

Octopus SDK

[1. octopus 3](#_Toc174991186)

[2. mysql 6](#_Toc174991187)

[3. mongo 9](#_Toc174991188)

[4. tcpserver 16](#_Toc174991189)

[5. tcpclient 18](#_Toc174991190)

[6. httpserver 20](#_Toc174991191)

[7. httpclient 23](#_Toc174991192)

[8. udpserver 24](#_Toc174991193)

[9. udpclient 27](#_Toc174991194)

[10. kcpserver 28](#_Toc174991195)

[11. kcpclient 31](#_Toc174991196)

[12. websocketServer 32](#_Toc174991197)

[13. aoi 35](#_Toc174991198)

[14. pb 37](#_Toc174991199)

[15. bson 37](#_Toc174991200)

[16. seri 38](#_Toc174991201)

[17. lfs 38](#_Toc174991202)

[18. cjson 39](#_Toc174991203)

# octopus

1. get\_octopus\_verion 得到octopus的版本号

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ret | octopus的版本号 |

1. set\_console\_handler 设置控制台的回调方法

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| function | handler | 回调函数 | 1 (必填参数) |

handler(cmd)

1. add\_timer 增加一个时钟

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | interval | 间隔时间(毫秒) | 1 (必填参数) |
| function | handler | 回调函数 | 2 (必填参数) |
| table | param | 回调参数 | 3 (可选参数) |

handler(timer\_id, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| int | timer\_id | 时钟ID |

1. remove\_timer 删除一个时钟

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | timer\_id | 时钟ID | 1 |

1. stop\_all\_timer 停止/删除所有时钟
2. set\_log\_level 设置日志的显示等级

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | log\_level | 日志等级1--4 | 1 |

1. get\_log\_level 得到日志的显示等级

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| int | ret |  |

1. save\_log 保存日志
2. write\_log 写日志

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | str | 日志内容 | 1 |
| int | log\_level | 日志等级1—4 | 2 |

1. stop 停止服务

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| function | handler | 回调函数 | 1 |

1. exit 退出服务
2. loadxml 读xml文件到table

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | file\_name | 文件名 | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| table | t |  |

1. get\_config\_file 得到config的文件名

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ret |  |

1. get\_ms\_time 得到的当前时间ms

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| int | ret |  |

# mysql

1. new 创建一个mysql对象
2. initialize 初始化数据库连接，并创建n个连接线程

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | hostname | 主机名(IP) | 1 |
| int | port | 端口号 | 2 |
| string | username | 用户名 | 3 |
| string | password | 密码 | 4 |
| string | database | 数据库名 | 5 |
| string | charset\_name | 字符编码utf8 | 6 |
| int | conn\_count | 连接的数量 | 7 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 是否成功 |

1. shutdown 关闭数据库的所有连接
2. get\_next\_connect 得到下一个连接

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| int | conn\_idx | 连接ID |

1. get\_free\_connect 得到一个空闲的连接

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| int | conn\_idx | 连接ID |

1. add\_connect\_ref 增加连接引用计数

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. remove\_connect\_ref移除连接引用计数

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. add\_query 增加一个sql查询操作

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | sql | sql语句 | 1 |
| funtion | handler | 回调函数 | 2 |
| table | sql\_param | sql参数 | 3 |
| int | bind\_conn\_idx | 绑定的连接ID | 4 |
| table | param | 回调参数 | 5 |
| int | tran\_flag | 事务类型 | 6 |

handler(is\_success, rs, param)

1. insert\_frist\_query 在查询队列的最前面插入一个sql查询操作

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | sql | sql语句 | 1 |
| funtion | handler | 回调函数 | 2 |
| table | sql\_param | sql参数 | 3 |
| int | bind\_conn\_idx | 绑定的连接ID | 4 |
| table | param | 回调参数 | 5 |

handler(is\_success, rs, param)

1. ping 连接存活检测心脏包

# mongo

1. new 创建一个mongo对象
2. initialize 初始化数据库连接，并创建n个连接线程

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | dri | 连接串 | 1 |
| string | db\_name | 数据库名 | 2 |
| int | conn\_count | 连接数量 | 3 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 是否成功 |

1. shutdown 关闭数据库的所有连接
2. get\_next\_connect 得到下一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_free\_connect 得到一个空闲连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. add\_connect\_ref 增加连接引用计数

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. remove\_connect\_ref 移除连接引用计数

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. command\_simple 执行commond命令

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| table | cmd | 命令内容 | 1 |
| function | handler | 回调函数 | 2 |
| table | param | 回调参数 | 3 |
| int | conn\_idx | 连接ID | 4 |

handler(is\_success, rs, param)

1. ping 连接存活检测心脏包
2. create\_index 创建一个索引

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | data | 索引内容 | 2 |
| function | handler | 回调函数 | 3 |

handler(is\_success, rs)

1. drop\_index 删除一个索引

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| string | index\_name | 索引名 | 2 |
| function | handler | 回调函数 | 3 |

handler(is\_success, rs)

1. drop 删除一个表结构

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| function | handler | 回调函数 | 2 |

handler(is\_success, rs)

1. find\_one 查询表的一行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | query | 查询内容 | 2 |
| table | field | 字段内容 | 3 |
| function | handler | 回调函数 | 4 |
| table | param | 回调参数 | 5 |
| int | conn\_idx | 连接ID | 6 |

handler(is\_success, rs, param)

1. find查询表的多行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | query | 查询内容 | 2 |
| table | field | 字段内容 | 3 |
| function | handler | 回调函数 | 4 |
| table | param | 回调参数 | 5 |
| int | conn\_idx | 连接ID | 6 |
| int | limit | 限制的行数 | 7 |

handler(is\_success, rs, param)

1. update\_one 更新表的一行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | update | 更新内容 | 2 |
| table | filter | 更新过滤(限制) | 3 |
| function | handler | 回调函数 | 4 |
| table | param | 回调参数 | 5 |
| int | conn\_idx | 连接ID | 6 |

handler(is\_success, rs, param)

1. update\_many 更新表的多行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | update | 更新内容 | 2 |
| table | filter | 更新过滤(限制) | 3 |
| function | handler | 回调函数 | 4 |
| table | param | 回调参数 | 5 |
| int | conn\_idx | 连接ID | 6 |

handler(is\_success, rs, param)

1. insert\_one 插入一行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | insert | 插入内容 | 2 |
| function | handler | 回调函数 | 3 |
| table | param | 回调参数 | 4 |
| int | conn\_idx | 连接ID | 5 |

handler(is\_success, rs, param)

1. insert\_many插入多行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | insert | 插入内容 | 2 |
| function | handler | 回调函数 | 3 |
| table | param | 回调参数 | 4 |
| int | conn\_idx | 连接ID | 5 |

handler(is\_success, rs, param)

1. delete\_one 删除表的一行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | delete | 删除内容 | 2 |
| function | handler | 回调函数 | 3 |
| table | param | 回调参数 | 4 |
| int | conn\_idx | 连接ID | 5 |

handler(is\_success, rs, param)

1. delete\_many 删除表的多行记录

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | delete | 删除内容 | 2 |
| function | handler | 回调函数 | 3 |
| table | param | 回调参数 | 4 |
| int | conn\_idx | 连接ID | 5 |

handler(is\_success, rs, param)

1. find\_and\_modify 查找并更新一个字段值(用于serial\_idx递增+1)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | table\_name | 表名 | 1 |
| table | query | 查询内容 | 2 |
| table | update | 更新内容 | 3 |
| function | handler | 回调函数 | 4 |
| table | param | 回调参数 | 5 |
| int | conn\_idx | 连接ID | 6 |

handler(is\_success, rs, param)

# tcpserver

1. new 创建一个tcpserver对象
2. start 开启服务器监听

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 成功/失败 |

1. close 关闭服务器
2. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# tcpclient

1. new 创建一个tcpclient对象
2. connect 连接服务器

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

1. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# httpserver

1. new 创建一个httpserver对象
2. start 开启服务器监听

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 成功/失败 |

1. close 关闭服务器
2. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

1. clear 清除Response缓存

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. set\_error设置Response错误码

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| int | code | 代号 | 2 |
| string | msg | 内容 | 3 |

1. set\_body 设置Response文本

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | body | 主本 | 2 |

1. response发送响应

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

# httpclient

1. new 创建一个httpclient对象
2. init 初始化httpclient对象

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 是否成功 |

1. close 关闭连接
2. set\_timeout 设置消息回复的超时时间

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | time | 超时时间(毫秒) | 1 |

1. get 请求一个get操作

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | url | 请求URL地址 | 1 |
| funtion | handler | 回调函数 | 2 |
| string | custom\_http\_head | http附加的头信息 | 3 |

handler(is\_success, data, use\_time)

1. post 请求一个post操作

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | url | 请求URL地址 | 1 |
| string | data | post的数据 | 2 |
| funtion | handler | 回调函数 | 3 |
| string | custom\_http\_head | http附加的头信息 | 4 |

handler(is\_success, data, use\_time)

# udpserver

1. new 创建一个udpserver对象
2. start 开启服务器监听

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 成功/失败 |

1. close 关闭服务器
2. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# udpclient

1. new 创建一个tcpclient对象
2. connect 连接服务器

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

1. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# kcpserver

1. new 创建一个udpserver对象
2. start 开启服务器监听

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 成功/失败 |

1. close 关闭服务器
2. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# kcpclient

1. new 创建一个tcpclient对象
2. connect 连接服务器

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

1. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# websocketServer

1. new 创建一个websocketserver对象
2. start 开启服务器监听

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| string | ip | IP | 1 |
| int | port | 端口号 | 2 |
| function | onconnected\_handler | 新连接的回调函数 | 3 |
| function | onclose\_handler | 连接断开的回调函数 | 4 |
| function | onrecv\_handler | 接收消息的回调函数 | 5 |
| int | sendbuffersize | 发送缓冲区的字节数 | 6 |
| int | recvbuffersize | 接收缓冲区的字节数 | 7 |
| table | onconnected\_param | 新连接的回调参数 | 8 |
| table | onclose\_param | 连接断开的回调参数 | 9 |
| table | onrecv\_param | 接收消息的回调参数 | 10 |

onconnected\_handler(conn\_idx, is\_success, param)

onclose\_handler(conn\_idx, param)

onrecv\_handler(conn\_idx, data, len, param)

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| bool | ret | 成功/失败 |

1. close 关闭服务器
2. send\_msg 发送消息(带协议封装)

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |
| string | msg | 消息内容 | 2 |
| int | len | 消息长度 | 3 |

1. disconnect 断开一个连接

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

1. get\_ip\_address 得到连接的IP地址

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | conn\_idx | 连接ID | 1 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| string | ip | IP |

# aoi

1. new 创建一个aoi对象
2. create 创建一个AOI对象

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | width | 宽 | 1 |
| int | Height | 高 | 2 |
| int | range | AOI范围(n\*n个格子) | 3 |
| function | onenter\_handler | 进入的回调函数 | 4 |
| function | onmove\_handler | 移动的回调函数 | 5 |
| function | onexit\_handler | 退出的回调函数 | 6 |
| int | scene\_obj\_idx | 场景ID | 7 |

onenter\_handler(src\_list, dst\_list, scene\_obj\_idx)

onmove\_handler(src\_list, dst\_list, scene\_obj\_idx)

onexit\_handler(src\_list, dst\_list, scene\_obj\_idx)

1. enter 进入AOI

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | unit\_type | 对象类型 | 1 |
| int64 | obj\_idx | 对象ID | 2 |
| int | x | 2D坐标x | 3 |
| int | y | 2D坐标y | 4 |

1. move 移动

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | unit\_type | 对象类型 | 1 |
| int64 | obj\_idx | 对象ID | 2 |
| int | x | 2D坐标x | 3 |
| int | y | 2D坐标y | 4 |

1. exit 退出AOI

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | unit\_type | 对象类型 | 1 |
| int64 | obj\_idx | 对象ID | 2 |
| int | x | 2D坐标x | 3 |
| int | y | 2D坐标y | 4 |

1. get\_9\_cell 得到九宫格的玩家ID列表

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量名 | 描述 | 参数 |
| int | x | 2D坐标x | 1 |
| int | y | 2D坐标y | 2 |

返回值:

|  |  |  |
| --- | --- | --- |
| 类型 | 变量名 | 描述 |
| table | ret | UnitID列表 |

# pb

1. pack
2. unpack

# bson

1. encode
2. decode
3. encode\_order
4. date
5. timestamp
6. regex
7. binary
8. objectid

# seri

1. pack
2. unpack

# lfs

1. attributes
2. chdir
3. currentdir
4. dir
5. link
6. lock
7. mkdir
8. rmdir
9. symlinkattributes
10. setmode
11. touch
12. unlock
13. lock\_di

# cjson

1. new
2. encode
3. decode
4. pack
5. unpack
6. encode\_sparse\_array
7. encode\_max\_depth
8. decode\_max\_depth
9. encode\_number\_precision
10. encode\_keep\_buffer
11. encode\_invalid\_numbers
12. decode\_invalid\_numbers