EQ Home APIs （EQ 家园 API 接口文档）

API 制定规则：

这里是 API 制定规则，客户端提供给脚本的接口分三大类。

1. **第一类为策划类 API，API 访问域为 DESIGN**，此类 API 只限于跟界面逻辑相关有需求才开放提供，这部分开放出来的 API 主要也是给策划进行开发上的使用，开放的 API 接口原则上应该简单，易用，主要有全局函数，游戏事件触发，游戏对象及全局变量。
   1） 游戏对象：开放 UI 界面显示数值相关的操作和 UI 事件触发时调用的接口
   2） 游戏事件：开放 UI 界面显示需用到的事件触发
   3） 全局函数：开放 UI 界面显示数值相关的操作和 UI 事件触发时调用的接口
   4） 全局变量：开放 UI 界面显示及逻辑操作需用到的且 C++也用到的全局变量，脚本自己用到的全局变量单独建一个文件夹来存放定义
2. **第二类为程序类 API，API 访问域为 PROGRAM**，此类开放的 API 则主要是提供给使用网络消息发送与接收处理这块，根据服务器提供的封包处理方式来开发，其 API 相比第一部分要复杂些，由程序来使用这类 API 进行开发上的使用，主要有全局函数，游戏事件触发，游戏对象及全局变量。
   1） 游戏对象：开放与网络消息相关的对象接口
   2） 游戏事件：开放各种与网络消息处理有关的游戏事件
   3） 全局函数：开放与网络消息相关的全局函数
   4） 全局变量：开放与网络消息处理有关且 C++也用到的全局变量，脚本自己用的全局变量单独建一个文件来存放定义
3. **第三类是为策划类和程序类通用型的，API 访问域为 COMMON** ，游戏事件在这边应用会比较多

备注：

接口错误调用的提示和 LOG 记录

非法的接口调用时根据弹出的窗口信息提示进行修改，相应的历史记录可以在 LOG 文件里找到信息

API 的整理分类和规则细化

阶段性的进行 API 接口整理分类，及规则更细化

（以下先列出几个示例，因为后面开发需要对前期 DEMO 使用的 API 再进行全部的整理，归档）

全局函数

1．Void changeGirdMode(int sel)

参数： 改变的模式
返回值： void
示例： changeGirdMode(sel))
描述： 切换格子显示模式
访问域： DESIGN

2． Void changeFitmentMode(int sel)

参数： 改变的模式
返回值： void
示例： changeFitmentMode(sel))
描述： 切换家装模式和正常游戏模式
访问域： DESIGN

3．Bool versinoRequest(string account, string serverIP, int serverPort)

参数： 帐号，服务器 IP，服务器端口
返回值： bool
示例： versinoRequest(account, serverIP, serverPort)
描述： 版本验证请求
访问域： PROGRAM

4．Bool createPlayerRdquest(string account, string username)

参数： 帐号，用户名
返回值： bool
示例： createPlayerRdquest (account,username)
描述： 创建角色请求
访问域： PROGRAM.

游戏事件

事件	*描述*	*访问域*
GAME_INIT_DONE	游戏初始化完毕	COMMON.
VERSION_OK	版本验证成功	COMMON
VERSION_FAILED	版本验证失败	COMMON
LOGIN_OK	登录成功	COMMON.
LOGIN_FAILED	登录失败	COMMON
ROLE_LIST_RETURN_EMPTY	角色列表返回空	COMMON
ROLE_CREATED_FAILED	创建角色失败	COMMON
REQUEST_SCENE_OK	请求场景成功	COMMON
REQUEST_SCENT_FAILED	请求场景失败	COMMON
FIT_NEW_ADDED	家装新添加	COMMON

游戏对象

1． LuaManager

调用：　　EQGlobaClass.getLuaManager()
访问域：　　COMMON

函数信息：
loadState(string)

返回值：　　无
示例：　　loadState(“***.lua”)
描述：　　重新载入 LUA 脚本文件
访问域：　　COMMON

2． GameEventMgr

调用：　　EQGlobaClass.getGameEventMgr()
访问域：　　COMMON

函数信息：
1）regEvent(string)
参数：　　string 事件名称

返回值：　　无
示例：　　regEvent(“fit_new_added”)
描述：　　注册新的游戏事件
访问域：　　comment

2）subscribeEvent(string,string)
参数：　　1：事件名称 2：被绑定 LUA 函数名称
返回值：　　无
示例：　　subscribeEvent(“FIT_NEW_ADDED”,”recefitaddednotify”)
描述：　　绑定函数到游戏事件
访问域：　　comment

3）fireEvent(string)
参数：　　游戏事件名称
返回值：　　无
示例：　　fireEvent(“on_clicked_login”)
描述：　　执行游戏事件绑定函数
访问域：　　comment

全局变量

变量初始值
PRO_BOOKTYPE_WOR_TIME　　4.0
描述：　　描述在这里
访问域：　　COMMON
。。。。