

# 英雄联盟本地数据采集器开发文档 V1.0

## 1. GameData 数据格式

### 1.1 根目录下

accountId 用户账户 ID (每个用户账户 ID 是全网唯一的)  
gameLength 比赛时间长度, 单位是秒  
gameMode 游戏模式 CLASSIC(常规地图模式)、ARAM(大乱斗地图模式) 其实 gameMode 本人看来叫做地图模式更加恰当, 因为这个字段是能够表示用户战斗的地图选择  
gameType 比赛类型 MATCHED\_GAME  
queueType 比赛类型(这个最重要, 通过这个可以判断玩家是打电脑、打常规、打匹配), BOT(对战电脑模式)、NORMAL(常规赛)、RANKED\_FLEX\_SR(灵活组排)、RANKED\_SOLO\_5x5(排位)、ARAM\_UNRANKED\_5x5(大乱斗)  
skinId 你的账户头像 Id  
summonerId 你账户的 Id  
summonerName 你账户的名称  
userId 你账户的 Id(与 summonerId 重复了)  
reportGameId 这场比赛的唯一 Id

### 1.2 players 键值下的信息

botPlayer 是否是电脑玩家  
items 代表用户的装备情况(如果有需要, 后面会补充)  
leaver 是否逃跑  
profileIconId 头像 Id  
skinName 使用的英雄名称  
spell1Id 小技能栏 1(D)  
spell2Id 小技能栏 2(F)  
summonerId 用户账户 Id  
summonerName 账户名称  
teamId (100 或 200)代表两个不同的队伍  
userId 也是用户账户 Id, 重复了  
wins 胜利的总场次  
losses 失败的总场次  
level 账户等级(最高 30 级)

stats 下面是本用户的详细数据(这个子键下面的数据非常非常非常重要! 是一个玩家这场比赛的输出和其他的数据, 要利用好!)

ASSISTS 助攻次数  
BARRACKS\_KILLED 拆敌方兵营数目  
CHAMPIONS\_KILLED 击杀敌方英雄数

COMBAT\_PLAYER\_SCORE  
 GAME\_ENDED\_IN\_EARLY\_SURRENDER 是否是 5 分钟投降  
 GAME\_ENDED\_IN\_SURRENDER 是否是 20 分钟投降  
 GOLD\_EARNED 刷金钱的总数  
 LARGEST\_CRITICAL\_STRIKE 最大关键袭击  
 LARGEST\_KILLING\_SPREE 最大连杀次数  
 LARGEST\_MULTI\_KILL 最大多杀次数  
 LEVEL 等级  
 MAGIC\_DAMAGE\_DEALT\_PLAYER 魔法输出量  
 MAGIC\_DAMAGE\_DEALT\_TO\_CHAMPIONS 对英雄魔法输出量  
 MAGIC\_DAMAGE\_TAKEN 承受魔法伤害  
 MINIONS\_KILLED 补兵数目  
 NEUTRAL\_MINIONS\_KILLED 野怪击杀数目  
 NEUTRAL\_MINIONS\_KILLED\_ENEMY\_JUNGLE 在敌方野区击杀野怪数  
 NEUTRAL\_MINIONS\_KILLED\_YOUR\_JUNGLE 在我方野区击杀野怪数  
 NUM\_DEATHS 死亡次数  
 OBJECTIVE\_PLAYER\_SCORE ? ? ?  
 PHYSICAL\_DAMAGE\_DEALT\_PLAYER 物理输出量  
 PHYSICAL\_DAMAGE\_DEALT\_TO\_CHAMPIONS 对英雄物理输出量  
 PHYSICAL\_DAMAGE\_TAKEN 承受物理输出量  
 SIGHT\_WARDS\_BOUGHT\_IN\_GAME 购买眼的数量  
 TOTAL\_DAMAGE\_DEALT 总输出量，物理输出、魔法输出等  
 TOTAL\_DAMAGE\_DEALT\_TO\_BUILDINGS 对建筑的总输出  
 TOTAL\_DAMAGE\_DEALT\_TO\_CHAMPIONS 对英雄的总输出  
 TOTAL\_DAMAGE\_DEALT\_TO\_TURRETS 对塔的总输出量  
 TOTAL\_DAMAGE\_SELF\_MITIGATED 自身通过护甲等减少敌人对自己的伤害量  
 TOTAL\_DAMAGE\_TAKEN 承受的总伤害量  
 TOTAL\_HEAL 总治愈量  
 TOTAL\_TIME\_CROWD\_CONTROL\_DEALT 技能控制敌方英雄的总时间  
 TOTAL\_TIME\_SPENT\_DEAD 死亡的总时间  
 TRUE\_DAMAGE\_DEALT\_PLAYER 真实伤害总量  
 TRUE\_DAMAGE\_DEALT\_TO\_CHAMPIONS 对敌方英雄真实伤害的总量  
 TRUE\_DAMAGE\_TAKEN 承受真实伤害的总量  
 TURRETS\_KILLED 推塔数目  
 VISION\_SCORE 插真眼数目  
 VISION\_WARDS\_BOUGHT\_IN\_GAME 买真眼数目  
 WARD\_KILLED 反眼数目  
 WARD\_PLACED 插眼数目  
 WIN 是否胜利

## 2. LoginData 数据格式

进入游戏选完英雄开始匹配获取的数据(可以代表比赛开始，开始进入游戏界面)

displayName 玩家账号名称  
playerType 玩家类型(玩家或电脑)  
summonerId 玩家账号 Id  
championId 所选英雄的 Id, 如果有需要后期提供对应  
spell1Id 小技能栏(D)  
spell2Id 小技能栏(F)

### 3. UserInfo 数据格式

userId 用户账户 Id  
puuid 用户的 puuid  
userAuthToken 用户的 token 相关信息, 这个包含很多有用信息, 例如用户的 QQ 号、IP 地址等都可以在这里找到  
displayName 用户账户名称  
lastSeasonRank 上个赛季的段位  
rankedTierInfo 本赛季的段位信息(一般会有两个段位, 一个是灵活组排、一个是普通排位)  
serverId 服务器大区 ID  
serverName 服务器大区名称  
summonerLevel 玩家账号等级  
profileIconId 玩家账号头像 ID

### 4. LolDataHelper 的使用方法

#### C++的调用方法

```
//回调函数, 用于接收采集器抓到的数据
void _stdcall ReceiveMessage(char *args)
{
    std::string message = args;//这里 message 编码是 GBK 格式
    std::cout << message << std::endl;
}

typedef void (*pMain)(DWORD procAddress);//回调函数的定义
int _tmain(int argc, _TCHAR* argv[])
{
    std::string filePath = "LolDataHelper.dll";//dll 文件的路径

    HMODULE hModule = LoadLibrary(filePath.c_str());
    if (hModule == NULL)
    {
        std::cout << "程序运行失败!" << std::endl;
        return -1;
    }
}
```

```

pMain Main = (pMain)GetProcAddress(hModule, "Main");
Main((DWORD)ReceiveMessage);

getchar();
return 0;
}

```

## C#的调用方法

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Runtime.InteropServices;
using System.Threading;

namespace LolDemoSp
{
    class Program
    {
        [DllImport("LolDataHelperSp.dll", EntryPoint =
"GameStart", CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.Cdecl)]
        public static extern bool MyGameStart();

        [DllImport("LolDataHelperSp.dll", EntryPoint =
"GetLolMessage", CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.Cdecl)]
        public static extern bool MyGetLolMessage(StringBuilder
str);

        static void Main(string[] args)
        {
            if(MyGameStart() == true)//dll 初始化成功!
            {
                Console.WriteLine("LolDataHelper 初始化成功!");
                while(true)
                {
                    StringBuilder szMessage = new
StringBuilder("", 1024 * 100);//60K 大小满足所有数据格式
                    if (MyGetLolMessage(szMessage) == true)

```

```
        {  
            Console.WriteLine(szMessage);  
        }  
  
        Thread.Sleep(1000); //1S 遍历一次  
    }  
}  
else  
    Console.WriteLine("LolDataHelper 初始化失败!");  
  
Console.ReadLine();  
}  
}  
}
```

## 注意:

1. 在我给你们的正式版本里面, 通过回调函数传给你们的数据是通过 AES128/256 进行加密的, 为了使保护你们的数据免受这些外挂作者的破坏和非法模拟篡改, 保护你们的游戏数据安全, 保证你们的游戏规则免受这些破坏者的干扰。
2. 我给你们的没一个人的程序都是不相同的, 每一个人的程序都是有标志的, 我不希望程序会被滥用(倒卖或者违法使用), 如果被我发现我会禁用带这个标志程序, 禁用后的结果是, 轻则程序会自动退出, 重者使用会重启计算机。
3. 本人会不断优化此程序, 不定期跟新最新的插件版本, 具体的通知请查看群里的通知(群号: 595364642)

数据周杰伦  
2017-7-15 日, 深夜