# C++导出到lua中的接口

## 通用接口

### void log\_info(const char\* log)

重要信息日志

log：日志

### void log\_error(const char\* log)

错误日志

log：日志

### void log\_warning(const char\* log)

警告日志

log：日志

### void log\_debug(const char\* log)

调试日志

log：日志

### void log\_assert(const char\* log)

断言日志

log：日志

### void add\_lua\_timer(int index, float delay)

添加定时器

index：索引号

delay：延迟时间，单位：秒

### int get\_second\_time()

得到当前时间，time\_t类型

### int cur\_to\_days()

得到当前相对天数，用于比较

### int to\_days(int t)

得到相对天数，用于比较

t：time\_t时间

## DBServer接口

### void reg\_center\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册CenterServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_game\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册GameServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void send2center(unsigned short msgid, std::string pb)

给CenterServer发送消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2game(int game\_id, unsigned short msgid, std::string pb)

给GameServer发送消息

game\_id：游戏服ID

msgid：消息ID

pb：消息内容，protobuffer格式

### LuaDBConnectionPool\* get\_account\_db()

得到账号数据库连接

### LuaDBConnectionPool\* get\_game\_db()

得到游戏数据库连接

### LuaDBConnectionPool\* get\_log\_db()

得到日志数据库连接

### LuaDBConnectionPool

数据库连接

#### void execute\_lua(const char\* sql)

执行一条无返回sql语句

sql：sql语句

#### void execute\_update\_lua(const char\* func, int index, const char\* sql)

执行一条更新sql语句

func：回调函数名字

index：回调索引

sql：sql语句

#### void execute\_query\_lua(const char\* func, int index, bool more, const char\* sql)

执行一条有返回sql语句

func：回调函数名字

index：回调索引

more：是否返回多行数据

sql：sql语句

### RedisReply

Redis回复

#### bool is\_nil()

是否是nil类型数据

#### bool is\_error()

是否是错误数据

#### bool is\_status()

是否是状态类型数据

#### bool is\_string()

是否是字符串

#### bool is\_integer()

是否是整数

#### bool is\_array()

是否是数组

#### long long get\_integer()

得到整数

#### const char\* get\_string()

得到字符串

#### int size\_element()

得到数组长度

#### RedisReply\* get\_element(int index)

得到数组index项值

### RedisConnectionPool

Redis连接

#### void command\_lua(const char\* cmd)

执行一条无返回redis语句

cmd：redis语句

#### void command\_query\_lua(const char\* func, int index, const char\* cmd)

执行一条有返回redis语句

func：回调函数名字

index：回调索引

cmd：redis语句

### void redis\_command(const char\* cmd)

执行一条无返回redis语句

cmd：redis语句

### void redis\_command\_query(const char\* func, int index, const char\* cmd)

执行一条有返回redis语句

func：回调函数名字

index：回调索引

cmd：redis语句

### std::string to\_hex(std::string src)

二进制转十六进制

### std::string from\_hex(std::string src)

十六进制转二进制

## CenterServer接口

### void reg\_login\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册LoginServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_db\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册DBServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_game\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册GameServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_gate\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册GateServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void send2login(unsigned short msgid, std::string pb)

给LoginServer发送消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2db(unsigned short msgid, std::string pb)

给DBServer发送消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2game(int game\_id, unsigned short msgid, std::string pb)

给GameServer发送消息

game\_id：游戏服ID

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2client(int guid, int gate\_id, unsigned short msgid, std::string pb)

给客户端发送消息

guid：玩家guid

gate\_id：网关ID

msgid：消息ID

pb：消息内容，protobuffer格式

### void broadcast2client(unsigned short msgid, std::string pb)

给客户端广播消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2gate(int gate\_id, unsigned short msgid, std::string pb)

给GateServer发送消息

gate\_id：网关ID

msgid：消息ID

pb：消息内容，protobuffer格式

### int find\_a\_default\_lobby()

得到默认大厅

### CenterPlayer

CenterServer中玩家

#### int get\_guid()

得到guid

#### int get\_gate\_id()

得到网关ID

#### std::string get\_account()

得到账号名字

#### int game\_id()

得到有效服务ID

### CenterPlayer\* find\_player(int guid)

通过guid查找玩家

### CenterPlayer\* find\_player\_by\_account(const char\* account)

通过账号查找玩家

## GameServer接口

### void reg\_center\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册CenterSever发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_db\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册DBServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void reg\_gate\_dispatcher(const char\* msg, unsigned short msgid, const char\* func, const char\* callback, bool show\_log)

注册GateServer发送过来的消息处理分配器

msg：消息名字

msgid：消息ID

func：消息处理函数

callback：C++调用脚本的统一回调

show\_log：是否显示日志

### void send2center(unsigned short msgid, std::string pb)

给CenterServer发送消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2db(unsigned short msgid, std::string pb)

给DBServer发送消息

msgid：消息ID

pb：消息内容，protobuffer格式

### void send2client(int guid, int gate\_id, unsigned short msgid, std::string pb)

给LoginServer发送消息

guid：玩家guid

gate\_id：网关ID

msgid：消息ID

pb：消息内容，protobuffer格式