PetCare系统服务器端接口文档

作 者： 郑 磊

编写日期： 2015-04-26

**更改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 修改类型 | 修改描述 | 修改人 | 版本 |
| 2015-04-26 | A | 新增 | 郑磊 | 1.0.0 |
| 2015-05-04 | U | 修改接口实现方式  添加上传和推送录音接口 | 郑磊 | 1.0.1 |
| 2015-05-10 | U | 修改接口  添加接口十九到二十三 | 郑磊 | 1.0.1 |

目录

[PetCare系统服务器端接口文档 1](#_Toc419053210)

[**1.** **系统接口说明** 4](#_Toc419053211)

[**2.** **整体规范** 4](#_Toc419053225)

[**3.** **接口一：账号注册接口** 6](#_Toc419053226)

[**4.** **接口二：登录验证接口** 7](#_Toc419053227)

[**5.** **接口三：修改注册信息接口** 7](#_Toc419053228)

[**6．接口四：添加好友接口** 8](#_Toc419053229)

[**7. 接口五：实时接收信息接口** 8](#_Toc419053230)

[**8. 接口六：传输实时信息接口** 9](#_Toc419053231)

[**9. 接口七：轨迹回放接口** 10](#_Toc419053232)

[**10. 接口八：删除轨迹信息接口** 10](#_Toc419053233)

[**11. 接口九：用户发布信息接口** 11](#_Toc419053234)

[**12 接口十：取得所有用户发布信息接口** 12](#_Toc419053235)

[**13 接口十一：取得当前用户发布信息接口** 12](#_Toc419053236)

[**14 接口十二：上传图片接口** 13](#_Toc419053237)

[**15 接口十三：上传视频接口** 14](#_Toc419053238)

[**16 接口十四：删除发布信息接口** 14](#_Toc419053239)

[**17 接口十五：用户定制推送消息接口** 15](#_Toc419053240)

[**18 接口十六：取得定制的推送消息接口** 15](#_Toc419053241)

[**19 接口十七：上传录音接口** 16](#_Toc419053242)

[**20 接口十八：推送录音接口** 17](#_Toc419053243)

[**21 接口十九：增加绑定宠物硬件接口** 17](#_Toc419053244)

[**22 接口二十：删除绑定宠物硬件接口** 18](#_Toc419053245)

[**23 接口二十一：变更用户账号与宠物硬件绑定接口** 18](#_Toc419053246)

[**24 接口二十二：增加宠物安全区域接口** 19](#_Toc419053247)

[**25 接口二十三：删除宠物安全区域接口** 20](#_Toc419053248)

1. **系统接口说明**
   1. RESTful HTTP接口简介

本系统服务器端暂定使用RESTful HTTP接口。REST是一种风格，而不是标准。因为既没有REST RFC，也没有REST协议规范或者类似的规定。REST架构是Roy Fielding（他也是HTTP和URI规范的主要作者之一）在一篇论文中描述的。像REST这样的架构风格通常会定义一组高层决定让应用程序去实现。所有实现了某种特定架构风格的应用程序，都使用相同的模式，也用相同的方式使用别的架构元素，如缓存，分布式策略等。Roy Fielding把REST定义成一种架构风格，其目标是“使延迟和网络交互最小化，同时使组件实现的独立性和扩展性最大化”。

REST并没有创造新的技术，组件或服务，隐藏在RESTful HTTP背后的理念是使用Web的现有特征和能力。RESTful HTTP定义了如何更好地使用现有Web标准中的一些准则和约束。

* 1. 使用JSON字符串作为接口间的数据传输

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。

(1) JSON 语法规则:

1. 数据在名称/值对中 ② 数据由逗号分隔

③ 花括号保存对象 ④ 方括号保存数组

(2) JSON 名称/值对

JSON 数据的书写格式是：名称/值对。

例如："firstName":"John"

(3) JSON 值

JSON值可以是：①数字(整数或浮点数)；②字符串(在双引号中)；

③逻辑值(true 或 false)；④数组(在方括号中)；

⑤对象(在花括号中)；⑥null

1. **整体规范**

* 传输前对所有参数进行URLEncode.encode(xxx,”utf-8”)编码
* 签名加解密所用key由技术人员单独发送

【key：^%petcare20150426 )(-+~$】

* 参数名称必须严格遵守，区分大小写，顺序可以改变，不能重复
* 签名顺序必须严格遵守，编码格式为UTF-8
* URL地址：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)(具体URL待定)
  1. 所有接口公有参数及返回结果说明

全局请求参数：

全局参数被放在http请求头里面，一部分全局参数会涉及到签名运算，形式如：httpclient.setHeader("appKey","appkey值");

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| appKey | String | 是 | 应用唯一标识 |
| appDate | String | 是 | 请求时间yyyyMMddHHmmss |
| validate | String | 是 | 签名验证，由(appKey+appDate+key+data(请求参数))经MD5加密算得 |

返回说明

接口验证成功时,HttpResponse状态码是200。验证、处理失败时HttpResponse状态码返回400（包括参数校验失败、逻辑处理失败、服务器异常等信息。例如登陆验证接口，只有在登陆成功的情况下code返回200，失败时返回400，不管是什么原因引起）。HttpResponse状态码是200情况下默认接口验证、处理是成功的，返回的json也是按照每个具体接口拟定的协议形式返回。当状态码是400时，返回的json协议形式是统一的，详见如下示例：

{

"status": 20002,

"result": "false"

}

|  |  |  |  |
| --- | --- | --- | --- |
| **序列** | **返回字段** | **Type** | **说明** |
| 1 | status | varchar | 状态码 |
| 2 | result | varchar | 请求响应结果 |

全局状态码

status列表

|  |  |
| --- | --- |
| status | **描述** |
| 0 | 请求正常响应 |
| 10001 | 服务器异常 |
| 10002 | http连接失败 |
| 20001 | 请求参数不正确 |
| 20002 | 签名验证失败 |

签名算法验证：

由客户端计算签名算法、服务器进行验证。详细介绍如下：

具体的签名算法如下示例：

public static String createSign(String param) throws Exception {

StringBuffer result = new StringBuffer();

MessageDigest md5 = MessageDigest.getInstance("MD5");

md5.update(param.getBytes("UTF-8"));

byte[] b = md5.digest();

for(int i=0;i<b.length;++i) {

int x = b[i]&0xFF;

int h = x>>>4;

int l = x&0x0F;

result.append((char)(h+((h<10)?'0':'a'-10)));

result.append((char)(l+((l<10)?'0':'a'-10)));

}

return result.toString().substring(8, 24);

}

签名值：

String validate = createSign(appKey+appDate+key+data);

其中data的值是每次请求所携带的参数。具体值如下：

1、当get方式请求时，data的值等于'?'后面的参数。例如下请求时[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/register.json?username=abc&password=123&nickName=def

data="username=abc&password=123&nickName=def"

2、当post方式请求时，data的值等于请求里面的json值。例如登陆验证时，data="{'username':'abc','password':'123','nickName':'def' }"

1. **接口一：账号注册接口**
   1. 功能描述：调用该接口进行账号注册
   2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/register.json
   3. HTTP支持方式：POST或GET
   4. 请求参数

提交单据参数data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 注册账号 |
| password | String | 是 | 注册密码 |
| petAccount | String | 是 | 宠物硬件账号 |
| nickName | String | 是 | 昵称 |

* 1. 返回结果
     1. 提交单据返回结果data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30001 账号已被注册  30002 昵称已被注册 |
| result | String | 是 | true/false,代表注册成功or失败 |

data Json格式举例：

{“status”:0,”result”:”true” }

1. **接口二：登录验证接口**
   1. 功能描述：验证用户的登录信息是否正确
   2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/login.json
   3. HTTP支持方式：POST或GET
   4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 注册账号 |
| password | String | 是 | 注册密码 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30003 用户名或密码错误 |
| result | String | 是 | true/false,代表登录成功or失败 |

data Json格式举例：

{“status”:0,”result”:”true” }

1. **接口三：修改注册信息接口**
   1. 功能描述：修改用户注册信息（譬如，昵称、性别、手机号码、所在地等信息）
   2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/modifyRegisterInfo.json
   3. HTTP支持方式：POST或GET
   4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| nickName | String | 否 | 昵称 |
| password | String | 否 | 注册密码 |
| petAccount | String | 否 | 宠物硬件账号 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30004 昵称已被注册 |
| result | String | 是 | true/false,代表修改成功or失败 |

data Json格式举例：

{“status”:30004,”result”:”false” }

**6．接口四：添加好友接口**

* 1. 功能描述：用户在社区中添加好友
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/appendFriend.json
  3. HTTP支持方式：POST或GET
  4. Action：appendFriend
  5. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| myID | String | 是 | 用户ID |
| friendsID | String | 是 | 好友ID |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30005 已经添加好友 |
| result | String | 是 | true/false,代表添加成功or失败 |

data Json格式举例：

{“status”:0,”result”:”true” }

**7. 接口五：实时接收信息接口**

* 1. 功能描述：接收并存储硬件设备返回的宠物位置信息
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/receivePetInfo.json
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| petPosition | String | 是 | 宠物位置信息 |
| petStatus | String | 是 | 宠物状态信息 |
| receiveTime | String | 是 | 接收信息时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30006 宠物信息保存失败 |
| result | String | 是 | true/false,代表接收成功or失败 |

data json格式举例：

{“status”:0,” result”:”true” }

**8. 接口六：传输实时信息接口**

* 1. 功能描述：返回实时位置信息和状态数据
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/transferPetInfo.json
  3. HTTP支持方式：POST或GET
  4. Action：transferPetInfo
  5. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30007 没有宠物信息 |
| result | String | 否 | 该json包含三个参数：<宠物账号（硬件IMEI）、宠物位置信息、宠物状态信息>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“petAccount”: “\*\*”, “petPosition”: “\*\*”， “petStatus”: “\*\*” }, {“petAccount”: “\*\*”, “petPosition”: “\*\*”， “petStatus”: “\*\*” }]” }

**9. 接口七：轨迹回放接口**

* 1. 功能描述：返回轨迹回放数据
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/trackPlayBack.json
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| startTime | String | 是 | 回放开始时间 |
| endTime | String | 是 | 回放结束时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30007 没有宠物信息 |
| result | String | 否 | 该json包含三个参数：<宠物账号（硬件IMEI）、宠物位置信息、宠物状态信息>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“petAccount”: “\*\*”, “petPosition”: “\*\*”， “petStatus”: “\*\*”,”time”:”yyyyMMddHHmmss” }, {“petAccount”: “\*\*”, “petPosition”: “\*\*”， “petStatus”: “\*\*” ,”time”:”yyyyMMddHHmmss”}]” }

**10. 接口八：删除轨迹信息接口**

* 1. 功能描述：删除轨迹信息
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/deleteTrack.json
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| startTime | String | 是 | 删除开始时间 |
| endTime | String | 是 | 删除结束时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30007 没有宠物信息 |
| result | String | 是 | true/false,代表删除成功or失败 |

data json格式举例：

{“status”:0, ”result”:”true” }

**11. 接口九：用户发布信息接口**

* 1. 功能描述：用户发布信息
  2. 接口URL：[http://127.0.0.1:8080/petCare](http://127.0.0.1:8080/ws/wsMainEnter)/user/release.json
  3. HTTP支持方式：POST或GET
  4. Action：release
  5. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| releaseInfo | String | 是 | 发布的信息 |
| releaseTime | String | 是 | 发布的时间 |
| releaseType | String | 是 | 1：发布消息；2：评论；3：点赞 |
| friendAccount | String | 否 | 点赞或评论的朋友账号（当点赞或评论时该字段不能为空，当发布消息时该字段为空） |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30008 发布失败 |
| result | String | 是 | true/false,代表发布成功or失败 |

data json格式举例：

{“status”:0, ”result”:”true” }

**12 接口十：取得所有用户发布信息接口**

* 1. 功能描述：取得所有用户发布信息
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/getAllReleaseInfo.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| pageNo | String | 是 | 查看的页数 |
| pageSize | String | 是 | 每页条数 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30009 没有信息 |
| result | String | 否 | 该json包含八个参数：<好友账号、好友发布信息、点赞账号，点赞数量，评论账号，评论内容，图片URL，视频URL>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“friendAccount”: “\*\*”, “releaseInfo”: “\*\*”， “priseAccount”: “\*\*”,”priseCount”:”\*\*”,“commentAccount”: “\*\*”, “commentInfo”: “\*\*”， “imageUrl”: “\*\*”,”videoURL”:”\*\*” },

{“friendAccount”: “\*\*”, “releaseInfo”: “\*\*”， “priseAccount”: “\*\*”,”priseCount”:”\*\*”,“commentAccount”: “\*\*”, “commentInfo”: “\*\*”， “imageUrl”: “\*\*”,”videoURL”:”\*\*” }]” }

**13 接口十一：取得当前用户发布信息接口**

* 1. 功能描述：取得当前用户发布信息
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/getReleaseInfo.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| pageNo | String | 是 | 查看的页数 |
| pageSize | String | 是 | 每页条数 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30009 没有信息 |
| result | String | 否 | 该json包含八个参数：<用户账号、发布信息、点赞账号，点赞数量，评论账号，评论内容，图片URL，视频URL>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“account”: “\*\*”, “releaseInfo”: “\*\*”， “priseAccount”: “\*\*”,”priseCount”:”\*\*”,“commentAccount”: “\*\*”, “commentInfo”: “\*\*”， “imageUrl”: “\*\*”,”videoURL”:”\*\*” },

{“account”: “\*\*”, “releaseInfo”: “\*\*”， “priseAccount”: “\*\*”,”priseCount”:”\*\*”,“commentAccount”: “\*\*”, “commentInfo”: “\*\*”， “imageUrl”: “\*\*”,”videoURL”:”\*\*” }]” }

**14 接口十二：上传图片接口**

* 1. 功能描述：上传图片
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/uploadImage.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| releaseInfo | String[] | 是 | 发布信息 |
| imageData | byte[][] | 是 | 图片（字节流,二维数组表示可以同时上传多张图片） |
| uploadTime | String | 是 | 上传时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30010 图片上传失败 |
| result | String | 是 | true/false,代表上传成功or失败 |

data json格式举例：

{“status”:0,”result”:”true” }

**15 接口十三：上传视频接口**

* 1. 功能描述：上传视频
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/uploadVideo.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| releaseInfo | String[] | 是 | 发布信息 |
| videoInfo | String[] | 是 | 视频描述信息 |
| videoData | byte[] | 是 | 视频（字节流） |
| uploadTime | String | 是 | 上传时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30011 视频上传失败 |
| result | String | 是 | true/false,代表上传成功or失败 |

data json格式举例：

{“status”:0,”result”:”true” }

**16 接口十四：删除发布信息接口**

* 1. 功能描述：删除发布信息
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/deleteRelease.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| releaseId | String | 是 | 发布信息ID |
| uploadTime | String | 是 | 删除时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30012 删除发布信息失败 |
| result | String | 是 | true/false,代表删除成功or失败 |

data json格式举例：

{“status”:0,”result”:”true” }

**17 接口十五：用户定制推送消息接口**

* 1. 功能描述：用户定制推送消息
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/customPushMsg.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| pushMsgType | String | 是 | 推送消息类型 |
| customTime | String | 是 | 删除时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30013 定制推送消息失败 |
| result | String | 是 | true/false,代表定制成功or失败 |

data json格式举例：

{“status”:0,”result”:”true” }

**18 接口十六：取得定制的推送消息接口**

* 1. 功能描述：取得定制的推送消息
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/getPushMsg.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| pageNo | String | 是 | 查看的页数 |
| pageSize | String | 是 | 每页条数 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30014 没有推送消息 |
| result | String | 否 | 该json包含四个参数：<用户账号、推送消息类型、推送消息，推送时间>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“account”: “\*\*”, “pushMsgType”: “\*\*”， “pushMsg”: “\*\*”,”pushTime”:”\*\*”},{“account”: “\*\*”, “pushMsgType”: “\*\*”， “pushMsg”: “\*\*”,”pushTime”:”\*\*”}]” }

**19 接口十七：上传录音接口**

* 1. 功能描述：上传录音
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/uploadAudio.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| audioData | byte[] | 是 | 录音（字节流） |
| uploadTime | String | 是 | 上传时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30015 录音上传失败 |
| result | String | 是 | true/false,代表上传成功or失败 |

data json格式举例：

{“status”:0,”result”:”true” }

**20 接口十八：推送录音接口**

* 1. 功能描述：推送录音
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/pushAudio.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| audioId | String | 是 | 录音标识 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30015 录音上传失败 |
| result | String | 否 | 该json包含三个参数：<用户账号、宠物账号（硬件IMEI）、录音服务器路径>多组循环 |

data json格式举例：

{“status”:0, ”result”:”[{“account”: “\*\*”, “petAccount”: “\*\*”， “audioUrl”: “\*\*” },{“account”: “\*\*”, “petAccount”: “\*\*”， “audioUrl”: “\*\*”]” }

**21 接口十九：增加绑定宠物硬件接口**

* 1. 功能描述：增加绑定宠物硬件，一个用户（手机号）可对应绑定多个硬件IMEI
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/addBundImei.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| createTime | String | 是 | 增加绑定时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30016 宠物账号不存在  30017 宠物账号已被绑定 |
| result | String | 否 | true/false,代表增加绑定成功or失败 |

data json格式举例：

{“status”:0, ”result”:” true” }

**22 接口二十：删除绑定宠物硬件接口**

* 1. 功能描述：删除绑定宠物硬件
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/delBundImei.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| deleteTime | String | 是 | 删除绑定时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30016 宠物账号不存在  30018 宠物账号未被绑定 |
| result | String | 否 | true/false,代表删除绑定成功or失败 |

data json格式举例：

{“status”:0, ”result”:” true” }

**23 接口二十一：变更用户账号与宠物硬件绑定接口**

* 1. 功能描述：变更用户账号与宠物硬件绑定
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/changeUserAndImei.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| oldUserName | String | 是 | 原用户账号 |
| password | String | 是 | 原用户注册密码 |
| newUserName | String | 是 | 新用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| changeTime | String | 是 | 变更绑定时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30016 宠物账号不存在  30019 原用户账号不存在  30020 新用户账号不存在  30021 原用户账号和密码不匹配  30022 宠物账号已被新用户绑定  30023 宠物账号未被原用户绑定 |
| result | String | 否 | true/false,代表变更绑定成功or失败 |

data json格式举例：

{“status”:0, ”result”:” true” }

**24 接口二十二：增加宠物安全区域接口**

* 1. 功能描述：设定宠物安全区域
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/addSaveArea.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| longCood | String | 是 | 纬度坐标 |
| latiCood | String | 是 | 经度坐标 |
| radius | String | 是 | 安全区域半径 |
| addTime | String | 是 | 增加时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30024 增加安全区域失败 |
| result | String | 否 | true/false,代表增加成功or失败 |

data json格式举例：

{“status”:0, ”result”:” true” }

**25 接口二十三：删除宠物安全区域接口**

* 1. 功能描述：删除宠物安全区域
  2. 接口URL：<http://127.0.0.1:8080/petCare/user/delSaveArea.json>
  3. HTTP支持方式：POST或GET
  4. 请求参数

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| userName | String | 是 | 用户账号 |
| petAccount | String | 是 | 宠物账号（硬件IMEI） |
| longCood | String | 是 | 纬度坐标 |
| latiCood | String | 是 | 经度坐标 |
| radius | String | 是 | 安全区域半径 |
| addTime | String | 是 | 增加时间 |

* 1. 返回结果

data说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必填** | **描述** |
| status | String | 是 | 30025 删除安全区域失败 |
| result | String | 否 | true/false,代表删除成功or失败 |

data json格式举例：

{“status”:0, ”result”:” true” }