iamhere后台服务设计说明文档

# 1. 后台服务设计：

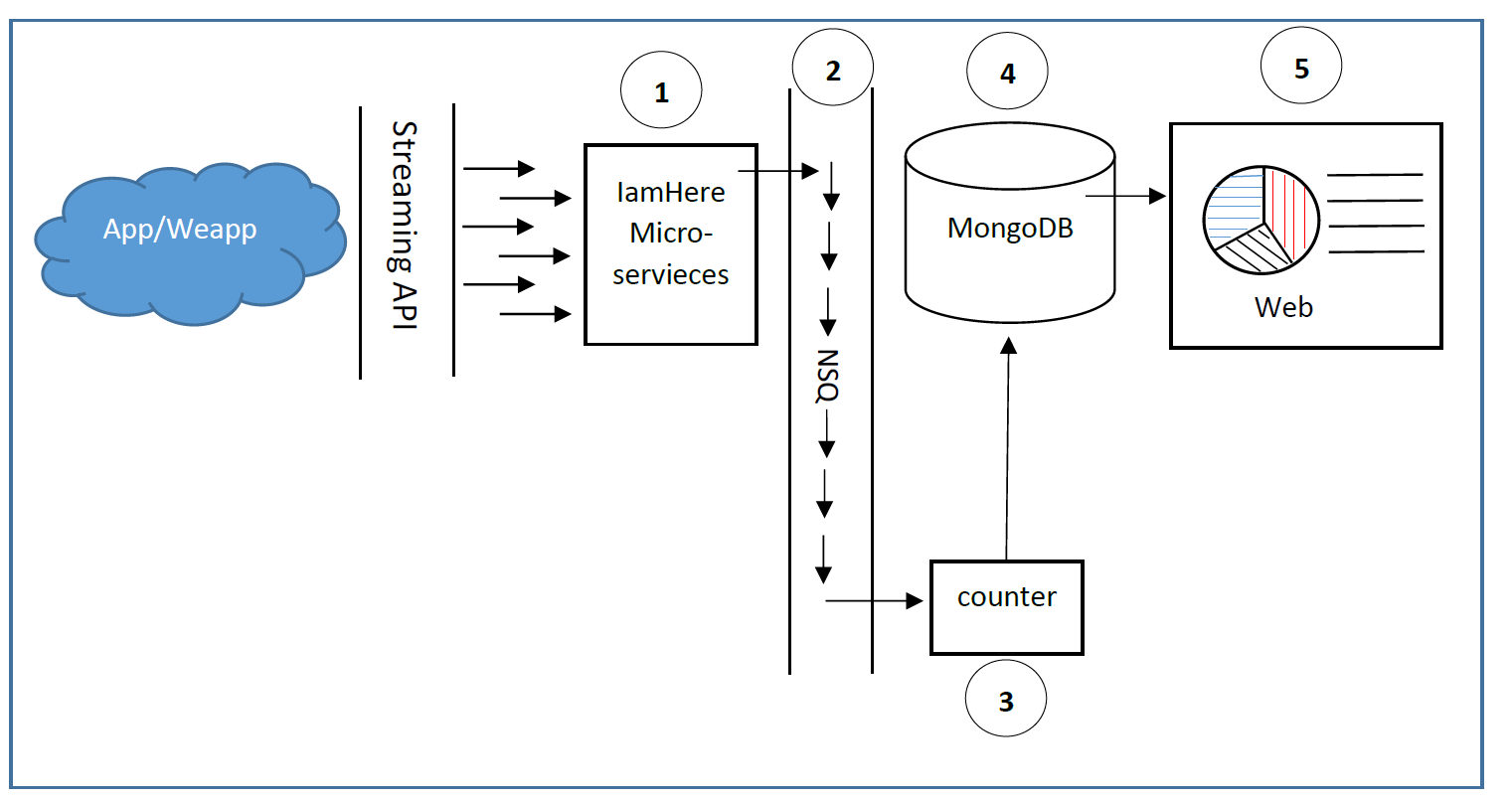
## 1.1 Iamhere一期后台服务技术说明：

* 后台服务基于golang语言开发；
* 支持HTTP连接方式（HTTPS连接协议后期会开发）；
* 采用面向资源的RESTful形式接口；
* 后台数据库采用MongoDB database；
* 采用NSQ处理消息队列，便于后期分布式数据库设计；
* 采用google cloud platform 云服务托管，静态ip地址访问（35.187.154.122）；

## 1.2 后台服务架构设计

一期后台服务架构设计如图1，

* ①：服务器API服务，包括登录，认证，后台数据请求等所有和前端交互的服务；
* ②：NSQ消息队列，用于后期分布式数据库扩展；
* ③：从NSQ消息队列获取消息并周期性写入到数据库中；
* ④：数据库
* ⑤：web服务程序，用于显示后台数据

****

## 2 接口说明

### 2.1查询附近区域

#### 2.1.1输入输出需求：

输入参数：用户坐标

输出参数：附近区域若干，包括区域中心坐标及半径。

#### 2.1.2数据表设计

“area”

{

"\_id": "random string",//string

"name": "Name of this area", //string

“address1”:“Address line 1”, //string

“address2”:“Address line 2”, //string

“category”:“tag of this area for quick sorting” //int

“type”: “official defined or user difined”, //int

"latitude": “latitude of the center of this area” //double

"longitude": “longitudeof the center of this area” //double

"altitude": “altitudeof the center of this area” //double

"radius": “radius of this area” //double

}

#### 2.1.3 接口设计

|  |  |  |
| --- | --- | --- |
| **Request** | **Description** | **Use case** |
| GET /areas | Read all areas | Show a list of areas to the users |
| GET / areas /{id} | Read the area | Show details or results of a specific areas by id |
| GET / areas /{type} | Read the areas | Show details or results of some areas by type |
| GET / areas /{category} | Read the areas | Show details or results of some areas by category |
| GET / areas /{ latitude, longitude, altitude, radius} | Read the areas | Show a list of areas to the users by specify its center coordinates and radius |

### 2.2留言接口

#### 2.2.1 输入输出需求

输入参数：用户名，留言内容，坐标，留言公开时间，用户输入地址，微信ID。

Sample：

curl --data '{"areaid":"test","userid":"test","content":"This is a test messages","timestamp":123,"userdefaddr":"test","expirytime":123,"latitude":123,"longitude":123,"altitude":123}' -X POST [http:// hostname:8080/messages/?key=abc123](http://localhost:8080/messages/?key=abc123)

\*\*\*hostname: hosstname将会替换成静态**35.187.154.122** （下同）

\*\*\*key：abc123仅用于当前测试（下同）

输出参数：调用成功/失败. 限制调用次数，每个用户每天可调用若干次。可用微信ID判别。

Sample:

{"code":0,"reasone":"success","data":null}

Or

{"code":-1,"reasone":"wrong APIkey","data":null}

删除消息：

curl -X DELETE [http://hosstname:8080/msg/541727b08ea48e5e5d5bb189?key=abc123](http://localhost:8080/msg/541727b08ea48e5e5d5bb189?key=abc123)

Sample:

{"code":0,"reasone":"success","data":null}

Or

{"code":-1,"reasone":"wrong id","data":null}

541727b08ea48e5e5d5bb189为要删除的消息的id

#### 2.2.2数据表设计

“message”

{

"\_id": "random string",//string

"areaId": "Area of this msg was kept", //string

"userId": "User (or wechat ID) of this msg was made by", //string

“content”:“message content”, //string

“timestamp”:“timestamp of this message was kept” //long

“userDefAddr”: “address of user defined”, //string

“expiryTime”: “Keep this msg invisible until its expired”, //long

"latitude": “latitude of the center of this area” //double

"longitude": “longitudeof the center of this area” //double

"altitude": “altitudeof the center of this area” //double

}

#### 2.2.3接口设计

|  |  |  |
| --- | --- | --- |
| **Request** | **Description** | **Use case** |
| GET /messages/ | Get all messages list | Get all messages list  获取当前所以消息数据，以json数组形式返回，目前仅用作调试 |
| POST / messages/{  areaId= Mandatory,  userId= Mandatory,  content= Mandatory,  timestamp=optional,  userDefAddr=optional,  expiryTime= optional,  latitude = Mandatory,  longitude = Mandatory,  altitude = Mandatory,  } | Create a new msg | Create a new msg  Mandatory表示必填字段；optional表示可选字段；  timestamp=NULL表示填入服务器记录该条消息时间作为时间戳；  expiryTime=NULL,标明立即公开该条消息，即默认的留言保密过期时间同timestamp |
| DELETE /messages/{\_id= Mandatory} | Delete a msg | Delete a message specified by its id |

### 2.3查看留言接口

#### 2.3.1输入输出需求

输入参数：用户坐标 /区域ID （探索世界里面需要以区域ID检索，如何限制以区域检索的次数是一个需要考虑的问题）

Sample：

curl -v --data '{"areaid":"test”}' -X GET http://hosstname:8080/messages/?key=abc123

或者

curl -v --data '{"latitude":123,"longitude":123,"altitude":123}' -X GET http:// hosstname:8080/messages/?key=abc123

输出参数：该区域留言若干。可重复调用，每次返回不同的留言。

Sample：

curl -X GET [http://hosstname:8080/messages/?key=abc123](http://localhost:8080/messages/?key=abc123)

Result of success:

{"code":0,"reasone":"success","data":[{"id":"5b6548efe8cb917741ca61b4","areaid":"test","userid":"test","content":"This is a test messages","timestamp":123,"userdefaddr":"test","expirytime":123,"latitude":123,"longitude":123,"altitude":123,"apikey":"abc123"},{"id":"5b654c66e8cb917c749ed8d9","areaid":"test","userid":"test","content":"This is a test messages","timestamp":123,"userdefaddr":"test","expirytime":123,"latitude":123,"longitude":123,"altitude":123,"apikey":"abc123"}]}

Result of failure:

{"code":-1,"reasone":"wrong paramter","data":null}

#### 2.3.2接口设计

|  |  |  |
| --- | --- | --- |
| **Request** | **Description** | **Use case** |
| GET /messages/{  (areaId= optional)/(  latitude = optional,  longitude = optional,  altitude = optional)  } | Show a list of specific area messages | Show a list of specific area messages  Mandatory表示必填字段；optional表示可选字段；  areaId/coordiantes：其中一个是必选项 |

2.4查看区域接口（探索世界功能）

#### 2.4.1输入输出需求

输入参数：区域级别（1级，2级），区域ID(当查询2级区域列表时需要输入1级区域ID，当查询1级区域列表时不需要该参数)

输出参数：区域列表（当输入为1级时输出大区列表，当输入为2级时输出该大区区域列表）

考虑加入区域列表更新查询接口，做版本管理，减少后台访问次数。