

# 接口规范

## 接口规范

### 1 接口总体规范

总体规范定义传输协议、字符编码、消息结构等，具体如下

#### 1.1 接口方式

接口协议： HTTPS{POST+JSON}

请求内容字符编码： UTF-8编码

Http Header content-type部分需要遵循下面的约定

content-type=application/json

#### 1.2 接口安全策略

1. 所有API接口必须采用HTTPS以免遭到中间人攻击。
2. 对服务器或第三方接口必须制定IP访问限制策略。
3. 接口数据内容加密方案由项目小组根据项目情况自行决定。

#### 1.3 消息结构

本项目采用标准的Http消息结构传输数据，消息包括两个部分：Message

Header(消息头)与Message Content(消息体)。其中，Message Header用于存储交易请求以及应答的公共部分和消息的控制数据，Message

Content包含交易请求和应答的业务数据。在本文档中以表格的形式表示JSON格式，以下为表格中对元素约束（出现次数）的表示方法的描述：

	A	B
1	符号	含义
2	?	0..1, 可选项
3	*	0..n, 可以没有, 也可以有多项
4	+	1..n, 至少有1项, 也可以有多项
5	1	数字1, 代表必须且只能填1项

## 2 接口请求

## 2.1.Http消息头定义:

	A	B	C	D
1	参数名称	约束	类型	描述
2	pv	?	String	协议版本号
3	code	1	String	请求接口编号
4	token	?	String	用户标识(用户的Token, 有效期72小时)
5	seq	1	String	请求序号 (尚未实现)

消息体为JSON格式并采用Post方式发送

## 2.2.例:

请求Url:

<http://baseurl>

Http Header:

Content-Type:application/json;

token:{token};

code:R001;

seq:123

.....

消息体见具体的接口

## 3 接口应答

### 3.1. 消息头:

	A	B	C	D
1	参数名称	约束	类型	描述
2	code	1	int	错误代码
3	seq	1	int	下次请求序号 (尚未实现)
4	data	?	object	响应数据

### 3.2. 消息说明:

1.Status Code 采用标准的Http状态码实现。

2.Description采用标准的Http状态码说明实现, 为错误代码的简要说明。

3.如果状态码为200，消息体中为返回的数据，否则消息体中为错误详细信息。

### 3.3. 例：

Http Header:

消息体：

```
JSON
1 {"code":0,"data":{"token":"base64字符串"}}
```

## 模板

接口序号: 接口名

接口名称	XXXX		
接口编号	模块英文缩写+接口编号（补满3位）,如U001		
接口说明	XXXXXXXX		
请求方法	Post		
Headers参数	参数名称	类型	参数说明
	以下为每次请求必须带的参数		
	code	string	接口编号
	seq	int	请求序号 (该值只针对游戏服务器; 第一次请求为进入游戏,此时seq=0; 为了防止高频刷接口)（尚未实现)
token	string	授权编码(在登录成功和进入游戏这两个token是不一样的)	
Body参数	参数名称	类型	参数说明
	参数英文名用缩写, 尽量简短,如 name	String	账户名
	pwd	String	账户密码
错误码	模块编号() +(接口编号-1)+错误码顺序, 如100001	-	错误码说明要详细

应答参数	参数名称	类型	参数说明
	code	int	错误编码
	seq	int	下次请求序号（尚未实现）
	data	object 或 array	响应数据，具体根据前端需求定义
<b>示例</b>	<p><b>请求实体：</b></p> <pre>JSON</pre> <pre>1 {"uid":"test","pwd":"123456"}</pre> <p><b>应答实体：</b></p> <pre>JSON</pre> <pre>1 {"code":0,"data":{"token":"base64字符串"}}</pre>		