兰亭游戏联机平台SDK UE4接口说明

目录

[兰亭游戏联机平台SDK UE4接口说明 1](#_Toc24622339)

[1. API接口调用流程 2](#_Toc24622340)

[2. API接口函数 3](#_Toc24622341)

[2.1 连接平台接口 3](#_Toc24622342)

[2.2 获取房间列表接口 4](#_Toc24622343)

[2.3 创建游戏房间接口 5](#_Toc24622344)

[2.4 刷新房间信息 6](#_Toc24622345)

[2.5 用户加入指定房间接口 7](#_Toc24622346)

[2.5 获取指定房间用户信息接口 8](#_Toc24622347)

[2.6 用户刷新修改个人信息接口 9](#_Toc24622348)

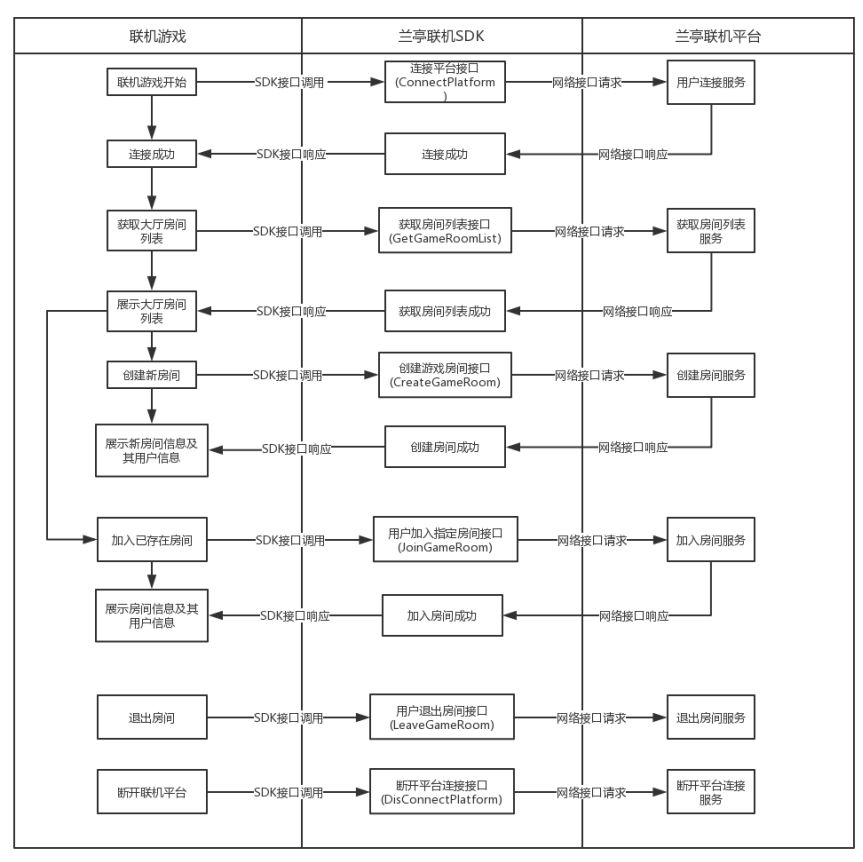
[2.7 用户修改房间状态接口 9](#_Toc24622349)

[2.8 用户退出房间接口 10](#_Toc24622350)

[2.9 断开平台连接接口 10](#_Toc24622351)

[2.10 断开网络连接接口 11](#_Toc24622352)

[2.11 用户心跳接口 11](#_Toc24622353)

1. API接口调用流程

**2. API接口函数**

所有接口均会返回 code和message值，用以判断通信是否成功，其余部分为各个接口独立的返回值，在个接口说明中均有说明。

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| code | 返回状态码（00000为成功；其他状态码为失败） | string |  |
| msg | 提示信息 | string |  |

2.1 连接平台接口

void ConnectPlatform(string userNick)

说明：通过调用该接口实现连接联机平台

调用方式：

1）调用发送消息接口

LetinvrSDK::ConnectPlatform(string cpid, string cpgameid, string userNick);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK::GetConnectData()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| cpId | 厂商ID | String |  |
| gameId | 游戏ID | String |  |
| userName | 用户名称 | String |  |
| userNick | 用户昵称 | String |  |

独立返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| sessionid | 身份ID | String | 32位uuid字符串 |
| userId | 用户ID | String |  |
| userName | 用户名称 | String |  |
| userNick | 用户昵称 | String |  |
| ip | 用户IP | String |  |
| port | 用户端口 | int |  |

**2.2 获取房间列表接口**

void GetGameRoomList()

说明：通过调用该接口获取平台上已有房间列表

调用方式：

1）调用发送消息接口

LetinvrSDK:: GetGameRoomList (std::string cpId, std::string gameId);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK::GetRoomListData()->xxx

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| cpId | 厂商ID | string |  |
| gameId | 游戏ID | string |  |

独立返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| roomId | 房间ID | string |  |
| roomName | 房间名 | string |  |
| maxNumbers | 最多用户数 | int |  |
| currentNumbers | 当前用户数 | int |  |
| status | 状态（0 等待；1 游戏中） | int |  |
| custom | 可选参数，不用时为空 | string |  |

**2.3 创建游戏房间**接口

void CreateGameRoom(std::string cpId, std::string gameId, std::string roomName, int maxMembers, std::string custom = "")

说明：连接平台后, 用户通过调用该接口创建新的游戏房间

调用方式：

1）调用发送消息接口

LetinvrSDK:: CreateGameRoom(string cpid, string cpgameid, string userNick);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: GetCreateRoomData()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| cpId | 厂商ID | string |  |
| gameId | 游戏ID | string |  |
| roomName | 房间名 | string |  |
| maxMembers | 最多用户数 | int |  |
| custom | 可自定义属性 | string | 可选参数，不用时为空 |

独立返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| roomId | 房间ID | string |  |
| roomName | 房间名称 | string |  |

2.4 刷新房间信息

void RefreshRoomInfo(std::string roomId, std::string custom = "");

说明：刷新房间的自定义信息。

1）调用发送消息接口

LetinvrSDK:: RefreshRoomInfo(string cpid, string cpgameid, string userNick);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK::GetRefreshRoomData()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| roomId | 房间ID | string |  |
| custom | 可选参数，不用时为空 | string |  |

独立返回参数：无

**2.5 用户加入指定房间接口**

void JoinGameRoom(string roomId)

说明：用户通过调用该接口实现加入指定房间或刷新已经入房间用户数据功能

调用方式：

1）调用发送消息接口

LetinvrSDK:: JoinGameRoom(std::string roomId);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: GetJoinRoomData ()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| roomId | 房间ID | string |  |

独立返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| sessionid | 身份ID | string | 32位uuid字符串 |
| userNick | 用户昵称 | string |  |
| ip | 用户IP | string |  |
| port | 用户端口 | int |  |
| isMaster | 是否是主机：1主；0 从机 | int |  |
| custom | 可选参数，不用时为空 | string |  |

**2.5 获取指定房间用户信息接口**

void GetGameRoomUserList(string roomId)

说明：用户通过调用该接口实现获取房间用户数据功能

调用方式：

1）调用发送消息接口

LetinvrSDK::GetGameRoomUserList (std::string roomId);

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: GetUserListData()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| roomId | 通过获取房间列表接口获得 | string |  |

独立返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** |
| sessionid | 身份ID | String | 32位uuid字符串 |
| userNick | 用户昵称 | String |  |
| ip | 用户IP | String |  |
| port | 用户端口 | int |  |
| isMaster | 是否是主机：1主；0 从机 | int |  |
| custom | 可选参数，不用时为空 | string |  |

**2.6 用户刷新修改个人信息**接口

void ModifyUserStatus (std::string custom = "")

说明：用户通过调用该接口修改个人信息状态

调用方式：

1）调用发送消息接口

LetinvrSDK::ModifyUserStatus(std::string custom = "");

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK::GetModifyGameData()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| custom | 可选参数，不用时为空 | string |  |

独立返回参数：无

**2.7 用户修改房间状态**接口

void RefreshRoomStatus(std::string roomId, int status)

说明：用户通过调用该接口修改房间状态

调用方式：

1）调用发送消息接口

LetinvrSDK:: RefreshRoomStatus(std::string roomId, int status)

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: GetRefreshRoomStatuData ()->xxx

输入参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **备注** |
| roomId | 房间ID | String |  |
| status | 状态 | int | 0 结束；1 开始 |

独立返回参数：无

**2.8 用户退出房间**接口

void QuitGameRoom(std::string roomId)

说明：用户通过调用该接口退出已加入的游戏房间

调用方式：

1）调用发送消息接口

LetinvrSDK:: QuitGameRoom(std::string roomId)

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: GetQuitGameData()->xxx

输入参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名** | **含义** | **类型** | **长度** | **备注** |
| roomid | 创建或加入的房间ID | string |  |  |

独立返回参数：无

**2.9 断开平台连接接口**

void DisconnectPlatform()

说明：用户通过调用该接口实现发送断开连接消息

调用方式：

1）调用发送消息接口

LetinvrSDK:: DisconnectPlatform(std::string roomId)

2）根据GameData.h循环调用获取数值接口，数据会异步写入数据类。  
 LetinvrSDK:: DisconnectPlatform()->xxx

独立返回参数：无

2.10 断开网络连接接口

void CloseConnectWithServer ()

说明：用户通过调用该接口实现断开平台网络连接，在收到断开平台返回值成功后调用。关闭与平台网络连接，不再能接收到任何消息。

调用方式：

调用断开接口：

LetinvrSDK:: CloseConnectWithServer ();

2.11 用户心跳接口

void ConnectHeartBeats()

说明：用户心跳，定时发送心跳请求，每10秒一次，若3次没收到用户心跳，则认为用户已断开连接。

用户断开连接处理：

1. 房间用户列表中清除用户。
2. 断开平台连接，清除用户信息。

输入参数：无

独立返回参数：无