**用户角色信息服务端对接文档**

[1接口说明 1](#_Toc505099909)

[2签名说明 1](#_Toc505099910)

[3 cp接入拇指游玩服务器 1](#_Toc505099911)

[3.1请求URL 1](#_Toc505099912)

[3.2请求格式示例 2](#_Toc505099913)

[3.3请求参数说明 2](#_Toc505099914)

[3.4响应说明 3](#_Toc505099915)

[3.5 JAVA示例 3](#_Toc505099916)

[4 cp接入拇指游玩服务器 4](#_Toc505099917)

[3.1 请求URL 4](#_Toc505099918)

[3.2请求参数示例 4](#_Toc505099919)

[3.3请求参数说明 4](#_Toc505099920)

[3.5 响应返回JSON参数示例 5](#_Toc505099921)

本文档主要用于拇指游玩以及CP合作方，对用户角色信息交互接口进行必要的规范。

## 1接口说明

1）所有通讯接口采用http协议，POST方法，JSON格式进行通讯。

2）参数类型为JSON字符串，参数以流的形式传入。

3）编码格式统一为：UTF-8格式。

## 2签名说明

1）签名算法：MD5加密

2）签名公式："data="+data+"apikey="+ Api\_key

备注：Api\_key 需要由我们统一配置

## 3 cp接入拇指游玩服务器

在用户创建角色成功之后，调用接口，将相应的角色信息传入。

### 3.1请求URL

为保证生产环境数据的真实性，对接时需先使用测试环境测试好，再切换生产URL。

沙盒地址：http://139.199.69.184:8080/sdk/userRoleInfo/updateRoleInfo.do

生产地址：

http://gm.91muzhi.com:8080/sdk/userRoleInfo/updateRoleInfo.do

### 3.2请求格式示例

URL：

http://139.199.69.184:8080/sdk/userRoleInfo/updateRoleInfo.do

参数：{"data":"/2QGqLu15N5hg4tVVxMJaowPDsq+n4EhPWf1+fJMpp9ejJjGfJmBP2MQQ2xmDhfq7i76tWmdSICn\r\nypALEM/otddtzbiz/itWFwkBlre49YqBMjtQCk6abIlo7D/JWI+WOHQSXQ7eJpJUWCETbVBtkuSX\r\nz0Hxv+4KMKeY+mzkoIa1gF+cH9ePb8l43hpP9T56lo2bLS5hSGVZ4TJSuv68X//q+tTR4DCzoIvQ\r\nc6BcQfbrP7uX9S6cmrGHmVW11pdK","sign":"76abc39d6a9cb93d7a433271a9729f18","game\_id":"672"}

### 3.3请求参数说明

请求参数列表如表3-1所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| data | 数据内容 |  | 必填 | 详见2.3.1 |
| game\_id | 游戏id | String | 必填 |  |
| encrypt | 加密方式 | String | 可选 | 定值：MD5 |
| sign | 签名 | String | 必填 | 详见[签名说明](#_2签名说明) |

表3-1 角色信息表

##### 3.3.1 请求data数据生成

为了保护cp方的数据安全，用户区服角色等信息需要进行加密操作。

data数据由JSON数组构成方便cp一次传入多个角色信息 每一条json记录代表一个角色信息。

JSON 数据构成如表3-2所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| user\_id | 拇指用户id | String | 必填 |  |
| game\_id | 游戏id | String | 必填 |  |
| ServerId | 服务器id | String | 必填 |  |
| ServerName | 服务器名称 | String | 必填 |  |
| RoleId | 角色id | String | 必填 |  |
| RoleName | 角色名称 | String | 必填 |  |
| RoleLevel | 角色等级 | String | 必填 |  |
| ChargeNum | 游戏充值金额 | String | 必填 | 单位（元） |
| RoleVipLv | 游戏vip等级 | String | 必填 |  |

表3-2 角色信息表

我们在生成JSON数组之后需要对JSON数组字符串进行ＡＥＳ加密。

说明：AES加解密这里采用CBC模式，并且为了配置简单，采用key=iv，且为16个字符128位的方式进行加密，这里的key和iv都由我们定制。

### 3.4响应说明

成功应答：success

失败应答：false

### 3.5 JAVA示例

JSONObject json = new JSONObject();

 json.put("RoleName", "小仙子");

 json.put("ServerId", "100001");

 json.put("ServerName", "CP666");

 json.put("RoleVipLv", "8");

 json.put("game\_id", "100");

 json.put("ChargeNum", "110");

 json.put("user\_id", "123456");

 json.put("RoleLevel", "23");

 json.put("RoleId", "2017072600001");

String data ="";

JSONArray jsonArray = new JSONArray();

jsonArray.add(json);

System.out.println(jsonArray);

AESOperator aesOperator = new AESOperator("wGqORfFEnNNjXDuD", "wGqORfFEnNNjXDuD");

try {

 data = aesOperator.encrypt(jsonArray.toString());

 System.out.println(data);

} catch (Exception e) {

 e.printStackTrace();

}

String sign = Rsa.getMD5("data="+data+"apikey="+"apikey520");

System.out.println(sign);

JSONObject param = new JSONObject();

param.put("data", data);

param.put("sign", sign);

param.put("game\_id", "672");

System.out.println(param.toString());

HttpUtils.SendPostJson("http://gm.91muzhi.com:8080/sdk/userRoleInfo/updateRoleInfo.do", param,1000);

## 4 cp接入拇指游玩服务器

为了获取用户角色信息，需要cp提供一个接口，用于获取角色的详细信息。

### 4.1 请求URL

由CP合作方提供URL，接收参数见表3-1。

### 4.2请求参数示例

{"data":"vtfdaO8Eo8DCql+A84l025gW13mTinRGIqLQUtWkiYM=","sign":" 7268a7e2f96a5a67215410d63a192ffc"}

说明：此处AES加密 key=iv= wGqORfFEnNNjXDuD ,API\_KEY= apikey520

### 4.3请求参数说明

请求参数列表如表3-1所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| data | 请求内容 | String | 必填 |  |
| encrypt | 加密方式 | String | 可选 | 定值：MD5 |
| sign | 签名 | String | 必填 | 详见[签名说明](#_2签名说明) |

表4-1 接口参数

#### 4.3.1 请求参数data内容

为了保护数据安全，拇指用户信息需要进行加密。

data数据由JSON数据构成以便版本更新

JSON 数据构成如表4-2所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| user\_id | 拇指用户id | String | 必填 |  |

表4-2 用户id参数

我们在生成json数据之后 需要对JSON数据格式字符串进行ＡＥＳ加密

说明：AES加解密这里采用CBC模式，并且为了配置简单，采用key=iv，且为16个字符128位的方式进行加密，这里的key和iv都由我们定制。

#### 4.4响应返回参数

接口响应参数列表4-3所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| data | 返回内容 | String | 必填 |  |
| code | 返回码 | String | 可选 |  |
| msg | 返回码说明 | String | 可选 |  |

表4-3 接口响应参数列表

#### 4.4.1 响应data参数内容

data数据由JSON数组构成方便cp一次传入多个角色信息 每一个json代表一个角色信息

JSON 数据构成如表3-4所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **数据类型** | **是否可选** | **参数说明** |
| ServerId | 服务器id | String | 必填 |  |
| ServerName | 服务器名称 | String | 必填 |  |
| RoleId | 角色id | String | 必填 |  |
| RoleName | 角色名称 | String | 必填 |  |
| RoleLevel | 角色等级 | String | 必填 |  |
| ChargeNum | 游戏充值金额 | String | 必填 | 单位（元） |
| RoleVipLv | 游戏vip等级 | String | 必填 |  |

表4-4响应data参数列表

### 4.5 响应返回JSON参数示例

{

 "code": 0,

 "msg": "成功",

 "data": [

 {

 "ServerId": 10001,

 "ServerName": "1服 大明湖畔",

 "RoleId": "6be60748-4f44-4f20-9249-6f2f8ba74f1f",

 "RoleName": "宣问晴",

 "RoleLevel": 31,

 "RoleVipLv": 0,

 "ChargeNum": 0

 }

 ]

}