****Halloway服务器端接口列表****

Halloway/鞠昊东

#### ****创建事件****

接受除了id，EventId之外所有的字段，

获取createId对应的roleName对应的最大事件数量，若超出，则返回@免费用户仅能创建3个事件，付费请联系xxxxxxx

存入数据库中。在relation表中记录事件id，用户id，Create。

#### ****参加一个事件****

传入 UserId, EventId, showContact，Message

首先确认， 用户id和事件id的关系，若是member，则返回 @已经在活动中了，无需参加， 若是Checking，则返回 @正在审核中，无需重复提交审核。 若是Create，则返回@不可参加自己创建的事件。

然后确认，事件中人数是否已满，如果人数已满则返回@参加人数已满。

在确认用户和事件没有关系后，获取事件的NeedPermission属性，

若NeedPermission为否，判断 showContact是否为真，若showcontact为真，则将用户，事件id，Member，用户表中的Contact以Key：Value;记录到数据库中，返回@参加成功， 若showcontact为假，则将用户，事件id，member记录到数据库中，返回@参加成功。

若NeedPermission为真，则重复showContact的判断流程，

记录用户，事件ID，Checking，和message到数据库中。

返回@good

#### ****返回一个事件的信息****

**接受事件id**

**添加属性，Members=[user],**

**通过Relation表收集事件的成员(where Type=’member’ or Type=’create’)，并将成员对应的ShowContact和Message属性加入user，最后在事件对象中添加成员信息集合。**

**返回{“EventInfo”:[]“Members”:[]}**

#### ****返回所有事件****

**按照返回一个事件的标准，返回所有事件。**

#### 返回和一个用户有关的所有事件

接受Userid

返回的值将被分为四个部分，分别储存在一个数组的0,1,2,3位置

其中分别代表了 用户创建的事件， 用户参加的事件， 用户申请加入的事件， 其他用户发来的申请。

首先读取Relation表中UserId所对应的所有事件。按照Type字段的值有分别处理。若是Member, 则放在数组的第二位，若是Checking，则放在数组的第一位。若是Create， 则放在数组的零位。而对于每一个关系为Create的事件，获取事件id所对应的所有Type为Checking的用户Id，对于每个id，分别把事件存储到数组的第四位，并在存储的时候将这个UserId存入额外添加的inUser属性中。

#### 通过审核

传入：事件Id，用户Id

首先判断人数是否已满，如果人数已满，返回@参加人数已满。

在人数未满的情况下，将Relation表中事件ID和用户ID对应的Relation，Type改为Member.

返回good

#### 退出事件

传入：事件Id，用户Id

判断事件Id和用户Id的关系，若不为Create，删除这一行，若无关系，返回@错误，您不在本事件中。

删除后返回good

#### 删除事件

传入：事件id，用户id

验证用户id是否和事件id的relation为Create，如果是Create，则删除事件表中事件id对应的事件，删除Relation表中事件id对应的Relation。

#### 搜索事件

传入：\*关键字，\*类型，\*人数，\*城市，\*国家，\*时间

若输入为空，则返回全部事件。

各个条件的结果为与关系，必须满足全部条件的才可以输出。

\*关键字：事件名中包含\*关键字。

\*类型：事件的类型等于类型。

\*人数：事件的最大人数-现有人数大于等于人数。

\*城市：事件的城市等于城市。

\*国家：事件的国家等于国家。

\*时间：事件结束的时间（endtime）大于等于\*时间

#### 获取成员列表

传入：事件id，

返回事件的成员列表

#### 更新或保存用户信息

传入：用户id，昵称，\*性别，\*[联系方式…as K:V;]，\*角色

使用传入的信息更新或保存用户id所对应的表。

#### 获取用户OpenID

传入 授权码，

返回 OpenID

#### 获取当前地址

传入 经度，纬度

返回，地址列表

获取当前经纬度

#### 传入地址，

返回经度，纬度