**银河数娱 平台GM工具协议规范**

更新时间：2015.09.28 版本号V1.0

目录

[**1、变更记录** 3](#_Toc432062039)

[**2、概述** 3](#_Toc432062040)

[**3、通用参数** 3](#_Toc432062041)

[**4、通信协议** 4](#_Toc432062042)

[**4.1 HTTP协议规范** 4](#_Toc432062043)

[**4.2 Socket 协议规范** 5](#_Toc432062044)

[**5、接口说明** 6](#_Toc432062045)

[**5.1 配置服（管理游戏客户端的.ini配置）管理接口** 6](#_Toc432062046)

[**5.2 游戏服针对单服、全服的操作接口** 6](#_Toc432062047)

# **1、变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更时间 | 变更内容 | 变更人 |
| V1.0 | 2015-09-28 | 文档初始化 | 刘长鑫 |
|  |  |  |  |

# **2、概述**

此文档定义了平台GM管理工具与银河各个手游平台间的GM工具交互协议与规范，平台GM管理工具为接口调用方，银河各手游平台为接口提供方。

# **3、通用参数**

|  |  |  |
| --- | --- | --- |
| 术语 | 定义 | 备注 |
| cmd | 对应执行操作的接口标识 | 其值由双方协定，字段类型为整型 |
| operatorid | 操作人编号 | 字段类型为整型 |
| sign | 签名 | 字段类型为字符串，为了防止请求数据在传输过程中被篡改，因此对请求传递的内容进行签名确认，签名等于传递的req的值（如果req没有参数传递，为保证签名正常，req赋为空数组，即：“[]“）先 base64加密，再连接operatorid和密钥（appkey）的值组成字符串(base64(req) + operatorid + appkey)，再sha1加密生成签名 |
| type | 向游戏服发送请求的类型 | “json”或”text”,字段类型字符串 |
| req | 向游戏服发送的请求 | 字段类型依据type的值。如果type 值为”json”，req的字段类型为对象；type值为”text”,req的字段类型为字符串。 |
| res | 游戏服返回的请求响应 | 字段类型依据type的值。如果type 值为”json”，req的字段类型为对象；type值为”text”,req的字段类型为字符串。 |
| code | 游戏服返回的请求响应状态码 | 0表示成功，其他值为失败，字段类型为整型  0:成功;  1: 接口错误；  2: 签名错误；  3: 自定义错误，如参数错误；  4：系统未知错误 |
| msg | 游戏服返回的请求响应描述信息 | 字段类型为字符串 |
| appkey | 密钥 | 用于签名使用，由双方约定 |

# **4、签名示例**

请求的数据格式示例：

|  |
| --- |
| {  “cmd”:10000,  “operatorid”:123456,  “sign”:” 6a4f3045baea6d8eeb860a758f09793b71f8fb74”,  “type”:”json”,  “req”:{  “roleId”:”123456”  }  } |

appKey = “65987456123645987546213654987235”

备注：上面的签名sign等于上面的req的值(json格式字符串)先 base64加密，再连接operatorid和密钥（appkey）的值组成字符串（base64（req） + operatorid + appkey），再sha1加密生成签名，

下面以NodeJS代码示例：

|  |
| --- |
| var crypto = require(“crypto”);  var base64 = new Buffer(‘{“roleId”:”123456”}’,’base64’);  var reqStr = base64.toString();  var operatorid = “123456”;  var appKey = “65987456123645987546213654987235”;  var signStr = reqStr + operatorid + appKey;  var shaStr = crypto.createHash(‘sha1’);  shaStr.update(signStr);  var sign = shaStr.digest(‘hex’); |

# **5、通信协议**

适配协议规范：HTTP协议和TCP Socket协议。

## **5.1 HTTP协议规范**

* 提交方式：POST
* 数据请求和响应格式：JSON
* 请求参数使用UTF-8编码
* 中文采用URLEncode编码，防止出现乱码
* 请求头参数：Content-Type = application/x-www-urlencode

请求JSON格式：

|  |
| --- |
| {  “cmd”:number,  “operatorid”:number,  “sign”:”string”,  “type”:”json”,  “req”:{  Req Param  }  } |
|
|
|
|

响应JSON格式：

成功响应：

|  |
| --- |
| {  “code”:0,  “msg”:”success”,  “type”:”json”,  “res”:{  Res Param}} |

错误返回：

|  |
| --- |
| {  “code”:1,  “msg”:”interface name error”  } |

## **5.2 Socket 协议规范**

基于TCP Socket的短链接，单个操作和服务器通信的流程是：

[connect host –> send request -> receive response -> shutdown socket]

注意：单次传多个requests 是不被支持的。

Stream 结构：

由游戏服务端来定

请求JSON格式：

|  |
| --- |
| {  “head”:{}  “body”:{  “cmd”:number,  “operatorid”:number,  “type”:”json”,  “req”:{  Req Param  }  }  } |

响应JSON格式：

成功响应：

|  |
| --- |
| {  “head”:{},  “body”:{  “code”:0,  “msg”:”success”,  “type”:”json”,  “res”:{  Res Param  }  }  } |

错误返回：

|  |
| --- |
| {  “head”:{},  “body”:{  “code”:1,  “msg”:” interface name error”  }  } |

# **6、接口说明**

## **6.1 配置服（管理游戏客户端的.ini配置）管理接口**

## **6.2 游戏服针对单服、全服的操作接口**

## 有分享【即将更新】

## HTTP://WWW.BL20166.COM

## 冰蓝娱乐

## HTTP://WWW.DF20016.COM