GM开放接口协议

|  |  |  |
| --- | --- | --- |
| 版本号 | 日期 | 修改说明 |
| 1.0 | 2013-03-18 | 新建 |

协议数据交换直接走TCP协议，协议要遵守JSON格式。所有来回交换的数据都要标识当前协议的类型。

示例JSON

{

“action”:0,

“data”:{…}

}

以下协议用于描述当前协议段“data”的数据结构。

1、OAuth协议授权流程

OAuth协议指在为用户提供安全开放的接入接口标准

OAuth协议协议分为两步：

获取随机的字条串。

2用密码加密随机的字条串，再与平台交互认证。

协议说明：密码认证，获取接入授权

协议类型：” action”:0

接口参数：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Values |
| ext | int | 暂时不用（保留） |  |

（表1-1）

请求JSON示例：

{

“ext”:0

}

接口返回：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| oauth\_string | string | 随机验证码 |  |

（表1-2）

请求JSON示例：

{

“oauth\_string”:”12345678”

}

协议说明：密码认证，获取接入授权

协议类型：” action”:1

接口参数：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Values |
| oauth\_name | string | 用户名 |  |
| oauth\_string | string | 用户随机验证码（DES加密） | （用密码加密由第一步获取的字条串） |

（表1-1）

请求JSON示例：

{

“oauth\_name”:”apple”,

“oauth\_string”:” d2ffe9d53f682f260128c02f54b0fb2c773b7cbb”

}

接口返回：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| oauth\_state | int |  | 0:登录成功  1:用户密码验证失败 |

（表1-2）

请求JSON示例：

{

“oauth\_state”:0

}

2、获取当前服务的运行状态流程

运行状态流程指在为用户提供获取当前服务器的运行状态

协议说明：获取服务器的运行状态

协议类型：” action”:2

接口参数：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| ext | int | 暂时不用 | 0 |

（表2-1）

请求JSON示例：

{

“ext”:0

}

接口返回：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| num | int | 显区个数 |  |
| Area0 | … | … | 参考（表2-2） |

（表2-1）

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| areaID | int | 区ID |  |
| num | int | 运行程序数 |  |
| Daemon0 | String | 进程描述信息 |  |
| Daemon1 | … | … | 参考（表2-3） |
| … |  |  |  |

（表2-2）

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| name | string | 程序名称 |  |
| useFlag | int | 是否启动 | 0：不启动  1：启动 |
| alive | int | 当前运行状态 | 0：进程死亡  1：进程运行中 |

（表2-3）

请求JSON示例：

{

“num”:1,

“Area0”:”daemon0”

{

“name”:”server”,

“useFlag”:1,

“alive”:1

}

}

3、增加玩家相关属性协议流程

增加玩家相关属性协议指在为用户提供相关玩家的属性加成接口标准

协议说明：相关玩家的属性加成

协议类型：” action”:3

接口参数：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Values |
| areaid | int | 区域Id |  |
| playerid | int | 玩家Id |  |
| type | int | 增加类型 | 1、铜钱  2、元宝  3、阅历  4、科技点  5、体力  6、建筑  7、物品  8、配置生效 |
| Parm1 | int | 增加值数量 |  |
| Parm2 | int | 增加值数量 | 暂时不用 |

请求JSON示例：

{

“areaid”:1,

“areaid”:5011,

“type”:1，

“Parm1”：100，

“Parm2”：100

}

接口返回：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| action | int | 协议类型 | 0 |
| state | int |  | 0：成功  1：区域不存在  2：玩家不存在  3：增加类型不存在  4：其它错误 |

请求JSON示例：

{

“state”:0

}

4、服务器控制运行协议流程

协议说明：服务器控制运行协议为关闭或者重启服务器提供*接口*标准

协议类型：” action”:4

接口参数：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Values |
| action | int | 协议类型 | 0 |
| areaid | int | 区域Id |  |
| useFlag | int | 是否重启 | 0：关闭  1：启用 |

请求JSON示例：

{

“areaid”:1,

“useFlag”:1,

}

接口返回：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description |  |
| state | int |  | 0:成功  1:区域不存在 |

请求JSON示例：

{

“state”:0

}