共享单车API接口技术文档

**文件编号： XXX-XXX-XXXX**

**产品名称： 共享单车云平台**

**项目名称：**

|  |  |  |  |
| --- | --- | --- | --- |
| 编 写 |  | 日 期 |  |
| 审 核 |  | 日 期 |  |
| 批 准 |  | 日 期 |  |

**（内部资料，注意保密）**

**XXXOOO有限公司**

修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **日期** | **修改纪要** | 修改人 | 审核人 |
| V0.1 | 2018-11-16 | 创建初稿 | 李智勇 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[共享单车API接口技术文档 1](#_Toc531640652)

[修改记录 2](#_Toc531640653)

[1. 文档介绍 5](#_Toc531640654)

[2. API接口协议定义 5](#_Toc531640655)

[2.1 协议头定义 6](#_Toc531640656)

[2.2 协议的type定义 7](#_Toc531640657)

[2.3 消息ID的定义 7](#_Toc531640658)

[2.4 消息序列号 9](#_Toc531640659)

[3. APP与云平台API接口定义 10](#_Toc531640660)

[3.1 用户获取登录验证码请求消息 10](#_Toc531640661)

[3.2 用户获取登录验证码回应消息 10](#_Toc531640662)

[3.3 用户登录请求消息 10](#_Toc531640663)

[3.4 用户登录回复消息 10](#_Toc531640664)

[3.5 用户充值请求消息 11](#_Toc531640665)

[3.6 用户充值回复消息 11](#_Toc531640666)

[3.7 查询账户余额请求消息 11](#_Toc531640667)

[3.8 查询账户余额回复消息 12](#_Toc531640668)

[3.9 查询所有消费记录请求消息 12](#_Toc531640669)

[3.10 查询所有消费记录回复消息 12](#_Toc531640670)

[3.11 解锁单车的请求消息 13](#_Toc531640671)

[3.12 解锁单车的回复消息 13](#_Toc531640672)

[3.13 单车关锁的请求消息 13](#_Toc531640673)

[3.14 单车关锁的回复消息 14](#_Toc531640674)

[3.15 上报单车故障请求消息 14](#_Toc531640675)

[3.16 上报单车故障恢复消息 15](#_Toc531640676)

[3.17 查询骑行记录请求消息 15](#_Toc531640677)

[3.18 查询骑行记录回复消息 15](#_Toc531640678)

[4. 单车与云平台API接口定义 16](#_Toc531640679)

[4.1 解锁单车请求消息 16](#_Toc531640680)

[4.2 解锁单车回复消息 16](#_Toc531640681)

[4.3 关锁单车请求消息 17](#_Toc531640682)

[4.4 关锁单车回复消息 17](#_Toc531640683)

[4.5 下载媒体资源文件请求消息 17](#_Toc531640684)

[4.6 下载媒体资源文件回复消息 18](#_Toc531640685)

[4.7 上传单车位置请求消息 18](#_Toc531640686)

[4.8 上传单车位置回复消息 18](#_Toc531640687)

[5. 公共部分API接口定义 19](#_Toc531640688)

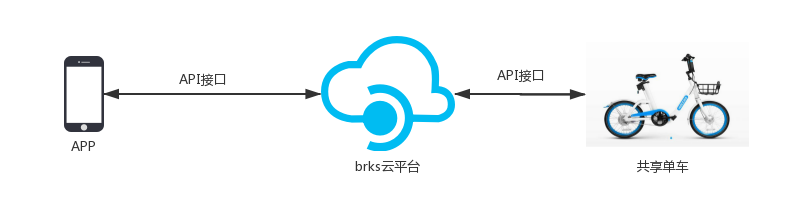
[5.1 心跳PING消息 19](#_Toc531640689)

[5.2 心跳PONG消息 19](#_Toc531640690)

[6. 错误码定义 19](#_Toc531640691)

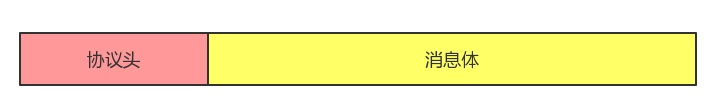
# 文档介绍

共享单车包括brks云平台、手机APP和共享自行车组成，该文档旨在说明手机APP/小程序和云平台的接口定义，共享自行车和云平台的接口定义。如下图：



# API接口协议定义

该接口为应用层协议，且采用TCP作为底层的传输协议，协议总共包含两部分，如下图所示：



### 2.1 协议头定义

下文中的u代表unsigned, u后面的数字代表是多少位，比如u16就表示unsigned short，u8代表unsigned char，u32代表unsigned int。

struct protocol\_head\_t

{

u16 tag\_; // 0xFBFC

u8 version\_; // protocol version, high 4 bit is master version,

// low 4 bit is sub version

u8 type\_; // content type，json协议格式

u16 len\_; // content length，消息的边界，TCP粘包

u16 msg\_id\_; // message id

u64 msg\_sn; // message serise number

u32 reserve\_; // reserve bytes

};

|  |  |  |
| --- | --- | --- |
| 字段 | 长度（byte） | 说明 |
| tag | 2 | 先导码，标志消息的起始和结束 |
| version | 1 | 协议版本号，高4位为主版本，低4位为子版本 |
| type | 1 | 消息体的类型，另述如下文 |
| len | 2 | 消息体的长度，不包含协议头 |
| msg\_id | 2 | 消息ID， GET HEAD POST |
| msg\_sn | 8 | 低14位表示用户手机号码的后4位数字，或者设备编号的后4位数字，再接着4位为编号服务器的机器ID，再接着6位为通信方向，剩余的40位表示消息序列号。 |
| reserve | 4 | 预留 |

### 2.2 协议的type定义

定义这个type是为了这个协议可以兼容不同的编码方式。

enum protocol\_type\_t

{

JSON\_PROTOCOL\_TYPE = 0, // json

PB\_PROTOCOL\_TYPE = 1, // Google Protocol Buffer

FB\_PROTOCOL\_TYPE = 2, // FlatBuffers

BINARY\_PROTOCOL\_TYPE = 3, // binary format

UNKNOW\_PROTOCOL\_TYPE = 0xFF

};

### 2.3 消息ID的定义

消息ID定义的是消息的类型，消息类型也有两大类，第一类是请求类型（以REQ结尾），另一类是回应类型（以RSP结尾），每请求必须要有回应，如果未收到回应，意为消息丢失或者接收方处理失败，此时需要按照以下规则处理：

规则一：发送方重新发送请求消息，重复的间隔单位时间是500毫秒，重复的系数是1,4,16,64，也就是说重复4次发送，第一次间隔是500毫秒，第二次是2秒，第三次是8秒，第四次是32秒。重发直到成功或者超过4次放弃发送消息。

规则二：如果消息是由云平台主动发送的，或者是云平台内部的，重发4次后依然未收到回复，则把该消息内容以告警（alarm）的形式发送给运营模块。

其类型定义有：

/\* 消息ID \*/

enum EventID

{

EEVENTID\_COMMON\_RSP = 0x00,

EEVENTID\_GET\_MOBILE\_CODE\_REQ = 0x01,

EEVENTID\_GET\_MOBILE\_CODE\_RSP = 0x02,

EEVENTID\_LOGIN\_REQ = 0x03,

EEVENTID\_LOGIN\_RSP = 0x04,

EEVENTID\_RECHARGE\_REQ = 0x05,

EEVENTID\_RECHARGE\_RSP = 0x06,

EEVENTID\_GET\_ACCOUNT\_BALANCE\_REQ = 0x07,

EEVENTID\_GET\_ACCOUNT\_BALANCE\_RSP = 0x08,

EEVENTID\_LIST\_ACCOUNT\_RECORDS\_REQ = 0x09,

EEVENTID\_LIST\_ACCOUNT\_RECORDS\_RSP = 0x10,

EEVENTID\_UNLOCK\_REQ = 0x11,

EEVENTID\_UNLOCK\_RSP = 0x12,

EEVENTID\_LIST\_TRAVELS\_REQ = 0x13,

EEVENTID\_LIST\_TRAVELS\_RSP = 0x14,

EEVENTID\_LOCK\_REQ = 0x15,

EEVENTID\_LOCK\_RSP = 0x16,

EEVENTID\_UNKOWN = 0xFFFF

};

### 2.4 消息序列号

消息序列号唯一标识一条消息，该消息序列号由4部分组成，如下所示：



账户或者设备编号，14 bits，为账户或者设备编号的后4位。

编号机器编号，4 bits，编号机器就是专门给整个系统编号服务的比如订单编号，消息序号编号等等，系统运行中，支持集群编号，因此需要规定编号机器的一个虚拟ID值，其ID值的范围为0~15，共支持16台编号机器的编号。在消息ID序列号中占据4位。

通信方向编号，6 bits，也即定义了所有的消息方向，其定义如下：

enum message\_arrow\_t

{

APP\_2\_CLOND\_ARROW = 0,

CLOND\_2\_APP\_ARROW = 1,

BIKE\_2\_CLOND\_ARROW = 2,

CLOND\_2\_BIKE\_ARROW = 3,

UNKONW\_ARROW = 63 // message arrow of 6 bit is 111111.

};

通信序列号，40 bits，每个发送模块单独负责编号，比如APP，brks云平台，自行车。但是回应类型的消息序号（40 bits）需与请求类型消息的序号一致。消息序号编号原则还需满足以下规则：

规则1：每个模块编号的消息不能重复，除非40个比特位全都编号完。

规则2：消息编号必须持久化。

# APP与云平台API接口定义

### 3.1 用户获取登录验证码请求消息

消息ID：EEVENTID\_GET\_MOBILE\_CODE\_REQ（0x01）

消息体：如下所示，mobile为必填字段。

{

“mobile”:”15200599665”

}

有效载荷：比较少的

### 3.2 用户获取登录验证码回应消息

消息ID：EEVENTID\_GET\_MOBILE\_CODE\_RSP（0x02）

消息体：消息如下所示，code和msg为必填字段，code为回复的状态码，见状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.3 用户登录请求消息

消息ID：EEVENTID\_LOGIN\_REQ

消息体：消息如下所示，code为用户手机收到的短信验证码，必填字段。

{

“mobile”: ”15200599665”,

“code”:”xxxooo”

}

### 3.4 用户登录回复消息

消息ID：EEVENTID\_LOGIN\_RSP

消息体：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.5 用户充值请求消息

消息ID：EEVENTID\_RECHARGE\_REQ

消息内容：消息如下所示，amount表示充值金额，为整数。

{

“mobile”: ”15200599665”,

“amount”:100,

}

### 3.6 用户充值回复消息

消息ID：EEVENTID\_RECHARGE\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.7 查询账户余额请求消息

消息ID：EEVENTID\_GET\_ACCOUNT\_BALANCE\_REQ

消息内容：消息如下所示。

{

“mobile”: ”15200599665”

}

### 3.8 查询账户余额回复消息

消息ID：EEVENTID\_GET\_ACCOUNT\_BALANCE\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义，balance是账户余额。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”,

“balance“: 95

}

### 3.9 查询所有消费记录请求消息

消息ID：EEVENTID\_LIST\_ACCOUNT\_RECORDS\_REQ

消息内容：消息如下所示。

{

“mobile”: ”15200599665”

}

### 3.10 查询所有消费记录回复消息

消息ID：EEVENTID\_LIST\_ACCOUNT\_RECORDS\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义，type为消费类型，取值如下所示所示：

0 : 骑行消费, 1 : 充值, 2 : 退款。

limit为消费的金额，timestamp为消息发生的时间，是格林威治时间。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”,

“records“: [

{

“type“: 0,

“limit“: 1.5,

“timestamp“: 1543288868

}，

{

“type“: 0,

“limit“: 1.5,

“timestamp“: 1543288868

}

{…}

]

}

### 3.11 解锁单车的请求消息

消息ID：EEVENTID\_UNLOCK\_REQ

消息内容：消息如下所示，bikecode是单车编号。

{

“mobile”: ”15200599665”，

“bikecode“: “1234567890“

}

### 3.12 解锁单车的回复消息

消息ID：EEVENTID\_UNLOCK\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.13 单车关锁的请求消息

消息ID：EEVENTID\_LOCK\_REQ

消息内容：消息如下所示，bikecode是单车编号。

{

“mobile”: ”15200599665”，

“bikecode“: “1234567890“

}

### 3.14 单车关锁的回复消息

消息ID：EEVENTID\_LOCK\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.15 上报单车故障请求消息

消息ID：EEVENTID\_REPORT\_BIKE\_FAULT\_REQ

消息内容：消息内容如下所示，bikecode是单车编码，part是故障部位，定义如下：

0：坐垫， 1：车头， 2：脚踏， 3：车把， 4：支腿， 5：刹车，

6：挡泥板， 7：车锁， 8：链条， 9：二维码，10：加私锁。

desc表示故障描述，pic是单车故障部位描述图片，转换成base64编码后传输。

{

“mobile”: ”15200599665”，

“bikecode“: “1234567890“，

“part“: 0,

“desc“: “坐垫松动“,

“pic“: “jdkjflkjsiueptqu“

}

### 3.16 上报单车故障恢复消息

消息ID：EEVENTID\_REPORT\_BIKE\_FAULT\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 3.17 查询骑行记录请求消息

消息ID：EEVENTID\_LIST\_TRAVELS\_REQ

消息内容：消息如下所示。

{

“mobile”: ”15200599665”

}

### 3.18 查询骑行记录回复消息

消息ID：EEVENTID\_LIST\_TRAVELS\_RSP

消息内容：消息如下所示，records是必填字段表示每一次骑行的记录，amount表示此次骑行消费金额，duration表示骑行时长，startTimeStamp表示骑行开始时间，是格林威治时间，mileage表示该用户的总骑行里程，discharge表示用户总的节能的碳排放量，colorie表示用户总的燃烧的卡路里。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”,

“mileage“: 10.3,

“discharge“: 115.2,

“calories“: 305.7,

“records“: [

{

“amount“: 1,

“duration“: 360,

“startTimeStamp“: 1543288868

}，

{

“amount“: 1,

“duration“: 720,

“timestamp“: 1543288868

}

{…}

]

}

# 单车与云平台API接口定义

### 4.1 解锁单车请求消息

消息ID：EEVENTID\_UNLOCK\_BIKE\_REQ

消息内容：消息内容如下所示，mobile为可选字段。

{

“mobile”: ”15200599665”

}

### 4.2 解锁单车回复消息

消息ID：EEVENTID\_UNLOCK\_BIKE\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 4.3 关锁单车请求消息

消息ID：EEVENTID\_LOCK\_BIKE\_REQ

消息内容：消息内容如下。

{

“mobile”: ”15200599665”

}

### 4.4 关锁单车回复消息

消息ID：EEVENTID\_LOCK\_BIKE\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 4.5 下载媒体资源文件请求消息

消息ID：EEVENTID\_DOWNLOAD\_MEDIA\_FILE\_RSQ

消息内容：消息内容如下所示，type表示资源的类型，0表示在开锁后播放的提示音，1表示关锁后播放的提示音，2表示开锁后的促销广告，在解锁单车提示音播放后播放（资源0播放后），3表示表示关锁后的促销广告，在关锁单车提示音播放后播放（资源1播放后）。

md5表示资源下载的校验码，校验资源下载的完整性。

{

“url”: ” <http://dongnaobike.com/fe0f43c1d1064180d9cf949b.mp3>”,

“type”:0,

“md5”:” fe0f43c1d10641d099b2cd80d9cf949b”

}

### 4.6 下载媒体资源文件回复消息

消息ID：EEVENTID\_DOWNLOAD\_MEDIA\_FILE\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

### 4.7 上传单车位置请求消息

消息ID：EEVENTID\_REPORT\_BIKE\_POSITION\_REQ

消息内容：如下所示，如果位置没有变化则不需要上报位置信息，否则每隔15秒上报一次。type表示单车类型，0为xxx，1为ooo。latitude为维度，longitude为经度，两者的精确到小数点后6位。

在骑行过程中，上传的地理位置信息可以替代PING/PONG消息，也即不用额外再单独发送心跳消息。

{

“bikecode”: ”xxjkjljkfish”,

“type“:0,

“latitude“:39.920005,

“longitude“:116.463892

}

### 4.8 上传单车位置回复消息

消息ID：EEVENTID\_REPORT\_BIKE\_POSITION\_RSP

消息内容：消息如下所示，code为回复的状态码，请参照状态码定义。

{

“code”:200,

“msg”:”success”,

“data”:”xxxxx”

}

# 公共部分API接口定义

### 5.1 心跳PING消息

消息ID：EEVENTID\_HEART\_BREAK\_PING\_MSG

消息内容：null，仅仅只发送一个消息头就够了，云平台为中心，所有的终端和APP都向云平台发送PING消息，云平台回复PONG消息，单车如果超过3次收不到云平台的PONG消息，则处理离线，云平台如果超过3次收到单车的PING消息，则云平台认为单车处于离线。

每隔30秒发送一个PING/PONG消息。

### 5.2 心跳PONG消息

消息ID：EEVENTID\_HEART\_BREAK\_PONG\_MSG

消息内容：null，仅仅发送一个消息头就可以了。

# 错误码定义

状态码定义如下表：

|  |  |
| --- | --- |
| code | 描述 |
| 200 | 成功 |
| 400 | 消息格式错误 |
| 404 | 消息数据错误 |
| 405 | 不支持的处理 |
| 406 | 处理错误了，msg错误消息信息，data可以填充更加详细的内容。 |
| 407 | 单车已被他人占用 |
| 408 | 单车已经开锁骑行了 |
| 409 | 单车已经损坏 |