### Raysync JS-SDK API 文档

直接引入raysync.min.js然后即可new Raysync({webServer:’http://127.0.0.1:8090/api’});

Raysync接收一个webServer参数：部署Raysync 服务的地址 xxxx.xxxx.xxxx/api

**关于上传下载的使用**

在成功建立客户端连接后，需要对传输系统做目录切换定位操作，即调用getFileList(**url**)，

此时**url** 即为后续上传文件的目标位置，在重新调用getFileList之前上传的文件都会在该目录下面，比如：

Raysync.getFileList(‘/raysync/test’) ;此时会返回结果中的list目录为/raysync/test 目录下的文件列表，

此时调用upload上传文件会全部传至这个路径下面，如：

Raysync.upload(type，path) //type为上传类型：0：文件；1：文件夹

//

此时调用下载download接口对这个目录下的所有文件进行下载操作，如：

Raysync.download([

‘/raysync/test/a.html’,

‘/raysync/test/b.html’,

])

此时就可以下载到该目录下的a.html和b.html，如需下载其他路径下的文件，需调用getFileList(**url**)切换至目标位置，然后调用下载即可

### 接口调用：

1. *l****ogin***登陆命令，传输流程必须先调用登陆

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| account | string | 账号 | Y |
| password | string | 密码 | Y |

返回值：

|  |  |
| --- | --- |
| type | ‘login’ |
| result | 0:成功；非0:失败 |
| data | userId: 用户id |
| account:账号 |
| client\_version:服务端版本号 |
| isActived: license 是否激活；0未激活1已激活 |
| message | 结果描述 |

简单示例：

Raysync.login(accout,password)

.then(function(res){

}

).catch(function(res){

})

1. ***CreateWebSocket*** 建立和传输客户端的连接

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| account | string | 账号 | Y |
| password | string | 密码 | Y |

Raysync.CreateWebSocket (accout,password) //账号：account,密码：password。必须

Raysync.obserable.on(‘webSocket’,function(){ //注册事件回调监听是否连接成功

})

1. ***getFileList***获取文件列表

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| url | string | 文件列表路径 | Y |

返回值：

|  |  |
| --- | --- |
| type | ‘LS |
| result | 0:成功；非0:失败 |
| LS（数组） | name: 文件/文件夹名 |
| Size:大小，单位Byte |
| time:最后修改时间，单位s |
| type:是否为文件夹，true文件夹false文件 |
| Dir | 当前列表路径 |
| message | 结果描述 |

Raysync.getFileList(url) //url：服务器上文件路径

Raysync.obserable.on(‘LS’,function(res){ //注册事件获取返回的列表信息

// res.result 状态，0为正常，非0失败

//res.list 文件列表数据，包含name（文件名）,size（大小）,time（最后修改时间）,type（是否为文件夹）信息

//res.Dir 当前列表对应的服务器路径

//res.message 信息结果说明

})

1. ***Upload***上传文件/文件夹

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| type | string | 0：文件；1：文件夹 | Y |
| path | string | 上传至服务端路径 | Y |

无返回

Raysync.upload(type,path) //type：0：文件；1：文件夹

1. ***Download***下载文件/文件夹

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| list | Arry | 由每个文件的路径组成的数组 | Y |

无返回

Raysync.download(list) //list：文件全路径组成的数组，

如：

[

‘/test/a.html’,

‘/test/b.js’

]

1. ***RM***删除文件/文件夹

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| list | Arry | 由每个文件的路径组成的数组 | Y |

无返回

Raysync.RM(list) //list：文件全路径组成的数组，

如：

[

‘/test/a.html’,

‘/test/b.js’

]

1. ***MKdirRequest***新建文件夹

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| dir | string | 新建文件夹名 | Y |
| url | string | 服务端路径 | Y |

Raysync.MKdirRequest(dir,url) //dir：新建文件夹名，url：服务器路径，即哪个目录//下新建

Raysync.obserable.on(‘MKDIR’,function(res){ //注册事件回调监听是否新建成功

//res.result 0成功；非0失败

})

1. ***Rename***重命名

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | type | 说明 | 是否必须Y/N |
| oldname | string | 新名（全路径形式传递） | Y |
| newname | string | j旧名（全路径形式传递） | Y |

Raysync.rename(oldname,newname) //oldname：新名，newname：旧名。以全路径形式传递，如：‘/test/a.html’更名‘/test/b.html’

Raysync.obserable.on(‘rename’,function(res){ //注册事件回调监听是否成功

//res.result 0成功；非0失败

})

1. ***getTranscount***获取当前用户正在传输的任务数

Raysync.getTranscount() // 无参数需求

Raysync.obserable.on(‘GET\_TRANS\_COUNT’,function(res){ //注册事件回调监听是否成功

//res.result 0成功；非0失败

//res.count 当前任务数

})

1. ***showTransferList***打开传输列表

Raysync.showTransferList() // 无参数需求