach”: 15,//噪点阈值 范围： 10-50 默认值15
“upload”: {//上传配置
“uploadMode”: 2,//是否边扫边上传 0：http 1：ftp 2: 不上传
“httpUrl”: null,//上传地址
“fileName”: null,//接收文件参数名
“httpMethod”: null,//上传方式 GET POST PUT
“header”: null,//请求头
“param”: null,//参数 json格式
“ftpUrl”: null,//ftp 地址
“ftpPath”: “/images”,//路径
“ftpUser”: null,//ftp 用户名
“ftpPassword”: null,//ftp 密码
“ftpPort”: 21,//端口号
“ftpMode”: 2//连接模式 1：主动 2：被动
}
}
}

## 设置扫描参数

方法名：**/setParams**

**参数**

- pid（设备id）
- params（
  {
  “device”: “Scanner”,//设备名称
  “autofeeder”: true,//自动进纸
  “pixel”: 1,//扫描模式 0：黑白 1：灰度 2：彩色
  “white”: false,//丢弃空白页
  ‘discardBlankThre’: 5,//跳过空白页阈值 1-100 默认值5
  “single”: false,//单页扫描
  “format”: “jpg”,//输出格式 jpg png bmp tiff pdf ofd
  “resolution”: 200,//扫描分辨率 范围 100-600 默认值200
  “orentation”:0,//图片旋转 0：原图 90：度 180：旋转180度 270：旋转270度

“paperType”:”Auto”,//扫描幅面 A3：A3幅面 Auto:自适应幅面 A4：A4幅面 A4R:A4横向
“‘splitImage”: 0,//图像分割 0：disable 1：垂直分割 2： 水平分割
“noiseDetachEnable”: true,//去除噪点
“noiseDetach”: 15,//噪点阈值 范围： 10-50 默认值15
“upload”: {//上传配置
“uploadMode”: 2,//是否边扫边上传 0：http 1：ftp 2: 不上传
“httpUrl”: null,//上传地址
“fileName”: null,//接收文件参数名
“httpMethod”: null,//上传方式 GET POST PUT
“header”: null,//请求头
“param”: null,//参数 json格式
“ftpUrl”: null,//ftp 地址
“ftpPath”: “/images”,//路径
“ftpUser”: null,//ftp 用户名
“ftpPassword”: null,//ftp 密码
“ftpPort”: 21,//端口号
“ftpMode”: 2//连接模式 1：主动 2：被动
}
}
）

- 类型：**POST**
- 返回值：**{ “code”: 200, “msg”: “设置成功”, “data”: “”}**

## 导出ofd

方法名：**/majorOfd**

- 类型：**POST**
- 参数 {“formDataString”:{
  “isAuto”: isAuto,
  “pid”: this.clientId
  }}
- isAuto(true:文档中的每一页都是图片原始尺寸 false:按照A4大小尺寸缩放)
- pid(设备id)
- 示例：
- var formData = {“isAuto”: isAuto, “pid”: this.clientId }
  jQuery.ajax({
  type: ‘post’,
  url: this.url + “/majorOfd”,
  data: {
  “formDataString”: JSON.stringify(formData)
  },
  success: function (data) {

callback(data);
},
error: function () {
var result = new Result(500, “网络错误”, null)
callback(result);
}
})

- 返回值：**{ “code”: 200, “msg”: null, “data”: “base64文件”}**

### 导出ofd（以流的形式返回）

方法名：**/majorOfdFile**

- 类型：**GET**
- 参数
  “isAuto”: true,
  “pid”: “clientId”
- isAuto(true:文档中的每一页都是图片原始尺寸 false:按照A4大小尺寸缩放)
- pid(设备id)
- 示例：
- var xhr = new XMLHttpRequest();
  xhr.open(‘GET’, this.url + “/majorOfdFile?pid=clientId&isAuto=true”, true);
  xhr.responseType = “blob”; // 返回类型blob
  xhr.onload = function () {
  callback(this);
  }
  // 发送ajax请求
  xhr.send()
- **

### 导出PDF

方法名：**/majorPdf**

- 类型：**POST**
- 参数：{
  “formDataString”: {
  “pid”: this.clientId,
  }
  }
- 示例：
- var formData = {“pid”: this.clientId}
  jQuery.ajax({

type: ‘post’,
url: this.url + “/majorPdf”,
data: {
“formDataString”: JSON.stringify(formData)
},
success: function (data) {
callback(data);
},
error: function () {
var result = new Result(500, “网络错误”, null);
callback(result);
}
})

- 返回值：**{ “code”: 200, “msg”: null, “data”: “base64文件”}**

### 导出PDF（以流的形式返回）

方法名：**/majorPdfFile**

- 类型：**GET**
- 参数：
  “pid”: “clientId”
- 示例：
- var xhr = new XMLHttpRequest();
  xhr.open(‘GET’, this.url + “/majorPdfFile?pid=clientId”, true);
  xhr.responseType = “blob”; // 返回类型blob
  xhr.onload = function () {
  callback(this);
  }
  // 发送ajax请求
  xhr.send()
  **

### 导出Tiff

方法名：**/majorTiff**

- 类型：**POST**
- 参数：
  “pid”: “clientId”
- 示例：
- jQuery.ajax({
  type: ‘post’,

url: this.url + “/majorTiff”,
cache: false,
data: {
“pid”: this.clientId
},
success: function (data) {
callback(data);
},
error: function () {
result = new Result(500, “网络错误”, null);
callback(result);
}
})

- 返回值：**{ “code”: 200, “msg”: null, “data”: “base64文件”}**

### 导出Tiff（以流的形式返回）

方法名：**/majorTiffFile**

- 类型：**GET**
- 参数：
  “pid”: “clientId”
- 示例：
- var xhr = new XMLHttpRequest();
  xhr.open(‘GET’, this.url + “/majorTiffFile?pid=clientId”, true);
  xhr.responseType = “blob”; // 返回类型blob
  xhr.onload = function () {
  callback(this);
  }
  // 发送ajax请求
  xhr.send()

### 导出为Zip压缩包

方法名：**/downLoadZip**

- 类型：**POST**
- 参数：
  “pid”: this.clientId
- 示例：
- jQuery.ajax({
  type: ‘post’,
  url: this.url + “/downLoadZip”,

cache: false,
data: {
“pid”: this.clientId
},
success: function (data) {
callback(data);
},
error: function () {
result = new Result(500, “网络错误”, null);
callback(result);
}
})

- 返回值：**{ “code”: 200, “msg”: null, “data”: “base64文件”}**

### 导出为Zip压缩包（以流的形式返回）

方法名：**/downLoadZipFile**

- 类型：**GET**
- 参数：
  “pid”: this.clientId
- 示例：
- var xhr = new XMLHttpRequest();
  xhr.open(‘GET’, this.url + “/downLoadZipFile?pid=clientId”, true);
  xhr.responseType = “blob”; // 返回类型blob
  xhr.onload = function () {}
  // 发送ajax请求
  xhr.send()
- **

### 图片上传

方法名：**/uploadImage**

- 参数：
- {
  “formDataString”: {
  id:pid,//设备id
  uploadMode: 0,//上传模式 0：http 1：ftp 0:无
  httpUrl: ”,//上传地址
  fileName: ”,//接收文件参数名
  httpMethod: ”,//上传方式 POST GET PUT
  header: ”,//请求头

param: ”,//参数 josn类型字符串
ftpUrl: ”,//ftp 地址
ftpPath: ‘/images’,//路径
ftpUser: ”,//ftp用户名
ftpPassword: ”,//ftp密码
ftpPort: 21,//端口号
ftpMode: 2,//连接模式 1 主动 2 被动
format: 2 //上传格式 0: ofd 1: pdf 2: zip
}
}

- 类型：**POST**
- 返回值：**{“code”:200,”msg”:null,”data”:”上传成功”}**

### 保存图片

方法名：**/image/saveImage**

- 参数：**pid (设备id) imageName(图片名称/路径) image(修改后图片的base64)**
- 类型：**POST**
- ◦ 示例：
- jQuery.ajax({
  type: ‘post’,
  url: this.url + ‘/image/saveImage’,
  data: {
  “pid”: this.clientId,
  “imageName”: imageName,
  “image”: base64
  }, success: function (data) {
  callback(data);
  },
  error: function () {
  var result = new Result(500, “网络错误”, null);
  callback(result);
  }
  })
- 返回值：**{ “code”: 200, “msg”: “”, “data”: “保存成功”}**

### 获取上次扫描结果

- 方法名：**/image/getImageByPid**
- 参数：**pid（设备id）**
- 类型：**GET**

- 返回值：**{“code”: 200,”msg”: null,”data”: [ {“imageName”: “图片名/路径”,”src”: “图片base64”}]}**

### 删除图片

> 方法名：**/image/deleteImage**
>
> - 参数：**pid（设备id） imageName（图片名称/路径）**
> - 类型：**POST**
> - 返回值：
>   **{“code”:200,”msg”:null,”data”:”删除成功”}**

### 删除全部图片

> 方法名：**/image/deleteAllImage**
>
> - 参数：**pid(设备id)**
> - 类型：**POST**
> - 返回值：
>   **{“code”:200,”msg”:null,”data”:”删除成功”}**

### 获取最后批次

> 方法名：**/getLastBetch**
>
> - 类型：**GET**
> - 返回值：**{“code”:200,”msg”:null,”data”:”设备id”}**

### 获取序列号

> 方法名：**/getSerialNumber**
>
> - 参数：**device（设备id）**
> - 类型：**GET**
> - 返回值：
> - **{“code”: 200, “msg”: null, “data”:”序列号”}**

### 根据图片名称获取图片

> 方法名：**/getImageByName**
>
> - 参数：**pid（设备id） imageName（图片名称/路径）**

- 类型：**GET**
- 返回值：
- **{“code”: 200, “msg”: null, “data”:”图片base64”}**

### 重置扫描index

方法名：**/image/resetPatchIndex**

- 类型：**POST**
- 返回值：
- **{“code”: 200, “msg”: null, “data”:”success”}**

### 分割图像

方法名：**/image/split**

- 参数：**pid（设备id）isHorizontal（是否是垂直切割 true:垂直拆分（图像将分为左右两部分 false:水平拆分（图像将分为上下两部分））） imageName（图片名称/路径）x1:开始分割的坐标点x y1:开始分割时的坐标点y x2:结束位置的坐标点x y2:结束位置的坐标点y**
- 类型：**POST**
- 示例：
- jQuery.ajax({
  type: ‘post’,
  url: this.url + “/image/split”,
  cache: false,
  data: {
  “pid”: this.clientId,
  “imageName”: imageName,
  “isHorizontal”: isHorizontal,
  “x1”: parseInt(x1),
  “y1”: parseInt(y1),
  “x2”: parseInt(x2),
  “y2”: parseInt(y2)
  }
  })
- 返回值：
- **{“code”: 200, “msg”: “切分成功”, “data”:{“oneSrc”:”拆分后的第一张base64图片”,”oneName”:”拆分后的第一张图片路径/名称”,”twoSrc”:”拆分后的第二张base64图片”,”twoName”:”拆分后的第二张图片路径/名称”}}**
- 说明：根据两个坐标点按照直线对图像进行拆分，拆分分为水平拆分和垂直拆分

### 合并图像

方法名：**/image/mergeHorizontal**

- 参数：**pid（设备id）isHorizontal（是否水平合并 true:水平合并（将多个图像按顺序水平合并为一张) false:垂直合并（将多个图像按顺序垂直合并为一张））） indexs（要合并图像的index值集合）**
- 类型：**POST**
- -示例：
- jQuery.ajax({
  type: ‘post’,
  url: this.url + “/image/mergeHorizontal”,
  cache: false,
  data: {
  “pid”: this.clientId,
  “isHorizontal”: isHorizontal,
  “indexs”: indexs
  }
  })
- 返回值：
- **{“code”: 200, “msg”: “合并完成”, “data”:{“imageName”:”合并后的图像路径/名称”,”src”:”base64图片”}**
- 说明：将多张图像进行合并，合并包含水平合并和垂直合并。

### 图片顺序交换

方法名：**/image/exchangeImage**

- 参数：**pid（设备id）indexs（要交换的两个图像index集合）**
- 类型：**POST**
- 返回值：
- **{“code”: 200, “msg”: null, “data”:”交换完成！”}**

### 书籍排序

方法名：**/image/bookSort**

- 参数：**pid（设备id）**
- 说明：将扫描后的图片进行一次书籍排序，此操作不可逆！
- 类型：**POST**
- 返回值：
- **{“code”: 200, “msg”: “书籍排序完成！”, “data”:书籍排序后的路径名/图片名集合}**

# socket通信部分

### 基本地址

> 地址：http://127.0.0.1:28999/

### 开始扫描

- 方法名：**scan**

### 停止扫描

- 方法名：**stop**

### 扫描仪回调：

- 名称：**connect_error**
  初始化连接服务失败
- 名称：**event**
  扫描仪状态事件
- 名称：**success**
  扫描仪连接成功
- 名称： **error**
  扫描仪错误事件
- 名称：**image**
  扫描仪图片回调
- 名称：**result**
  扫描结束