

## 目录

目录	1
一、文档说明	8
1、写在前面	8
BBBUG，一个开放的音乐聊天室	8
网站接入聊天室	8
1) 请求方式	8
2) 注意事项	8
2、错误代码表	9
错误代码表	9
错误代码说明	9
二、用户接口	10
1、发送邮箱验证码接口	10
一、接口说明	10
二、请求方式	10
三、请求参数	10
四、更多参数	10
五、返回数据	10
六、测试用例	10
2、用户登录接口	11
一、接口说明	11
二、请求方式	11
三、请求参数	11
四、更多参数	11
五、返回数据	11
六、测试用例	12
3、获取我的个人信息接口	13
一、接口说明	13
二、请求方式	13
三、请求参数	13
四、更多参数	13
五、返回数据	13
六、测试用例	15
4、修改我的资料接口	16
一、接口说明	16
二、请求方式	16
三、请求参数	16
四、更多参数	16
五、返回数据	16
六、测试用例	17
5、临时用户相关说明	18
临时用户介绍	18
临时用户接入	18
6、获取指定用户个人资料接口	19
一、接口说明	19
二、请求方式	19
三、请求参数	19
四、更多参数	19
五、返回数据	19
六、测试用例	20

7、禁言用户接口	21
一、接口说明	21
二、请求方式	21
三、请求参数	21
四、更多参数	21
五、返回数据	21
六、测试用例	21
8、禁止用户点歌接口	22
一、接口说明	22
二、请求方式	22
三、请求参数	22
四、更多参数	22
五、返回数据	22
六、测试用例	22
9、取消用户的禁言和禁歌	23
一、接口说明	23
二、请求方式	23
三、请求参数	23
四、更多参数	23
五、返回数据	23
六、测试用例	23
10、摸一摸用户接口	24
一、接口说明	24
二、请求方式	24
三、请求参数	24
四、更多参数	24
五、返回数据	24
六、测试用例	24
11、第三方登录接口	25
一、接口说明	25
二、请求方式	25
三、请求参数	25
四、更多参数	25
五、返回数据	25
六、测试用例	25
三、房间接口	26
1、获取热门房间接口	26
一、接口说明	26
二、请求方式	26
三、请求参数	26
四、更多参数	26
五、返回数据	26
六、测试用例	28
2、获取房间详情接口	29
一、接口说明	29
二、请求方式	29
三、请求参数	29
四、更多参数	29
五、返回数据	29
六、测试用例	30

3、获取房间在线人员列表	31
一、接口说明	31
二、请求方式	31
三、请求参数	31
四、更多参数	31
五、返回数据	31
六、测试用例	32
4、修改房间信息接口	33
一、接口说明	33
二、请求方式	33
三、请求参数	33
四、更多参数	33
五、返回数据	34
六、测试用例	34
5、获取指定房间当前正在播放音乐接口	35
一、接口说明	35
二、请求方式	35
三、请求参数	35
四、更多参数	35
五、返回数据	35
六、测试用例	36
6、获取聊天室指定房间的历史记录	37
一、接口说明	37
二、请求方式	37
三、请求参数	37
四、更多参数	37
五、返回数据	37
六、测试用例	38
7、撤回消息接口	39
一、接口说明	39
二、请求方式	39
三、请求参数	39
四、更多参数	39
五、返回数据	39
六、测试用例	39
四、歌曲接口	40
1、歌曲搜索接口	40
一、接口说明	40
二、请求方式	40
三、请求参数	40
四、更多参数	40
五、返回数据	40
六、测试用例	41
2、获取LRC歌词接口	42
一、接口说明	42
二、请求方式	42
三、请求参数	42
四、更多参数	42
五、返回数据	42
六、测试用例	43

3、歌曲播放地址跳转地址	44
一、接口说明	44
二、请求方式	44
4、管理员或房主切歌接口	45
一、接口说明	45
二、请求方式	45
三、请求参数	45
四、更多参数	45
五、返回数据	45
六、测试用例	45
5、用户点歌接口	46
一、接口说明	46
二、请求方式	46
三、请求参数	46
四、更多参数	46
五、返回数据	46
六、测试用例	46
6、指定房间当前的已点歌曲列表	47
一、接口说明	47
二、请求方式	47
三、请求参数	47
四、更多参数	47
五、返回数据	47
六、测试用例	49
7、将已点歌曲列表中指定歌曲置顶	50
一、接口说明	50
二、请求方式	50
三、请求参数	50
四、更多参数	50
五、返回数据	50
六、测试用例	50
8、管理或房主移除已点歌曲接口	51
一、接口说明	51
二、请求方式	51
三、请求参数	51
四、更多参数	51
五、返回数据	51
六、测试用例	51
9、获取我点过的歌列表接口	52
一、接口说明	52
二、请求方式	52
三、请求参数	52
四、更多参数	52
五、返回数据	52
六、测试用例	53
10、从我的歌单删除歌曲接口	54
一、接口说明	54
二、请求方式	54
三、请求参数	54
四、更多参数	54

五、返回数据	54
六、测试用例	54
五、Websocket	55
1、获取Websocket连接地址	55
一、接口说明	55
二、请求方式	55
三、请求参数	55
四、更多参数	55
五、返回数据	55
六、测试用例	56
2、发送文字消息接口	57
一、接口说明	57
二、请求方式	57
三、请求参数	57
四、更多参数	57
五、返回数据	57
六、测试用例	58
3、发送图片消息接口	59
一、接口说明	59
二、请求方式	59
三、请求参数	59
四、更多参数	59
五、返回数据	59
六、测试用例	60
4、WebSocket消息解析说明文档	61
数据说明	61
数据类型	61
六、图片接口	62
1、搜索图片表情包	62
一、接口说明	62
二、请求方式	62
三、请求参数	62
四、更多参数	62
五、返回数据	62
六、测试用例	62
2、上传图片接口	63
一、接口说明	63
二、请求方式	63
三、请求参数	63
四、更多参数	63
五、返回数据	63
六、测试用例	63
七、第三方合作接口	64
1、第三方接入聊天室使用手册与帮助文档	64
接入说明	64
接入方案	64
用户集成方案	64
临时用户说明	64
进入自己创建的房间	64
合作案例	64

更多功能	64
联系我	64
2、聊天室绑定第三方独立域名	65
一、使用平台提供的二级域名	65
二、使用独立域名（提供两种方式）	65
1. 使用http方式的域名直接解析	65
2. 使用反向代理方式接入独立域名	65
3、SVG外链和iFrame播放器外链使用说明	67
SVG房间信息外链	67
iframe播放器	67
4、合作网站	68
接入手册	68
合作站点	68
5、聊天室插件开发使用文档	69
开发说明	69
准备工作	69
API接口文档	69
1. 监听BBBUG开放的部分事件：	69
2. 调用BBBUG API：	70
首先，引入SDK：	70
接下来，你就可以使用BBBUG.xxx()调用我们提供的API啦~	70
3. 调用服务端API	71
八、下载相关	72
1、Windows客户端下载	72
客户端下载地址	72
开源说明	72
九、部署相关	73
0、写在最前面	73
最近更新	73
部署声明	74
部署步骤	74
1、部署API端步骤说明	75
架构说明	75
软件环境	75
开始部署	75
与Websocket服务数据交互部分	75
结语	75
2、部署Websocket服务端步骤	76
架构说明	76
部署步骤	76
结语	76
3、部署最后的前端部分	77
架构说明	77
部署说明	77
结语	77
再次声明	77
6、从零开始部署一个你自己的BBBUG音乐聊天室	78
本文说明	78
部署步骤	78
一、域名解析	78

二、部署API端	78
三、部署Websocket端	79
四、部署web前端	80
五、歌曲同步	81
六、邮箱配置和第三方登录等配置	81
七、其他配置	81
八、部署案例	81
Test	82

# 一、文档说明

## 1、写在前面

### BBBUG，一个开放的音乐聊天室

欢迎你与我们合作，你可以选择下面的方式：

QQ群1140258698

### 网站接入聊天室

聊天室开放第三方网站的接入，请查阅 <https://doc.bbbug.com/2811812.html>

聊天室支持了第三方独立域名绑定与系统二级域名绑定，请查阅 <https://doc.bbbug.com/2883813.html>

聊天室开放了所有的API，你可以自由构建对接一个自己的APP、小程序或网站，具体请查阅下方：

#### 1) 请求方式

- 本项目API接口除上传文件之外的所有POST请求统一采用JSON进行数据通信，所有数据交互时请严格在JSON相关约束下进行。
- Content-Type默认使用 application/application ，上传文件时使用 multipart/form-data
- 项目涉及到的错误代码默认 200 为成功，其他请参照 错误代码表 或查看返回的 msg 字段
- 项目涉及到的请求参数必须携带 plat=\*\*\* version=\*\*\* 两个标准公共参数。

#### 2) 注意事项

- API根路径为 <http://api.bbbug.com/api/> ，请求API时追加API地址即可。
- 图片等附件根路径为 <http://api.bbbug.com/uploads/> ，请求时追加相对路径 \*\*\*\*\* 即可。



## 2、错误代码表

### 错误代码表

code	说明
200	正常请求,操作成功
500	错误请求,操作失败
400	缺少参数
401	需要登录
403	无权限操作
404	接口不存在
503	访问的接口不可用
301	APP需要强制更新
302	APP弹出更新不强制

### 错误代码说明

以上仅列出部分常见错误代码，未提及的错误可具体查看返回的msg字段。

## 二、用户接口

### 1、发送邮箱验证码接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 00:30:56

#### 一、接口说明

请注意，验证码有效时间为10分钟，如果用户没收到，提醒用户查看回收站，有可能被ban了。

#### 二、请求方式

POST <https://api.bbbug.com/api/sms/email>

#### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
email	string	是	admin@hamm.cn	用户邮箱

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "email": "admin@hamm.cn"
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	500	-
msg	string	是	邮件发送过于频繁,请稍后再试	-

示例返回结果：

```
{
  "code": 500,
  "msg": "邮件发送过于频繁,请稍后再试"
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/70db6292e72eea93632f63c3331b73759467a8c8>

## 2、用户登录接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 00:34:45

### 一、接口说明

请注意，本接口支持ID/邮箱+密码登录，也同时支持使用邮箱验证码登录，请直接将验证码传入 `user_password` 字段即可。

### 二、请求方式

POST <https://api.bbbug.com/api/user/login>

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
user_account	string	是	<a href="mailto:admin@hamm.cn">admin@hamm.cn</a>	邮箱或者ID
user_password	string	是	123456	密码或者验证码
user_device	string	否	iPad	设备

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "user_device": "iPad",
  "user_account": "email",
  "user_password": "password"
}
```

`user_device` 可取的值目前有：Windows、iPad、iPhone、Ubuntu、Android、MacOS

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	登录成功	-
data	object	是	{}	-
data.access_token	string	是	abcdefghijklmn	-

示例返回结果：

```
{
  "code": 200,
  "msg": "登录成功",
  "data": {
    "access_token": "abcdefghijklmn"
  }
}
```

二、用户接口

六、测试用例

<https://tester.hamm.cn/#/20200921/0e43bfd1f4fc86bdd6fd2d692013f88fe70c5c31>

## 二、用户接口

### 3、获取我的个人信息接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 00:36:19

#### 一、接口说明

你可以在这里对接口进行一些简单的描述

#### 二、请求方式

POST <https://api.bbbug.com/api/user/getMyInfo>

#### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	abedefghijkl	登录后的身份凭证
plat	string	是	pc	-
version	number	是	10000	-

示例JSON：

```
{
  "access_token": "abedefghijkl",
  "plat": "pc",
  "version": 10000
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是		-
data	object	是	{}	-
data.user_id	number	是	10000	用户ID
data.user_parent	number	是	0	邀请人ID
data.user_sex	number	是	1	性别(1男0女)
data.user_account	string	是	admin@hamm.cn	邮箱
data.user_touchtip	string	是	帅气的脸庞	摸一摸文案
data.user_name	string	是	Hamm	昵称
data.user_head	string	是	头像URL	头像URL
data.user_remark	string	是	用户签名	用户签名
data.user_group	number	是	5	用户组1管理组5普通组
data.user_song	number	是	37	点歌数量
data.user_img	number	是	40	发图数量
data.user_chat	number	是	342	聊天数量

## 二、用户接口

data.user_gamesongscore	number	是	0	猜歌得分
data.user_app	number	是	1	所属应用1为内部应用
data.user_device	string	是	MacOS	登录设备
data.app_id	number	是	1	所属应用
data.app_name	string	是	BBBUG	应用名称
data.app_url	string	是	<a href="https://bbbug.com">https://bbbug.com</a>	应用主页
data.app_user	number	是	1	应用所有人
data.app_scope	string	是		应用权限
data.app_status	number	是	0	状态1禁用
data.myRoom	object	是	{}	我的房间,null为未创建
data.myRoom.room_id	number	是	888	房间ID
data.myRoom.room_user	number	是	10000	房间所有人
data.myRoom.room_online	number	是	3	在线人数
data.myRoom.room_name	string	是	BBBUG点歌音乐房	房间名称
data.myRoom.room_type	number	是	1	房间类型,见房间类型说明
data.myRoom.room_password	string	是		房间是否需要密码
data.myRoom.room_notice	string	是	房间公告	房间公告
data.myRoom.room_order	number	是	1000	排序
data.myRoom.room_reason	string	是	封禁原因	封禁原因
data.myRoom.room_status	number	是	0	状态1被封禁
data.user_admin	boolean	是	false	是否管理员
data.user_needmotify	boolean	是	false	是否需要提示修改信息

示例返回结果：

```
{
  "code": 200,
  "msg": "",
  "data": {
    "user_id": 10000,
    "user_parent": 0,
    "user_sex": 1,
    "user_account": "admin@hamm.cn",
    "user_name": "Hamm",
    "user_touchtip": "帅气的脸庞",
    "user_head": "https://api.bbbug.com/uploads",
    "user_remark": "欢迎来BBBUG聊天听歌划水呀~",
    "user_group": 5,
    "user_song": 37,
    "user_img": 40,
    "user_chat": 342,
    "user_gamesongscore": 0,
    "user_app": 1,
    "user_device": "MacOS",
    "app_id": 1,
    "app_name": "BBBUG",
    "app_url": "https://bbbug.com",
    "app_user": 1,
    "app_scope": "",
    "app_status": 0,
    "myRoom": {
      "room_id": 888,
      "room_user": 10000,
```

## 二、用户接口

```
"room_online": 3,  
"room_name": "BBBUG点歌音乐房",  
"room_type": 1,  
"room_password": "",  
"room_notice": "欢迎来到BBBUG，点歌系统升级完成，欢迎来点歌呀",  
"room_order": 10000000,  
"room_reason": "",  
"room_status": 0  
},  
"user_admin": false,  
"user_needmotify": false  
}  
}
```

## 六、测试用例

<https://tester.hamm.cn/#/20200904/4ad229addc934c6ed5aed1f99ef19faffd09d693>

## 二、用户接口

### 4、修改我的资料接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 10:10:49

#### 一、接口说明

修改我的资料，如需同时修改密码，则传入user\_password，否则请传入null。

#### 二、请求方式

POST <https://api.bbbug.com/api/user/updateMyInfo>

#### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
user_name	string	是	Hamm	昵称
user_head	string	是	<a href="https://027872a.png">https://027872a.png</a>	头像URL
user_remark	string	是	欢迎来BBBUG聊天听歌划水呀~	用户签名
user_sex	number	是	1	性别1男0女
user_password	string	是	null/123456	新密码
user_touchtip	string	是	帅气的脸庞	摸一摸文案

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "user_name": "Hamm",
  "user_head": "https://872a.png",
  "user_remark": "欢迎来BBBUG聊天听歌划水呀~",
  "user_touchtip": "帅气的脸庞",
  "user_sex": 1,
  "user_password": ""
}
```

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "user_name": "Hamm",
  "user_head": "https://872a.png",
  "user_remark": "欢迎来BBBUG聊天听歌划水呀~",
  "user_sex": 1,
  "user_password": "123456"
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据



## 二、用户接口

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{  
  "code": 401,  
  "msg": "登录过期，请重新登录"  
}
```

## 六、测试用例

<https://tester.hamm.cn/#/20200904/9255512ac09325e975188b5a248783bcd1ace695>

## 5、临时用户相关说明

### 临时用户介绍

临时用户指用户无需登录即可进入房间，可听歌看别人发言，但自己无任何权限。

### 临时用户接入

临时用户使用固定的access\_token接入：

```
45af3cfe44942c956e026d5fd58f0feffbd3a237
```

## 6、获取指定用户个人资料接口

本文档由 **Tester** 自动生成，最后修改时间 2020-09-04 11:19:14

### 一、接口说明

根据ID获取用户个人信息接口

### 二、请求方式

POST <https://api.bbbug.com/api/user/getUserInfo>

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
user_id	number	是	1	-

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "user_id": 1
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-
data	object	是	{}	用户对象
data.user_id	number	是	1	用户ID
data.user_song	number	是	253	点歌数
data.user_chat	number	是	32	聊天数
data.user_img	number	是	1	发图数
data.user_sex	number	是	0	1男0女
data.user_device	string	是	Windows	登录设备
data.user_gamesongscore	number	是	3	猜歌得分
data.user_name	string	是	Hamm	用户昵称
data.user_head	string	是	<a href="https://b74931565b.jpg">https://b74931565b.jpg</a>	头像
data.user_remark	string	是	多钱我都去玩	签名
data.app_id	number	是	1	应用ID
data.app_name	string	是	BBBUG	应用名称

## 二、用户接口

data.app_url	string	是	https://bbbug.com	应用地址
data.user_admin	boolean	是	true	是否管理员
data.app_status	number	是	0	应用状态1禁用
data.push_count	number	是	2	无敌顶歌次数
data.pass_count	number	是	0	无敌切歌次数

示例返回结果：

```
{
  "code": 200,
  "msg": "success",
  "data": {
    "user_id": 1,
    "user_song": 253,
    "user_chat": 32,
    "user_img": 1,
    "user_sex": 0,
    "user_device": "Windows",
    "user_gamesongscore": 3,
    "user_name": "Hamm",
    "user_head": "https://4931565b.jpg",
    "user_remark": "当前为第",
    "app_id": 1,
    "app_name": "BBBUG",
    "app_url": "https://bbbug.com",
    "user_admin": true,
    "app_status": 0,
    "push_count": 2,
    "pass_count": 0
  }
}
```

## 六、测试用例

<https://tester.hamm.cn/#/20200904/d4c79cc8b23534c2a1faf03e8df94c99a9337acf>

## 二、用户接口

### 7、禁言用户接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-15 23:34:22

#### 一、接口说明

此接口可禁止用户发言。

#### 二、请求方式

POST `https://api.bbbug.com/api/user/shutdown`

#### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
user_id	number	是	10000	-
room_id	number	是	888	-

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "user_id": 10000,
  "room_id": 888
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	禁止发言成功!	-

示例返回结果：

```
{
  "code": 200,
  "msg": "禁止发言成功!"
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200915/6ef838c465e6e51e6e7e3f44b45b25ffed01d32f>

## 8、禁止用户点歌接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-15 23:35:47

### 一、接口说明

此接口可禁止用户点歌

### 二、请求方式

POST <https://api.bbbug.com/api/user/songdown>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
user_id	number	是	10000	-
room_id	number	是	888	-

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "user_id": 10000,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	禁止点歌成功!	-

示例返回结果：

```
{
  "code": 200,
  "msg": "禁止点歌成功!"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200915/a936f8989987d103dd8490cfd0c4f61e7c9ffc01>

## 9、取消用户的禁言和禁歌

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-15 23:37:15

### 一、接口说明

此接口可取消用户的禁止发言与禁止点歌

### 二、请求方式

POST <https://api.bbbug.com/api/user/removeban>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
user_id	number	是	10000	-
room_id	number	是	888	-

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "user_id": 10000,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200915/9909ae9f495436f2a8f0ea6b55e47f4992fdd3f2>

## 10、摸一摸用户接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-10-22 17:05:45

### 一、接口说明

调用此接口可发起摸一摸

### 二、请求方式

POST <https://api.bbbug.com/api/message/touch>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
at	number	是	10000	被摸的用户ID
room_id	number	是	888	房间ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "at": 10000,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20201022/63dcc553be2ddadd6b1af48588856cd2b85ec7cc>



## 11、第三方登录接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-12-02 23:07:36

### 一、接口说明

首先调用第三方oauth2.0的oauth接口，所需要的appid请 [查看这里](#)

第三方授权回调后，拦截得到的code，请求此接口获得登录用户的access\_token

### 二、请求方式

POST <https://api.bbbug.com/api/user/thirdlogin>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是		-
plat	string	是	vue	-
version	number	是	10000	-
from	string	是	qq	-
code	string	是	第三方的code	-

示例JSON：

```
{
  "access_token": "",
  "plat": "vue",
  "version": 10000,
  "from": "qq",
  "code": "第三方的code"
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	500	-
msg	string	是	使用QQ账号登录失败,请重试	-

示例返回结果：

```
{
  "code": 500,
  "msg": "使用QQ账号登录失败,请重试"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20201202/82ffd9c2efe52cde61dafa91dafc707d5b9d92c>

## 三、房间接口

### 1、获取热门房间接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 00:42:40

#### 一、接口说明

你可以在这里对接口进行一些简单的描述

#### 二、请求方式

POST `https://api.bbbug.com/api/room/hotRooms`

#### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
access_token	string	是	123456	身份令牌

示例JSON：

```
{
  "plat": "pc",
  "access_token": "123456",
  "version": 10000
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	数据获取成功	-
data	array	是	[]	-
data.room_id	number	是	888	房间ID
data.room_user	number	是	10000	房间所有人
data.room_score	number	是	2	房间积分
data.room_online	number	是	2	在线人数
data.room_name	string	是	BBBUG点歌音乐房	房间名称
data.room_type	number	是	1	房间类型,见房间类型说明
data.room_public	number	是	1	1需要密码
data.room_notice	string	是	欢迎来到BBBUG	房间公告
data.room_order	number	是	10000000	房间排序
data.room_addsong	number	是	1	1仅房主可点歌

### 三、房间接口

data.room_sendmsg	number	是	0	1全员禁言0正常
data.room_robot	number	是	0	1打开机器人点歌
data.room_huya	string	是	<a href="https://www.huya.com/1234">https://www.huya.com/1234</a>	虎牙直播地址
data.room_reason	string	是	违规	封禁原因
data.room_status	number	是	0	状态1封禁
data.user_id	number	是	10000	房间所有人ID
data.user_name	string	是	Hamm	所有人昵称
data.user_admin	boolean	是	false	所有人是否管理员

示例返回结果：

```
{
  "code": 200,
  "msg": "数据获取成功",
  "data": [
    {
      "room_id": 888,
      "room_user": 10000,
      "room_online": 2,
      "room_score": 123,
      "room_name": "BBBUG点歌音乐房",
      "room_type": 1,
      "room_public": 0,
      "room_notice": "欢迎来到BBBUG，点歌系统升级完成，欢迎来点歌呀",
      "room_order": 10000000,
      "room_addsong": 1,
      "room_sendmsg": 0,
      "room_robot": 0,
      "room_huya": 'https://huya.com/1234',
      "room_reason": "",
      "room_status": 0,
      "user_id": 10000,
      "user_name": "Hamm",
      "user_admin": false
    },
    {
      "room_id": 6666,
      "room_user": 10007,
      "room_online": 0,
      "room_score": 123,
      "room_name": "听歌猜歌游戏房",
      "room_type": 2,
      "room_public": 0,
      "room_notice": "欢迎来猜歌，请等待上一局游戏结束后参与!",
      "room_order": 800000,
      "room_addsong": 1,
      "room_sendmsg": 0,
      "room_robot": 0,
      "room_huya": 'https://huya.com/1234',
      "room_reason": "",
      "room_status": 0,
      "user_id": 10007,
      "user_name": "猜歌小伙",
      "user_admin": false
    },
    {
      "room_id": 999,
      "room_user": 1,
      "room_online": 0,
      "room_score": 123,
      "room_name": "开发者房间",
      "room_type": 1,
      "room_public": 1,
      "room_notice": "这里是开发者房间，讨论项目开发等事情",
    }
  ]
}
```

### 三、房间接口

```
"room_order": 1001,  
"room_addsong": 1,  
"room_sendmsg": 0,  
"room_robot": 0,  
"room_huya": 'https://huya.com/1234',  
"room_reason": "",  
"room_status": 0,  
"user_id": 1,  
"user_name": "机器人",  
"user_admin": true  
  }  
]  
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200915/c46fc41ce0a28bd64c884f28b70e547bf989a63f>

## 2、获取房间详情接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 00:44:39

### 一、接口说明

你可以在这里对接口进行一些简单的描述

### 二、请求方式

POST `https://api.bbbug.com/api/room/getRoomInfo`

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
access_token	string	是	123456	身份令牌
version	number	是	10000	-
room_id	number	是	888	房间ID
room_password	null	是	null	房间密码,无需加密请填入空

示例JSON :

```
{
  "plat": "pc",
  "access_token": "123456",
  "version": 10000,
  "room_id": 888,
  "room_password": null
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	数据加载成功	-
data	object	是	{}	-
data.room_id	number	是	888	-
data.room_user	number	是	10000	所属人ID
data.room_online	number	是	2	在线人数
data.room_score	number	是	2	房间积分
data.room_name	string	是	BBBUG点歌音乐房	房间名称
data.room_type	number	是	1	房间类型
data.room_notice	string	是	欢迎来到BBBUG	房间公告
data.room_order	number	是	10000000	房间排序
data.room_addsong	number	是	1	1仅房主点歌

### 三、房间接口

data.room_sendmsg	number	是	0	1全员禁言
data.room_robot	number	是	0	1打开机器人点歌
data.room_huya	string	是	<a href="https://www.huya.com/1234">https://www.huya.com/1234</a>	虎牙直播地址
data.room_reason	string	是	-	封禁原因
data.room_status	number	是	0	状态1封禁
data.room_public	number	是	0	1为需要密码

示例返回结果：

```
{
  "code": 200,
  "msg": "数据加载成功",
  "data": {
    "room_id": 888,
    "room_user": 10000,
    "room_online": 2,
    "room_score": 123,
    "room_name": "BBBUG点歌音乐房",
    "room_type": 1,
    "room_notice": "欢迎来到BBBUG，点歌系统升级完成，欢迎来点歌呀",
    "room_order": 10000000,
    "room_addsong": 1,
    "room_sendmsg": 0,
    "room_robot": 0,
    "room_huya": 'https://huya.com/1234',
    "room_reason": "",
    "room_status": 0,
  }
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200915/333f7a45aed78dce54a7862f1b50950f987e30ab>

### 3、获取房间在线人员列表

本文档由 **Tester** 自动生成，最后修改时间 2020-09-04 09:23:13

#### 一、接口说明

主动请求在线人员接口

#### 二、请求方式

POST <https://api.bbbug.com/api/user/online>

#### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	-

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "room_id": 888
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-
data	array	是	[]	-
data.user_id	number	是	10032	用户ID
data.user_name	string	是	Hamm	用户昵称
data.user_head	string	是	头像地址	用户头像地址
data.user_group	number	是	5	用户组1管理5普通用户
data.user_remark	string	是	每个人都应该有签名	签名
data.user_device	string	是	Windows	登录设备
data.user_sex	number	是	1	性别1男0女
data.app_id	number	是	1000	所属应用
data.app_name	string	是	浑水摸鱼	应用名称
data.app_url	string	是	<a href="https://hunshuimoyu.cn">https://hunshuimoyu.cn</a>	应用地址
data.user_admin	boolean	是	false	是否管理
data.user_shutdown	boolean	是	false	是否被禁言

### 三、房间接口

data.user_songdown	boolean	是	false	是否被禁歌
--------------------	---------	---	-------	-------

示例返回结果：

```
{
  "code": 200,
  "msg": "success",
  "data": [
    {
      "user_id": 10000,
      "user_name": "Hamm",
      "user_head": "659edf8ae2af9743b6f5922bb027872a.png",
      "user_group": 5,
      "user_remark": "欢迎来BBBUG聊天听歌划水呀~",
      "user_device": "MacOS",
      "user_sex": 1,
      "app_id": 1,
      "app_name": "BBBUG",
      "app_url": "https://bbbug.com",
      "user_admin": false,
      "user_shutdown": false,
      "user_songdown": false
    }
  ]
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200904/534af9d7a23f6f00f3938bd11fc0d3bb17c8e2bf>



## 4、修改房间信息接口

本文档由 **Tester** 自动生成，最后修改时间 2020-09-09 20:47:49

### 一、接口说明

这里可以修改房间的信息

### 二、请求方式

POST <https://api.bbbug.com/api/room/saveMyRoom>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	abcdefg	-
plat	string	是	pc	-
version	number	是	10000	-
room_name	string	是	BBBUG点歌音乐房	-
room_notice	string	是	当前为夜间电台听歌模式，点歌请明早9:00来呀	-
room_type	number	是	1	-
room_addsong	number	是	1	-
room_sendmsg	number	是	0	-
room_robot	number	是	0	-
room_id	number	是	888	-
room_public	string	是	1	1加密房间
room_password	string	否	123456	public为1时必传
room_huya	string	否	<a href="https://www.huya.com/1234">https://www.huya.com/1234</a>	虎牙地址,room_type=5时必传
room_voteperpass	int	否	1	是否开启投票点歌
room_votepercent	int	否	30	投票比例

示例JSON：

```
{
  "access_token": "abcdefg",
  "plat": "pc",
  "version": 10000,
  "room_name": "BBBUG点歌音乐房",
  "room_notice": "当前为夜间电台听歌模式，点歌请明早9:00来呀",
  "room_type": 1,
  "room_public": 1,
  "room_password": "123456",
  "room_huya": "https://www.huya.com/1234",
  "room_voteperpass": 1,
  "room_votepercent": 30,
  "room_addsong": 1,
  "room_sendmsg": 0,
  "room_robot": 0,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 三、房间接口



### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	房间信息修改成功	-

示例返回结果：

```
{  
  "code": 200,  
  "msg": "房间信息修改成功"  
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200909/eec6002ba021113a7005fe6d54cb5a700802e7f9>

## 5、获取指定房间当前正在播放音乐接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-16 10:02:03

### 一、接口说明

你可以使用这个接口获取指定房间正在播放的音乐

### 二、请求方式

POST <https://api.bbbug.com/api/song/now>

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
room_id	string	是	888	房间ID

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "room_id": "888"
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
type	string	是	playSong	类型
time	string	是	10:00:04	无意义
user	object	是	{}	点歌人信息
user.user_id	number	是	10000	-
user.user_song	number	是	505	-
user.user_chat	number	是	2929	-
user.user_img	number	是	352	-
user.user_sex	number	是	1	-
user.user_extra	string	是		-
user.user_device	string	是	MacOS	-
user.user_gamesongscore	number	是	11	-
user.user_name	string	是	Hamm	-
user.user_head	string	是	<a href="https://a5fd159ad9dc.PNG">https://a5fd159ad9dc.PNG</a>	-
user.user_remark	string	是	欢迎来BBBUG听歌呀	-
user.app_id	number	是	1	-

### 三、房间接口

user.app_name	string	是	BBBUG	-
user.app_url	string	是	<a href="https://bbbug.com">https://bbbug.com</a>	-
user.user_admin	boolean	是	false	-
song	object	是	{}	歌曲信息
song.mid	number	是	27866905	-
song.name	string	是	Schnappi Das Kleine Krokodil	-
song.pic	string	是	<a href="http://40893.jpg">http://40893.jpg</a>	-
song.length	number	是	129	-
song.singer	string	是	Schnappi	-
at	boolean	是	false	送给谁
since	number	是	1600221602	开始播放时间
now	number	是	1600221604	当前时间

示例返回结果：

```
{
  "type": "playSong",
  "time": "10:00:04",
  "user": {
    "user_id": 10000,
    "user_song": 505,
    "user_chat": 2929,
    "user_img": 352,
    "user_sex": 1,
    "user_extra": "",
    "user_device": "MacOS",
    "user_gamesongscore": 11,
    "user_name": "Hamm",
    "user_head": "https://api.bbbug.com/9dc.PNG",
    "user_remark": "欢迎来BBBUG听歌呀",
    "app_id": 1,
    "app_name": "BBBUG",
    "app_url": "https://bbbug.com",
    "user_admin": false
  },
  "song": {
    "mid": 27866905,
    "name": "Schnappi Das Kleine Krokodil",
    "pic": "http://3440893.jpg",
    "length": 129,
    "singer": "Schnappi"
  },
  "at": false,
  "since": 1600221602,
  "now": 1600221604
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200916/215e95d25b6b3646d027ce70f4b2a851281500e7>

## 6、获取聊天室指定房间的历史记录

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-22 11:17:03

### 一、接口说明

你可以通过此接口获取聊天室指定房间的历史记录

### 二、请求方式

POST <https://api.bbbug.com/api/message/getMessageList>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	房间ID
page	number	否	1	页码 默认1
per_page	number	否	100	每页数量 默认100

示例JSON :

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "room_id": 888,
  "page": 1,
  "per_page": 100,
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-
data	array	是	[]	-
data.message_id	number	是	182	-
data.message_user	number	是	10004	-
data.message_type	string	是	text	-
data.message_where	string	是	channel	-
data.message_to	string	是	888	-
data.message_content	string	是	消息JSON字符串	-
data.message_status	number	是	0	-

### 三、房间接口

data.message_createtime	number	是	1600743474	-
data.message_updatetime	number	是	1600743474	-

示例返回结果：

```
{
  "code": 200,
  "msg": "success",
  "data": [
    {
      "message_id": 172,
      "message_user": 10000,
      "message_type": "text",
      "message_where": "channel",
      "message_to": "888",
      "message_content": "这是段消息JSON字符串",
      "message_status": 0,
      "message_createtime": 1600742846,
      "message_updatetime": 1600742846
    },
    {
      "message_id": 171,
      "message_user": 10164,
      "message_type": "text",
      "message_where": "channel",
      "message_to": "888",
      "message_content": "这是段消息JSON字符串",
      "message_status": 0,
      "message_createtime": 1600742830,
      "message_updatetime": 1600742830
    }
  ]
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200923/c53a2299a48d3a6a3c6cdd00108b69f68bef7f36>

## 7、撤回消息接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-23 00:15:11

### 一、接口说明

你可以通过请求此接口撤回一条已发送的消息。

### 二、请求方式

POST `https://api.bbbug.com/api/message/back`

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
message_id	string	是	12	消息的message_id
room_id	number	是	999	房间ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "message_id": "12",
  "room_id": 999
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200923/e9fa2fa10fd03b495a38582c4525374248de75b7>

## 四、歌曲接口

### 1、歌曲搜索接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 09:37:11

#### 一、接口说明

可根据歌手、歌曲等进行歌曲的搜索

#### 二、请求方式

POST <https://api.bbbug.com/api/song/search>

#### 三、请求参数

字段	类型	必填	示例	说明
keyword	string	是	周杰伦	-
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-

示例JSON：

```
{
  "keyword": "周杰伦",
  "access_token": "123456",
  "plat": "pc",
  "version": 10000
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是		-
data	array	是	[]	歌曲对象数组,后边会用到
data.mid	number	是	142655450	歌曲ID
data.name	string	是	Mojito	歌曲名称
data.pic	string	是	<a href="http://309391875.jpg">http://309391875.jpg</a>	歌曲头图
data.length	number	是	185	歌曲播放秒数
data.singer	string	是	周杰伦	歌手名称

示例返回结果：

```
{
  "code": 200,
  "msg": ""
}
```



#### 四、歌曲接口

```
"data": [  
  {  
    "mid": 142655450,  
    "name": "Mojito",  
    "pic": "http://img3.kuwo.cn/star/albumcover/300/48/47/309391875.jpg",  
    "length": 185,  
    "singer": "周杰伦"  
  },  
  {  
    "mid": 228908,  
    "name": "晴天",  
    "pic": "http://img2.kuwo.cn/star/albumcover/300/64/96/2266534336.jpg",  
    "length": 269,  
    "singer": "周杰伦"  
  },  
  {  
    "mid": 7149583,  
    "name": "告白气球",  
    "pic": "http://img2.kuwo.cn/star/albumcover/300/64/39/3540704654.jpg",  
    "length": 215,  
    "singer": "周杰伦"  
  },  
  {  
    "mid": 140064959,  
    "name": "稻香",  
    "pic": "http://img1.kuwo.cn/star/starheads/300/10/6/294045140.jpg",  
    "length": 198,  
    "singer": "周杰伦&林俊杰"  
  }  
]
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200922/10f5ba58efed41b91431d89646a9f290ced6e7f9>

## 2、获取LRC歌词接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:41:01

### 一、接口说明

根据歌曲ID查询歌曲对应的LRC歌词

### 二、请求方式

POST <https://api.bbbug.com/api/song/getLrc>

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
mid	number	是	148330690	歌曲ID,由websocket下发

示例JSON :

```
{
  "plat": "pc",
  "version": 10000,
  "mid": 148330690
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是		-
data	array	是	[]	-
data.lineLyric	string	是	人为情叹 - 阿悠悠	LRC歌词
data.time	string	是	1.47	LRC歌词开始时间(秒)

示例返回结果 :

```
{
  "code": 200,
  "msg": "",
  "data": [
    {
      "lineLyric": "人为情叹 - 阿悠悠 ",
      "time": "1.47"
    },
    {
      "lineLyric": "我走在城南的关",
      "time": "67.27"
    },
    {
      "lineLyric": "人为情而月下感叹",
      "time": "71.78"
    }
  ]
}
```

#### 四、歌曲接口

```
  },
  {
    "lineLyric": "倒不如不曾遇见",
    "time": "75.53"
  },
  {
    "lineLyric": "天各两端",
    "time": "78.03"
  },
  {
    "lineLyric": "我走在城南的关",
    "time": "197.53"
  },
  {
    "lineLyric": "看着远小的帆",
    "time": "199.78"
  },
  {
    "lineLyric": "人为情而月下感叹",
    "time": "201.77"
  },
  {
    "lineLyric": "一圈一圈紧紧捆在 心的左边",
    "time": "239.02"
  }
]
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/82cfec9ae690d5381739cedf7929e6c3642e1c7>

### 3、歌曲播放地址跳转地址

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:45:01

#### 一、接口说明

歌曲播放地址，无需请求，服务端将自动进行301跳转。

#### 二、请求方式

```
GET https://api.bbbug.com/api/song/playurl?mid=148330690
```

## 4、管理员或房主切歌接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:57:07

### 一、接口说明

管理员或房主切歌接口

### 二、请求方式

POST <https://api.bbbug.com/api/song/pass>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	房间ID
mid	number	是	381027	歌曲ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "room_id": 888,
  "mid": 381027,
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200910/0b9063f6e71d238ef8720b24da1ca39e264964c3>

## 5、用户点歌接口

本文档由 **Tester** 自动生成，最后修改时间 2020-09-04 09:59:20

### 一、接口说明

用户点歌接口

### 二、请求方式

POST <https://api.bbbug.com/api/song/addSong>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
mid	number	是	381027	歌曲ID
at	number	是	10000	送给谁
room_id	number	是	888	房间ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "mid": 4270206,
  "at": 10000,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200910/4326bff577fc134abd742d178e471075664250d5>

## 6、指定房间当前的已点歌曲列表

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 10:00:37

### 一、接口说明

获取指定房间的已点歌曲列表接口

### 二、请求方式

POST `https://api.bbbug.com/api/song/songList`

### 三、请求参数

字段	类型	必填	示例	说明
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	房间ID

示例JSON：

```
{
  "plat": "pc",
  "version": 10000,
  "room_id": 888
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是		-
data	array	是	[]	点歌人对象数组
data.user	object	是	{}	点歌人对象
data.user.user_id	number	是	10000	点歌人ID
data.user.user_song	number	是	37	点歌人点歌数
data.user.user_chat	number	是	346	点歌人聊天数
data.user.user_img	number	是	43	点歌人发图数
data.user.user_sex	number	是	1	点歌人性别1男0女
data.user.user_device	string	是	MacOS	点歌人设备
data.user.user_gamesongscore	number	是	0	点歌人猜歌得分
data.user.user_name	string	是	Hamm	点歌人昵称
data.user.user_head	string	是	<a href="https://659edf8ae2af9743b6f5922bb027872a.png">https://659edf8ae2af9743b6f5922bb027872a.png</a>	点歌人头像
data.user.user_remark	string	是	我是房主	点歌人签名
data.user.app_id	number	是	1	点歌人所属应用

#### 四、歌曲接口

data.user.app_name	string	是	BBBUG	点歌人所属应用名称
data.user.app_url	string	是	<a href="https://bbbug.com">https://bbbug.com</a>	点歌人应用地址
data.user.user_admin	boolean	是	false	点歌人是否管理
data.song	object	是	{}	歌曲对象数组
data.song.mid	number	是	4270206	歌曲ID
data.song.name	string	是	下个路口见	歌曲名称
data.song.pic	string	是	<a href="http://3035297292.jpg">http://3035297292.jpg</a>	歌曲头图
data.song.length	number	是	210	歌曲长度
data.song.singer	string	是	李宇春	歌手

示例返回结果：

```
{
  "code": 200,
  "msg": "",
  "data": [
    {
      "user": {
        "user_id": 10000,
        "user_song": 37,
        "user_chat": 346,
        "user_img": 43,
        "user_sex": 1,
        "user_device": "MacOS",
        "user_gamesongscore": 0,
        "user_name": "Hamm",
        "user_head": "https://659edf8ae2af9743b6f5922bb027872a.png",
        "user_remark": "我是房主",
        "app_id": 1,
        "app_name": "BBBUG",
        "app_url": "https://bbbug.com",
        "user_admin": false
      },
      "song": {
        "mid": 4270206,
        "name": "下个路口见",
        "pic": "http://3035297292.jpg",
        "length": 210,
        "singer": "李宇春"
      }
    },
    {
      "user": {
        "user_id": 10000,
        "user_song": 37,
        "user_chat": 346,
        "user_img": 43,
        "user_sex": 1,
        "user_device": "MacOS",
        "user_gamesongscore": 0,
        "user_name": "Hamm",
        "user_head": "https://659edf8ae2af9743b6f5922bb027872a.png",
        "user_remark": "我是房主",
        "app_id": 1,
        "app_name": "BBBUG",
        "app_url": "https://bbbug.com",
        "user_admin": false
      },
      "song": {
        "mid": 4270206,
        "name": "下个路口见",
        "pic": "http://3035297292.jpg",
        "length": 210,

```



#### 四、歌曲接口

```
    "singer": "李宇春"  
  }  
}  
]  
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/a88e8f587cdb33170e7562f45013dcb01a390977>

## 7、将已点歌曲列表中指定歌曲置顶

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 10:05:30

### 一、接口说明

置顶已点列表中选中的歌曲

### 二、请求方式

POST <https://api.bbbug.com/api/song/push>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	房间号
mid	number	是	4270206	歌曲ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "room_id": 888,
  "mid": 40098317,
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200910/82a440c4db298e5ded6d9c3dcedafd7c49ed6b56>

## 8、管理或房主移除已点歌曲接口

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 10:08:37

### 一、接口说明

管理员或房主移除指定的已点歌曲接口

### 二、请求方式

POST <https://api.bbbug.com/api/song/remove>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	房间号
mid	number	是	4270206	歌曲ID

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "room_id": 888,
  "mid": 228908,
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	401	-
msg	string	是	登录过期，请重新登录	-

示例返回结果：

```
{
  "code": 401,
  "msg": "登录过期，请重新登录"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200910/899e790dca1ef6dffe64035f57efc690b9dd253c>

## 9、获取我点过的歌列表接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 17:49:05

### 一、接口说明

获取自己最常点的50首歌，按播放次数desc排序。

### 二、请求方式

POST <https://api.bbbug.com/api/song/mySongList>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-
data	array	是	[]	歌曲对象数组
data.mid	number	是	75916	歌曲ID
data.name	string	是	蒙娜丽莎的眼泪	歌名
data.singer	string	是	林志炫	歌手
data.played	number	是	1	播放次数
data.length	number	是	165	歌曲长度
data.pic	string	是	<a href="http://abc.jpg">http://abc.jpg</a>	歌曲头图

示例返回结果：

```
{
  "code": 200,
  "msg": "success",
  "data": [
    {
      "mid": 75916,
      "name": "蒙娜丽莎的眼泪",
      "singer": "林志炫",
    }
  ]
}
```

#### 四、歌曲接口

```
"played": 1,  
"length": 165,  
"pic": "http://abc.jpg"  
},  
{  
  "mid": 65725181,  
  "name": "Got 2 B Erotic",  
  "singer": "Jmsn",  
  "played": 1,  
  "length": 165,  
  "pic": "http://abc.jpg"  
}  
]  
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/a170a872b014ef664f8b4aaa6b26b2e302d8ab7c>

## 10、从我的歌单删除歌曲接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-19 10:51:45

### 一、接口说明

通过调用此接口从我的歌单删除歌曲。

### 二、请求方式

POST <https://api.bbbug.com/api/song/deleteMySong>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	1234567	-
plat	string	是	pc	-
version	number	是	10000	-
room_id	number	是	888	-
mid	number	是	103151	音乐的MID

示例JSON：

```
{
  "access_token": "1234567",
  "plat": "pc",
  "version": 10000,
  "room_id": 888,
  "mid": 103151
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	删除歌单的歌曲成功	-

示例返回结果：

```
{
  "code": 200,
  "msg": "删除歌单的歌曲成功"
}
```

### 六、测试用例

<https://tester.hamm.cn/#/20200919/c7f0c71ff87f6a6c8c31b24f8b429506db1cb32a>

## 五、Websocket

### 1、获取Websocket连接地址

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:30:03

#### 一、接口说明

通过本接口获取Websocket连接的部分参数，并组装成为最终的Websocket连接地址。

#### 二、请求方式

POST <https://api.bbbug.com/api/room/getWebsocketUrl>

#### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	你的身份凭证	你的身份凭证
plat	string	是	pc	-
version	number	是	10000	-
channel	number	是	888	房间号,传入room_id
password	number	是	123456	房间密码,如无密码则留空

示例JSON：

```
{
  "access_token": "1234567",
  "plat": "pc",
  "version": 10000,
  "channel": 888
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-
data	object	是	{}	-
data.account	number	是	10000	用户ID,同user_id
data.channel	number	是	888	房间号,同room_id
data.ticket	string	是	验证凭据	-

示例返回结果：

```
{
  "code": 200,
  "msg": "success",
  "data": {
    "account": 10000,
  }
}
```

## 五、Websocket

```
"channel": 888,  
"ticket": "123456"  
}  
}
```

最终Websocket连接地址为：<wss://websocket.bbbug.com/?account=10000&channel=888&ticket=123456>

## 六、测试用例

<https://tester.hamm.cn/#/20200904/10c316cfa2afd902dc7695750ba4709dbb45826c>



## 2、发送文字消息接口

本文档由 **Tester** 自动生成，最后修改时间 2020-09-04 09:48:11

### 一、接口说明

普通文本消息发送接口

### 二、请求方式

POST <https://api.bbbug.com/api/message/send>

### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
where	string	是	channel	固定channel
to	number	是	888	发送的房间
type	string	是	text	固定为text
at	object	是	{}	@对象 不@请给null
at.user_id	number	否	10000	@的ID
at.user_name	string	否	Hamm	@的昵称
msg	string	是	%E6%B5%8B%E8%AF%95	urlencode后的消息

示例JSON :

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "where": "channel",
  "to": 888,
  "type": "text",
  "at": {
    "user_id": 10000,
    "user_name": "Hamm"
  },
  "msg": "%E6%B5%8B%E8%AF%95"
}
```

### 四、更多参数

Headers:

```
Content-Type:application/json;
```

### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-

示例返回结果 :

## 五、Websocket

```
{  
  "code": 200,  
  "msg": "success"  
}
```

## 六、测试用例

<https://tester.hamm.cn/#/20200904/feea9dd7e432becad6ffd3894008b34ec6a6b7bc>

### 3、发送图片消息接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:50:53

#### 一、接口说明

你可以在这里对接口进行一些简单的描述

#### 二、请求方式

POST <https://api.bbbug.com/api/message/send>

#### 三、请求参数

字段	类型	必填	示例	说明
access_token	string	是	123456	-
plat	string	是	pc	-
version	number	是	10000	-
where	string	是	channel	固定channel
to	number	是	888	发送的房间ID
type	string	是	img	类型,固定img
msg	string	是	<a href="http://20200901925545_RiZdJq.jpg">http://20200901925545_RiZdJq.jpg</a>	压缩图地址
resource	string	是	<a href="http://20200901925545_RiZdJq.jpg">http://20200901925545_RiZdJq.jpg</a>	原图地址

示例JSON：

```
{
  "access_token": "123456",
  "plat": "pc",
  "version": 10000,
  "where": "channel",
  "to": 888,
  "type": "img",
  "msg": "http://20200901925545_RiZdJq.jpg",
  "resource": "http://20200901925545_RiZdJq.jpg"
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是	success	-

示例返回结果：

```
{
  "code": 200,
  "msg": "success"
}
```

五、Websocket

六、测试用例

<https://tester.hamm.cn/#/20200904/a7e51f5ccce7528e9740aec91a3aeb3f6f2564be>

## 4、WebSocket消息解析说明文档

### 数据说明

所有WebSocket下发的数据均为JSON标准格式，但不保证因服务器错误返回错误格式的数据，请注意异常处理。

### 数据类型

事件	说明	包含数据
addSong	点歌	用户对象/歌曲对象
playSong	播放歌曲	用户对象/歌曲对象
push	置顶歌曲	用户对象/歌曲对象
system	系统消息	-
text	文字消息	用户对象/@对象/消息/key/sha
img	图片消息	用户对象/@对象/消息/key/sha
online	在线列表	用户对象
link	链接消息	-
removeSong	移除已点歌曲	-
removeban	解禁用户	-
shutdown	禁言	-
songdown	禁歌	-
pass	切歌	-
passGame	pass猜歌	-
back	撤回	-
roomUpdate	房间资料更新	-
game_music	猜歌消息下发	-
game_music_success	猜歌结果	-

## 六、图片接口

### 1、搜索图片表情包

本文档由 `Tester` 自动生成，最后修改时间 2020-09-04 09:43:06

#### 一、接口说明

通过关键词搜索表情包，该接口第三方服务器略卡，请注意UI交互。

#### 二、请求方式

POST <https://api.bbbug.com/api/attach/search>

#### 三、请求参数

字段	类型	必填	示例	说明
keyword	string	是	哈哈哈	表情包关键词

示例JSON：

```
{
  "keyword": "哈哈哈"
}
```

#### 四、更多参数

Headers:

```
Content-Type:application/json;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	200	-
msg	string	是		-
data	array	是	[]	URL字符串数组

示例返回结果：

```
{
  "code": 200,
  "msg": "",
  "data": [
    "http://20160330316287_vqgNep.jpg",
    "http://20160330316287_vqgNep.jpg",
    "http://20160330316287_vqgNep.jpg",
  ]
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/c31917d5a8799a6af08e9e4725c11e2270446884>

## 六、图片接口

### 2、上传图片接口

本文档由 [Tester](#) 自动生成，最后修改时间 2020-09-04 09:53:26

#### 一、接口说明

使用标准的multipart/form-data上传图片

#### 二、请求方式

POST <https://api.bbbug.com/api/attach/uploadimage>

#### 三、请求参数

字段	类型	必填	说明
access_token	string	yes	-
plat	string	yes	-
version	number	yes	-
file	file	yes	上传图片二进制流

示例请求参数：

```
access_token: 123456
plat: pc
version: 10000
file: (binary)
```

#### 四、更多参数

Headers:

```
Content-Type:multipart/form-data;
```

#### 五、返回数据

字段	类型	固定	示例值	说明
code	number	是	400	-
msg	string	是	plat参数为必须	-

示例返回结果：

```
{
  "code": 400,
  "msg": "plat参数为必须"
}
```

#### 六、测试用例

<https://tester.hamm.cn/#/20200904/c37377ec3fb062334168ae271fbf2f695df5878a>

## 七、第三方合作接口

### 1、第三方接入聊天室使用手册与帮助文档

#### 接入说明

本文旨在让第三方开发者快速将BBBUG聊天室快速接入到自己的网站中，以为自己网站快速提供群聊/音乐/游戏等聊天室功能。

#### 接入方案

我们使用iframe进行BBBUG聊天室地址的嵌入即可完成快速的集成，数据通信与交互采用HTML5的 postMessage 进行松耦合的方式完成。

#### 用户集成方案

第一步，将自己网站上的用户登录到BBBUG上，如用户未注册，则按OpenID自动注册。具体请求方式查看Tester上的用例：

```
https://tester.hamm.cn/#/20200907/8ee5587f6190e1a56cec5d18276cbef0e4b06602
```

字段	类型	必填	示例
appid	string	是	你的appid
appkey	string	是	你的appkey
openid	string	是	用户在你网站的 唯一账号 标识
head	string	是	用户的头像URL
nickname	string	是	用户的昵称
extra	string	是	回源URL中 #extra# 将替换为此字段

第二步，嵌入iframe到你的网站上，嵌入的URL为：

```
https://bbbug.com/?access_token=ACCESS_TOKEN&room_id=ROOM_ID
```

其中，ACCESS\_TOKEN 为第一步获取到的身份凭证，如不提供，则以临时用户进入。ROOM\_ID 为你想要进入的房间ID，请注意，房间必须为公开房间，如不传，则进入默认888房间。

#### 临时用户说明

如不需要接入自己的用户体系，直接集成上面第二步的URL，不传入 access\_token 即可。

#### 进入自己创建的房间

为了提供更为个性化的服务，我们支持你接入自己创建的房间，进行更好的用户管理，如需要，请集成第二步的URL时，传入指定的 room\_id 即可。

#### 合作案例

虚位以待

#### 更多功能

更多个性化服务正在开发设计中。

#### 联系我

你可以通过 BBBUG.COM ID:10000 或者 Email:admin#hamm.cn (#换成@)联系我们进行合作 :)



## 2、聊天室绑定第三方独立域名

🔔 温馨提示：新版本已经砍掉了域名相关的功能，如需要请下载老版本，本文档已作废！

### 一、使用平台提供的二级域名

聊天室已支持自定义二级域名绑定聊天室，请直接在房间管理里填写一个未使用的二级域名即可。

1. 登录bbbug.com，右上角换房创建自己的房间并进入；
2. 点击房间左上角的管理，设置一个二级域名地址，如 a.bbbug.com
3. 你可以通过二级域名访问到你的房间啦。

### 二、使用独立域名（提供两种方式）

#### 1. 使用http方式的域名直接解析

如需使用独立域名，请提前联系我们配置。否则第三方登录可能无法使用。  
联系邮箱：admin#hamm.cn（#换成@）

请注意，直接解析需要你的域名已在阿里云备案，同时：

1. 已经按顶上的二级域名步骤完成了创建自己的房间并配置好二级域名，
2. CNAME解析一个你的域名到你的二级域名上，如 a.bbbug.com

优点：速度快，无需服务器成本，不经过服务器转发 缺点：不支持https访问，无法替换一些关键词

#### 2. 使用反向代理方式接入独立域名

使用独立域名需要满足已下两个条件：

1. 已经按顶上的二级域名步骤完成了创建自己的房间并配置好二级域名，
2. 拥有一台你自己的服务器，如国内服务器，你自己已完成备案，
3. 创建站点并配置反向代理即可。

优点：支持自由配置站点https和反代策略 缺点：需要无需备案的服务器成本，速度稍慢，但可忽略不计。

教程：使用宝塔创建站点：

```
#宝塔创建好你的站点，点击配置反向代理，编辑反向代理配置文件，填入：
location / {
    proxy_pass https://120.79.234.189/; # 无需改动 平台IP
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Host "a.bbbug.com"; # 这里配置你在平台的二级域名
    proxy_set_header Connection "Upgrade";
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

教程：反向代理配置文件：

```
# 这里我们提供Nginx的配置Demo
server{
    listen 80;
    # 当然 这里你可以自由的配置https和站点的一些其他信息
    server_name yourdomain.com; # 这里配置你自己的域名
    location / {
        proxy_pass https://120.79.234.189/; # 无需改动 平台IP
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Host "you.bbbug.com"; # 这里配置你在平台的二级域名
        proxy_set_header Connection "Upgrade";
    }
}
```

## 七、第三方合作接口

```
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header Accept-Encoding "";
subs_filter_types text/css text/xml application/json application/javascript;
subs_filter 'BBBUG聊天室' '我的聊天室'; # 可以做一些关键词替换
}
}
```

### 3、SVG外链和iFrame播放器外链使用说明

#### SVG房间信息外链



```
//markdown方式
![BBBUG](https://api.bbbug.com/api/badge/888 "BBBUG")

//html方式


//将上面的888换成你的房间ID即可
```

体验demo:

[BBBUG on Gitee](#)

#### iframe播放器

```
<iframe src="https://api.bbbug.com/api/badge/player/888" frameborder="0" width="500" height="180"></iframe>
//将上面的888换成你的房间ID即可
```

体验demo:

[Hamm.cn](#)

## 4、合作网站

### 接入手册

<https://doc.bbbug.com/2811812.html>

### 合作站点

Hamm's Home Page

魔鬼清博客

## 5、聊天室插件开发使用文档

### 开发说明

为了更丰富BBBUG音乐聊天室的其他玩法，我们设计了基于iframe嵌入方式的插件开发，你可以使用简单的HTML/CSS/JavaScript进行一个第三方插件的开发

普通插件体验demo: <https://bbbug.com/110> 全屏插件体验demo: <https://bbbug.com/2222>

全屏插件为优质插件，如需开发全屏插件，请联系@Hamm (ID:110)

### 准备工作

1. 已部署https的静态网站，已创建好的房间；
2. 查看下面的开放API文档并着手开发。

### API接口文档

#### 1. 监听BBBUG开放的部分事件：

```
window.addEventListener('message', function(event) {
  switch(event.data.event){
    case 'lrc':
      //获取到歌词数据对象事件
      break;
    case 'lrcString':
      //单条歌词变更事件
      break;
    case 'playSong':
      //音乐播放事件
      break;
    case 'sendText':
      //发送文本消息事件
      break;
    case 'touch':
      //摸别人事件
      break;
    case 'addFavSong':
      //添加歌曲收藏事件
      break;
    case 'jump':
      //收到一张飞机票
      break;
    case 'addSong':
      //点歌事件
      break;
    case 'pushSong':
      //顶歌事件
      break;
    case 'removeSong':
      //歌曲移除事件
      break;
    case 'removeban':
      //用户被解禁事件
      break;
    case 'shutdown':
      //用户被禁言事件
      break;
    case 'songdown':
      //用户被禁止点歌事件
      break;
    case 'init':
      //初始化事件 得到用户信息和房间信息
      break;
    case 'sendImg':
      //发送图片事件
      break;
```

## 七、第三方合作接口

```
case 'volumeChanged':  
    //音量发生改变  
    break;  
case 'themeChanged':  
    //主题发生改变  
    break;  
case 'messageController':  
    //全屏应用接管消息  
    break;  
case 'messageList':  
    //聊天记录消息列表回调  
    break;  
default:  
    }  
});
```

### 2. 调用BBBUG API :

首先，引入SDK :

```
<script src="https://api.bbbug.com/sdk/bbbug.js?version=12345" type="text/javascript" charset="utf-8"></script>
```

上面的version=12345建议每次更新代码换一个，强刷一下SDK的JS缓存，保证SDK最新。

接下来，你就可以使用BBBUG.xxx()调用我们提供的API啦~

发送文本消息 :

```
BBBUG.sendMessage("Api says: Wish you fuck your bugs!");
```

显示左下角歌词 :

```
BBBUG.showLrc();
```

隐藏左下角歌词 :

```
BBBUG.hideLrc();
```

设置播放器音量 :

```
BBBUG.setVolume(50);
```

打开指定用户主页 :

```
BBBUG.showUserInfo(110);
```

显示播放器 :

```
BBBUG.showPlayer();
```

隐藏播放器 :

```
BBBUG.hidePlayer();
```

切歌或者不喜欢 :

```
BBBUG.passTheSong();
```

## 七、第三方合作接口

收藏播放的歌曲：

```
BBBUG.loveTheSong();
```

请求当前歌曲信息：(将回调到 `playSong` 事件上)

```
BBBUG.getNowSong();
```

请求当前歌曲歌词数据：(将回调到 `lrc` 事件上)

```
BBBUG.getLrcObj();
```

请求主题类型：(将回调到 `themeChanged` 事件上)

```
BBBUG.getThemeMode();
```

主动隐藏右边弹出的窗口

```
BBBUG.hideAll();
```

发送图片接口

```
BBBUG.sendMessage(url);
```

//请注意，你只能使用我们支持的图片地址，与BBBUG官网表情、搜索表情、上传图片URL等同域名。

### 3. 调用服务端API

目前，我们已经提供了公开的API，你可以直接尝试在插件中发起Ajax请求：

歌曲搜索API: <https://doc.bbbug.com/2811432.html>

获取歌曲LRC: <https://doc.bbbug.com/2811564.html>

歌曲播放地址: <https://doc.bbbug.com/2811630.html>

房间在线用户: <https://doc.bbbug.com/2811408.html>

待播放歌曲列表: <https://doc.bbbug.com/2811670.html>

更多API如有需要，请联系 @Hamm ，我们将合理开放部分API。

需求地址：[https://gitee.com/bbbug\\_com/ChatWEB/issues/new?issue%5Bassignee\\_id%5D=0&issue%5Bmilestone\\_id%5D=0](https://gitee.com/bbbug_com/ChatWEB/issues/new?issue%5Bassignee_id%5D=0&issue%5Bmilestone_id%5D=0)

## 八、下载相关

### 1、Windows客户端下载

客户端下载地址

[Windows V1.0.4.zip](#)

开源说明

Windows客户端采用C#.NET+WPF完成开发，目前已在Gitee开源，地址：

[https://gitee.com/bbbug\\_com/BBbug\\_Winform\\_Applicaiton](https://gitee.com/bbbug_com/BBbug_Winform_Applicaiton)



## 九、部署相关

### 0、写在最前面

#### 最近更新

最新部署文档查看这里：

<https://doc.bbbug.com/4059335.html>

## 九、部署相关

### 部署声明

本项目中使用的音乐来源于第三方平台，版权归他们自己所有，本项目以及其他各个端均仅用于学习和交流，请勿进行商业或违法使用！

### 部署步骤

部署本项目，你可能需要了解的知识点：

1. LNMP或LAMP架构的软件环境安装与配置
2. ThinkPHP框架的部署与使用相关知识
3. NodeJS与Webpack/Vue等项目部署的相关知识
4. 丰富的Web项目部署经验，如跨域、虚拟站点、https、wss、反向代理等知识

如果你准备好了，那就开始躁起来吧！

第一部分：部署API端步骤说明

第二部分：部署Websocket服务端步骤

第三部分：部署最后的前端部分

## 1、部署API端步骤说明

### 架构说明

BBBUG音乐聊天室API端基于 ThinkPHP6 开发，在部署前你可能需要先了解一下 ThinkPHP6 的相关知识。

### 软件环境

1. PHP7 (这里我们强烈建议你使用PHP7.1或7.2,避免使用其他版本带来其他的兼容问题)
2. Nginx或Apache架构(我们以 Nginx 为例做部署说明)
3. MySQL (我们建议使用MySQL5.6版本)
4. Redis
5. Composer (我们的依赖项都使用了Composer进行管理)

### 开始部署

1. Clone我们开源仓库的代码到你服务器: `git clone https://gitee.com/bbbug_com/ChatAPI.git`
2. 创建站点并指向 `public` 目录
3. 按 ThinkPHP 官方说明配置伪静态
4. 将 `Database.sql` 导入到你自己的数据库，并修改`conf`表中的部分配置
5. 复制 `.example.env` 到 `.env` 并修改你自己的数据库
6. 如果你的Redis服务器不在本机，你可能还需要修改一下 `config/cache.php` 文件里的Redis连接信息
7. 再站点根目录下执行 `composer install` 安装必要的依赖文件

不出意外，到这里如果直接访问 `http://your-domain.com/` 会跳转到BBBUG.COM官网聊天室,访问 `http://your-domain.com/api` 会输出Hello World的JSON数据。

因为前后端完全分离，这里你需要配置一下允许你的前端跨域访问API，具体可查看Nginx或Apache的允许跨域配置

到此，API端基本部署完成，接下来我们配置一下歌曲爬虫和歌曲同步部分：

1. 歌曲同步部分 `php think Song`
2. 死链歌曲下载部分 `php think Download`

当然，这里我们强烈推荐使用 Supervisor 对上面两个进程进行守护，鬼知道PHP什么时候又挂掉了。

### 与Websocket服务数据交互部分

当前版本的websocket与API服务是松耦合的设计(作者的遗留问题,之前的websocket服务是公有服务)，所以这里需要配置一下Websocket的连接信息  
修改`conf`表中的IP和端口地址为你部署的websocket服务地址。

Websocket服务的部署手册请点击我查看

### 结语

到这里基本完成了API端的所有配置，接下来去部署其他端吧~

## 2、部署Websocket服务端步骤

### 架构说明

老样子，因为作者的历史遗留问题，Websocket服务需要单独部署。Websocket服务基于NodeJS，你这里需要安装NodeJS(顺带会安装npm包管理器)。

### 部署步骤

1. 首先编辑 websocket.js 文件，将其中的API请求部分地址 `https://api.bbbug.com` 修改为你部署API端的域名地址
2. 安装NodeJS,并执行 `node websocket.js`，如果报错，请安装报错中提示的模块，比如 `npm install nodejs-websocket`
3. 如果你存在端口占用的问题，那么请记得修改一下端口号，同时变更一下API端部署时与Websocket的连接信息部分
4. 我们推荐你使用Nginx对这个Websocket服务进行反代，顺便实现一下基于TLS安全层的WSS服务。
5. 到此原则上Websocket已经部署成功，但我们建议你使用PM2对这个进程进行守护，鬼知道这个服务什么时候也会跟PHP一样炸掉呢。

### 结语

到这里Websocket服务就跑起来了，如果你有疑问或者问题，可以去BBBUG.COM官方聊天室找ID:10000协助解决。

### 3、部署最后的前端部分

#### 架构说明

BBBUG音乐聊天室前端基于Vue&Element开发，你可能需要了解Webpack相关的知识。

#### 部署说明

1. Clone我们提供的开源仓库，`git clone https://gitee.com/bbbug_com/ChatWEB.git`
2. 修改 `src/main.js` 下的 `global` 里的API请求地址和上传文件地址，修改 `src/App.vue` 中的Websocket连接地址 `wss://websocket.bbbug.com` 为你自己部署的Websocket服务地址
3. 安装Webpack相关的工具与NodeJS环境，并执行`npm run build`生成webpack打包后的dist目录
4. 将dist目录部署为你的实际访问站点地址，直接访问即可测试Websocket连接与API请求是否正常

#### 结语

到此，你基本完成了一个BBBUG聊天室的自由部署版本，Enjoy it!

#### 再次声明

歌曲信息来源于第三方，版权归他们自己所有，本项目仅用于学习交流使用，请勿用于商业或非法用途！

## 6、从零开始部署一个你自己的BBBUG音乐聊天室

### 本文说明

因很多小伙伴在部署时碰到一些问题，这里写一个简单的部署手册，以便小伙伴们碰到问题后可以快速查阅解决。

### 部署步骤

#### 一、域名解析

首先，将你的域名解析到服务器对应的IP地址，我这里为了方便，解析了个三级通配域名：

记录	记录类型	目标IP
bbbug_api	A	127.0.0.1
bbbug_www	A	127.0.0.1
bbbug_ws	A	127.0.0.1

接下来，我将 `bbbug_api.hamm.cn` 作为API端地址，`bbbug_ws.hamm.cn` 作为websocket地址，`bbbug_www.hamm.cn` 作为前端地址。

#### 二、部署API端

准备工作，你需要给服务器安装redis、supervisor、MySQL等组件

1. Clone我们代码仓库的API仓库：`git clone https://gitee.com/bbbug_com/ChatAPI.git` BBBUG\_API
2. 创建虚拟站点，目录指向 BBBUG\_API 文件夹下的 public 目录：
3. 配置Thinkphp在Nginx下的rewrite，并配置跨域头；
4. 如果需要配置https，请百度查阅资料进行https的配置。

完整配置如下：

```
server
{
    listen 80;
    server_name bbbug_api.hamm.cn;
    index index.php index.html;

    #允许跨域header
    add_header Access-Control-Allow-Origin * always;
    add_header Access-Control-Allow-Headers * always;
    add_header Access-Control-Allow-Methods * always;
    add_header Access-Control-Max-Age 86400 always;

    #路径为你Clone项目的public目录
    root /home/bbbug/BBBUG_API/public;

    #强制http跳转到https访问
    if ( $scheme = "http" ){
        rewrite (.* ) https://$http_host$1;
    }

    #配置https
    listen 443 ssl http2; #我这里开启了http2 如不需要 删掉 http2
    ssl_certificate /usr/local/nginx/conf/ssl/hamm.cn/fullchain.cer; #证书地址
    ssl_certificate_key /usr/local/nginx/conf/ssl/hamm.cn/hamm.cn.key; #证书密钥
    ssl_session_timeout 10m;
    ssl_session_cache shared:SSL:10m;
    ssl_prefer_server_ciphers on;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
    #我这里启动了TLSv1.3 如果你服务器不支持，删掉即可
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;preload" always;
    #HSTS配置 如不需要 删掉本行

    #配置thinkphp伪静态rewrite规则
    location / {
```

## 九、部署相关

```
if (!-e $request_filename) {
    rewrite ^(.*)$ /index.php?s=/$1 last;
    break;
}
}

location ~ [^/]\.php(/|$)
{
    try_files $uri =404;
    fastcgi_pass unix:/tmp/php-cgi.sock;
    fastcgi_index index.php;
    include fastcgi.conf;
}
location ~ .*\.?(gif|jpg|jpeg|png|bmp|swf|mp3)
{
    expires 30d;
}
access_log off;
}
```

5. 在 BBBUG\_API 目录下执行 `composer install`，期间可能发生N个错误，请根据错误提示解决composer的相关问题：)

直到你看到这个提示：**Succeed!**

如果不出意外，你将会看到我满怀情深的问候：**Hello BBBUG.COM!**

接下来，将根目录下的 `.example.env` 复制或者修改一份为 `.env` 并修改其中的数据库配置。

这时，我们建议设置 `APP_DEBUG=true`，以便你解决接下来遇到的各种问题：)

同时，创建数据库并将根目录下的 `Database.sql` 导入到数据库中，如果不出意外，访问 [https://bbbug\\_api.hamm.cn/api](https://bbbug_api.hamm.cn/api) 将提示 **请先配置sa\_conf表的api\_url字段数据**

于是将 [https://bbbug\\_api.hamm.cn/](https://bbbug_api.hamm.cn/) 配置到 `sa_conf` 表中的 `api_url` 和 `static_url` 字段数据上。

再次访问，你将得到一段JSON

```
{"code":200,"msg":"open","data":{"success":1,"systemVersion":1622013180}}
```

API端基本部署完成。

### 三、部署Websocket端

websocket基于node开发，请先安装好node环境，并安装好相关的依赖库，如 `pm2`，`node-websocket` 等。

修改BBBUG\_API目录下的 `websocket_bbbug_chat.js`，将 `apiUrl` 修改为刚才部署好的 [https://bbbug\\_api.hamm.cn](https://bbbug_api.hamm.cn)

然后执行 `node websocket_bbbug_chat.js` 将会启动websocket服务：

```
Ready now.
*****
服务启动成功(10011)Websocket
当前在线连接数：(0)
Web服务{10012}启动成功!
当前在线连接数：(0)
当前在线连接数：(0)
当前在线连接数：(0)
当前在线连接数：(0)
当前在线连接数：(0)
当前在线连接数：(0)
```

我们建议用PM2对这个服务进行守护，避免这东西跟PHP一样不争气

如果你将要配置https站点，你可能需要按下面的nginx反向代理进行wss配置：

```
server{
    listen 80;
    server_name bbbug_ws.hamm.cn;
```

## 九、部署相关

```
#配置https
listen 443 ssl http2; #我这里开启了http2 如不需要 删掉 http2
ssl_certificate /usr/local/nginx/conf/ssl/hamm.cn/fullchain.cer; #证书地址
ssl_certificate_key /usr/local/nginx/conf/ssl/hamm.cn/hamm.cn.key; #证书密钥
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_prefer_server_ciphers on;
ssl_ciphers HIGH:!aNULL:!MD5;
ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3; # 我这里启动了TLSv1.3 如果你服务器不支持，删掉即可
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;preload" always; #HSTS配置 如不需要 删掉本行

location / {
    proxy_pass http://127.0.0.1:10011;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Host $host;
    proxy_set_header Connection "Upgrade";
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 3600s;
    proxy_send_timeout 3600s;
}
}
```

不出意外，websocket服务端部署完成！

## 四、部署web前端

1. Clone我们提供的前端仓库：`git clone https://gitee.com/bbbug_com/ChatWEB.git BBBUG_WEB`
2. 修改BBBUG\_WEB/src/main.js中的：

```
apiUrl: "https://bbbug_api.hamm.cn/", staticUrl: "https://bbbug_api.hamm.cn/", wssUrl: "wss://bbbug_ws.hamm.cn/",
```

如果你无需https，上面的wssUrl改为 `ws://服务器IP:10011`

3. 安装相关依赖项，在BBBUG\_WEB根目录执行 `npm install`
4. 编译并生成发布文件 `npm run build`，接下来会在BBBUG\_WEB下生成dist目录
5. 创建虚拟站点并绑定刚才创建的dist目录：

```
server
{
    listen 80;
    server_name bbbug_www.hamm.cn;
    index index.html;

    #绑定编译的dist目录
    root /home/bbbug/BBBUG_WEB/dist;

    #强制http跳转到https访问
    if ( $scheme = "http" ){
        rewrite (.* ) https://$http_host$1;
    }

    #解决SPA下404的问题
    location / {
        try_files $uri $uri /index.html;
    }

    #配置https
    listen 443 ssl http2; #我这里开启了http2 如不需要 删掉 http2
    ssl_certificate /usr/local/nginx/conf/ssl/hamm.cn/fullchain.cer; #证书地址
    ssl_certificate_key /usr/local/nginx/conf/ssl/hamm.cn/hamm.cn.key; #证书密钥
    ssl_session_timeout 10m;
    ssl_session_cache shared:SSL:10m;
    ssl_prefer_server_ciphers on;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3; # 我这里启动了TLSv1.3 如果你服务器不支持，删掉即可
```



## 九、部署相关

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;preload" always; #HSTS配置 如不需要 删掉本行  
access_log off;  
}
```

同时将部署好的前端地址 [https://bbbug\\_www.hamm.cn/](https://bbbug_www.hamm.cn/) 更新到 sa\_conf 表的 front\_url 字段数据上

前端就基本部署完成啦~

## 五、歌曲同步

接下来回到BBBUG\_API目录中，启动两个命令：

- php think Song
- php think Download

我们推荐用Supervisord守护一下这两个进程，避免频繁挂掉。

到此，你的聊天室应该已经有音乐啦！

## 六、邮箱配置和第三方登录等配置

邮箱登录部分，请到邮件服务商找到邮件服务器、端口、账号密码等，然后修改到sa\_conf表中即可。

第三方登录，请申请第三方登录的相关appid、secret，修改到sa\_app表中，并修改 app/api/controller/user.php 中的 thirdlogin 方法对应的参数中即可。

另外，前端项目BBBUG\_WEB/src/main.js中同时需要修改这部分参数，并重新执行 npm run build 编译即可。

## 七、其他配置

小程序相关参数配置请直接填入sa\_conf表中即可。

## 八、部署案例

LEMNDO音乐聊天室 <https://www.lemndo.com/>

<https://bbbug.bubu.show/>

<https://music.postlog.cn/>

<https://music.plcms.cn/>

<http://bbug.troycm.top/>

<http://music.fykcy.top/>

<https://cpdd.wang/>

<http://game.fxnews.com/>

<http://chat.bohongdata.cn/>

<http://piaoyaoi.top/>

<http://123.56.234.111/>

## Test

```
| App.vue  
| index.html  
| main.js  
| vendor.js  
  
├─api  
|   | admin.js  
|   | login.js  
|   └─system  
|       menu.js  
|       role.js  
|       user.js  
  
├─assets  
|   └─css  
|       | admin.css  
|       | animation.css  
|       | element.css  
|       | element.rewrite.css  
|       | theme.mes.css  
|       | vue.preview.css  
|       └─fonts  
|           element-icons.ttf  
|           element-icons.woff  
|           Gotham-Book.woff2  
|   └─img  
|       bg.jpg  
  
├─components  
|   BaseSearcher.vue  
|   ChildMenu.vue  
|   EditForm.vue  
|   FieldSelector.vue  
|   Pagination.vue  
|   SingleDelete.vue  
|   ToolBar.vue  
|   ViewTypeSelector.vue  
  
├─field  
|   menu.js  
|   role.js  
|   user.js  
  
├─helper  
|   field.js  
|   request.js  
|   router.js  
|   secret.js  
|   tool.js  
  
├─pages  
|   403.vue  
|   404.vue  
|   admin.vue  
|   home.vue  
|   login.vue  
|   └─system  
|       └─menu  
|           index.vue  
|       └─role  
|           authorize.vue  
|           index.vue
```

## 九、部署相关

```
|   └─user  
|       index.vue  
└─static  
    favicon.ico
```

