一、接口简介

1. 通信基于 websocket 协议，端口默认情况下是 38999，可以通过配置文件修改。

2. 消息为 json 字符串格式，分为请求消息、响应消息和事件消息，通过 func 字段标识具体的功能。

3. 请求消息可以携带的字段除了 func 外，还可以有 iden 和该请求特有的字段。iden 可以用于标识该请求消息。

4. 响应消息可以携带的字段除了 func 外，还必须有 ret 字段（表示返回值），ret 不为 0 时必须有 err_info 字段（表示错误信息），可以有 iden 和该响应特有的字段。iden 与对应的请求消息的 iden 相同。

5. 事件消息可以携带的字段除了 func 外，还可以有 iden 和该事件特有的字段。iden 与对应的请求消息的 iden 相同。

二、基本接口

1. 设置全局配置

请求：

```
{
    “func”:”set_global_config”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    // 文件保存
    “file_save_path”:”C:\\”,  // 默认是文档目录
    “file_name_prefix”:””,    // 文件名前缀
    “file_name_mode”:”date_time”,  // 可以是 date_time、random

    // date_time: 图片直接以时间命名保存；
    // random：图片以随机数值命名保存；

    // 图像保存
     “image_format”:”jpg”,        // 格式，支持 jpg、bmp、png、tif、pdf、ofd、ocr-pdf 和 ocr-ofd
    “image_jpeg_quality”:80,// jpg 质量，0-100
    “image_tiff_compression”:”none”,     // 可以为 none、 lzw 和 jpeg
    “image_tiff_jpeg_quality”:80          // tiff-jpeg 质量，0-100
}
```

响应：

```
{
    “func”:”set_global_config”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

2. 获取全局配置

请求：

```
{
    “func”:”get_global_config”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”get_global_config”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    // 文件保存
    “file_save_path”:”C:\\”,   // 默认是文档目录
    “file_name_prefix”:””,     // 文件名前缀
    “file_name_mode”:”date_time”,   // 可以是 date_time、random
    // 图像保存
     “image_format”:”jpg”,          // 格式，支持 jpg、bmp、png、tif、pdf、ofd、ocr-pdf 和 ocr-ofd
    “image_jpeg_quality”:80,// jpg 质量，0-100
    “image_tiff_compression”:”lzw”,        // 可以为 none、 lzw 和 jpeg
    “image_tiff_jpeg_quality”:80        // tiff-jpeg 质量，0-100
}
```

3. 加载本地图像

请求：

```
{
    “func”:”load_local_image”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “image_path”:”C:\\1.jpg”       // 图片本地路径
}
```

响应：

```
{
    “func”:”load_local_image”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    “image_base64”:”xxx”     // base64 字符串，带前缀
}
```

4. 保存本地图像

请求：

```
{
    “func”:”save_local_image”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “image_base64”:”xxx”    // base64 字符串，带前缀
}
```

响应：

```
{
    “func”:”save_local_image”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    “image_path”:”C:\\1.jpg”
}
```

5. 删除本地文件（为了安全，仅能删除后端生成的文件）

请求：

```
{
    “func”:”delete_local_file”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “file_path”:”C:\\1.jpg”         // 本地路径
}
```

响应：

```
{
    “func”:”delete_local_file”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

6. 清理全局文件保存目录（为了安全，只会删除后端生成的文件）

请求：

```
{
    “func”:”clear_global_file_save_path”,
    “iden”:”56fc”           // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”clear_global_file_save_path”,
    “iden”:”56fc”,
    “ret”:0,
```

```
    “err_info”:””
}
```

7. 上传本地文件

请求：

```
{
    “func”:”upload_local_file”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “file_path”:”C:\\1.jpg”,
    “remote_file_path”:”/path/1.jpg”,      // 远程文件路径
    “upload_mode”:”http”,   // http 或 ftp
    “http_host”:”192.168.1.100”,
    “http_port”:80,
    “http_path”:”/upload.php”,
    “ftp_user”:”xxx”,
    “ftp_password”:”xxx”,
    “ftp_host”:”192.168.1.100”,
    “ftp_port”:21
}
```

响应：

```
{
    “func”:”upload_local_file”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

三、图像采集接口

1. 设备初始化

请求：

```
{
    “func”:”init_device”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”init_device”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

事件：

```
{
    “func”:”device_arrive”,
    “iden”:”56fc”,
    “device_name”:”scanner”
}
```

事件：

```
{
    “func”:”device_remove”,
    “iden”:”56fc”,
    “device_name”:”scanner”
}
```

2. 设备反初始化（需初始化后调用）

请求：

```
{
    “func”:”deinit_device”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”deinit_device”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

3. 获取设备是否已初始化

请求：

```
{
    “func”:”is_device_init”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”is_device_init”,
    “iden”:”56fc”,
    “ret”:0, // 0 表示已初始化，非 0 表示未初始化
    “err_info”:””
}
```

4. 获取设备列表（需初始化后调用）

请求：

```
{
    “func”:”get_device_name_list”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”get_device_name_list”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    “device_name_list”:[“scanne_1”, “scanne_2”]
}
```

5. 打开设备（需初始化后调用，每次只能打开一台设备）

请求：

```
{
    “func”:”open_device”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “device_name”:”scanner”，为空表示第一台设备
}
```

响应：

```
{
    “func”:”open_device”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

6. 关闭设备（需打开设备后调用）

请求：

```
{
    “func”:”close_device”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”close_device”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
```

}

7. 设置设备参数（需打开设备后和未开始扫描时调用，设置的时候有些参数可能成功，有些参数可能失败，有一个参数设置失败时就会返回非 0，并设置相关的错误信息）

请求：

```
{
    “func”:”set_device_param”,
    “iden”:”56fc”,          // 可以是随机字符串，该字段可以省略
    “device_param”:
    [
        {
            “name”:”xxx”,
            “value”:”xxx”
        },
        {
            “name”:”xxx”,
            “value”:0
        },
        {
            “name”:”xxx”,
            “value”:0.0000
        },
        {
            “name”:”xxx”,
            “value”:true
        }
    ]
}
```

响应：

```
{
    “func”:”set_device_param”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

8. 获取设备参数（需打开设备后调用）

请求：

```
{
    “func”:”get_device_param”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
```

}

响应：

```
{
    “func”:”get_device_param”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    “device_param”:
    [
        {
            “group_name”:”xxx”,
            “group_param”:
            [
                {
                    “name”:”xxx”,
                    “value_type”:”string”,
                    “value”:”xxx”,
                    “range_type”:”list”,
                    “value_list”:[“xxx”, “xxx”]
                },
                {
                    “name”:”xxx”,
                    “value_type”:”int”,
                    “value”:100,
                    “range_type”:”list”,
                    “value_list”:[100, 200]
                },
                {
                    “name”:”xxx”,
                    “value_type”:”int”,
                    “value”:20,
                    “range_type”:”min_max”,
                    “value_min”:10,
                    “value_max”:30
                },
                {
                    “name”:”xxx”,
                    “value_type”:”bool”,
                    “value”:true
                }

            ]
        },
        {
```

```
            // 其他组
        }
    ]
}
```

9. 重置设备参数（需打开设备后和未开始扫描时调用）

请求：

```
{
    “func”:”reset_device_param”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”reset_device_param”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
```

10. 获取当前设备名称（需打开设备后调用）

请求：

```
{
    “func”:”get_curr_device_name”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”get_curr_device_name”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””,
    “device_name”:”scanner”
}
```

11. 开始扫描（需打开设备后调用）

请求：

```
{
    “func”:”start_scan”,
    “iden”:”56fc”,         // 可以是随机字符串，该字段可以省略
    “local_save”:true,     // 是否保存到本地，默认保存
    “get_base64”:false     // 是否获取 base64，默认不获取
```

```
}
响应：
{
    “func”:”start_scan”,
    “iden”:”56fc”,
    “ret”:0,
    “err_info”:””
}
事件：
{
    “func”:”scan_begin”,
    “iden”:”56fc”
}
事件：
{
    “func”:”scan_end”,
    “iden”:”56fc”
}
事件：
{
    “func”:”scan_info”,
    “iden”:”56fc”,
    “is_error”:false,            // 是否是错误消息
    “info”:”start scanning”    // 消息内容
}
事件：
{
    “func”:”scan_image”,
    “iden”:”56fc”,
    “image_path”:”C:\\1.jpg”,     // 扫描图片的本地路径
    “image_base64”:”xxx”    // base64 字符串，带前缀
}
```

12. 停止扫描（需开始扫描后调用）

```
请求：
{
    “func”:”stop_scan”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
响应：
{
    “func”:”stop_scan”,
    “iden”:”56fc”,
```

```
    “ret”:0,
    “err_info”:””
}
```

13. 获取设备是否正在扫描

请求：

```
{
    “func”:”is_device_scanning”,
    “iden”:”56fc”          // 可以是随机字符串，该字段可以省略
}
```

响应：

```
{
    “func”:”is_device_scanning”,
    “iden”:”56fc”,
    “ret”:0, // 0 表示正在扫描，非 0 表示未进行扫描
    “err_info”:””
}
```