GeoHash

场景

- 查找车辆附近的加油站，如果车和加油站距离在2 miles以内，则查找成功。
- Nearby 问题

API

POST /place

Request: body = {

place_name: "Julia's Kitchen",

latitude: 85,

longitude: 10,

category: "Restaurant",

description: "Best Italian restaurant",

photos: ["url1", "url2", "url3"] (Array of photos)

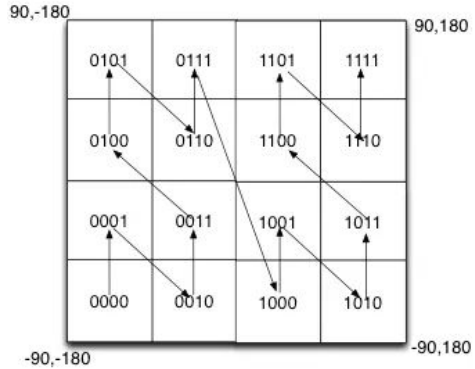
}

Response: Response of the newly created place with place_id

基本原理

GeoHash是一种地址编码方法，能够把二维的空间经纬度数据编码成一个字符串。

经度范围是东经180到西经180，纬度范围是南纬90到北纬90，我们设定西经为负，南纬为负，所以地球上的经度范围就是[-180， 180]，纬度范围就是[-90，90]。如果以本初子午线、赤道为界，地球可以分成4个部分。



Geohash算法步骤 - 1 将经纬度变成二进制。

点（39.923201，116.390705）

最后纬度的二进制表示为：

10111000110001111001

同理，经度116.390705的二进制表示为：

11010010110001000100

	纬度范围	划分区间0	划分区间1	39.928167
1	（-90,90）	（-90,0）	（0,90）	1
2	（0,90）	（0，45）	（45,90）	0
3	（0，45）	（0,22.5）	（22.5，45）	1
4	（22.5,45）	（22.5，33.75）	(33.75, 45)	1
5	(33.75,45)	(33.75, 39.375)	(39.375,45)	1
6	(39.375,45)	(39.375,42.1875)	(42.1875,45)	0
…	…	…	…	…
15	(39.92431640625,39.935302734375)	(39.924316406625,39.9298095703125)	(39.9298095703125, 39.935302734375)	0

Geohash算法步骤 - 2 将经纬度的二进制合并

经度占偶数位，纬度占奇数位，注意，第0位也是偶数位。

11100 11101 00100 01111 00000 01101 01011 00001

Geohash算法步骤 - 3 通过Base32编码表编码

Base32编码表的其中一种如下，是用0-9、b-z（去掉a, i, l, o）这32个字母进行编码。具体操作是先将上一步得到的合并后二进制转换为10进制数据，然后对应生成Base32码。将5个二进制位转换成一个base32码。

上述例子：

wx4g0ec1

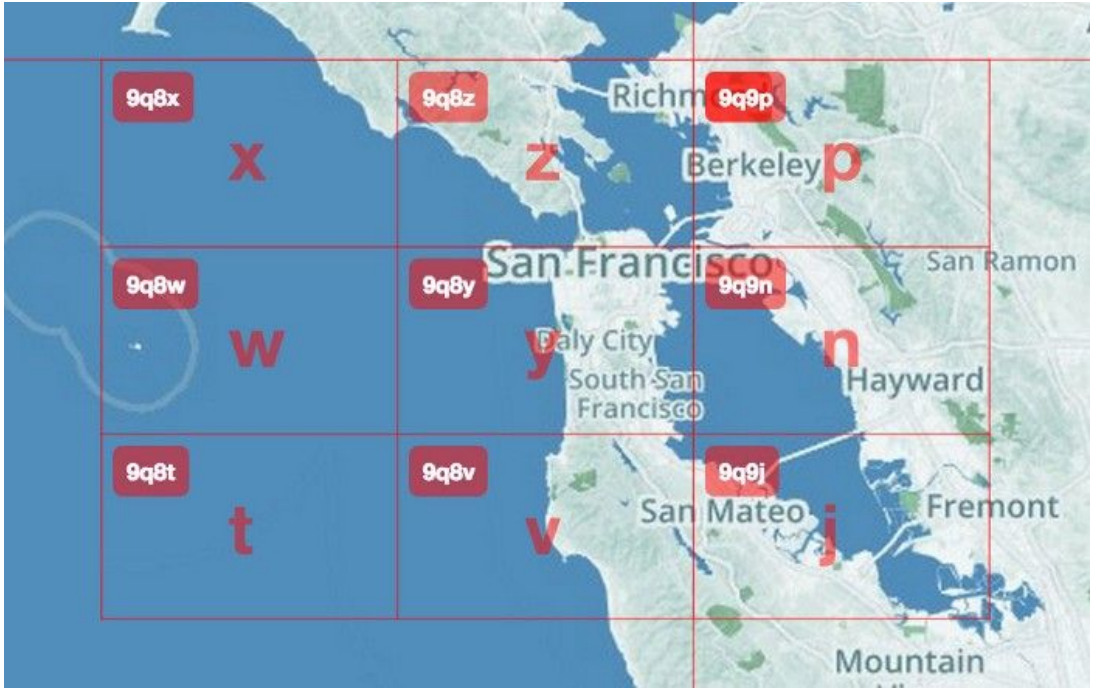
Geohash算法 - 特点

Geohash比直接用经纬度的高效很多，而且使用者可以发布地址编码，既能表明自己位于某处附近，又不至于暴露自己的精确坐标，有助于隐私保护。

- GeoHash用一个字符串表示经度和纬度两个坐标。在数据库中可以实现在一列上应用索引(某些情况下无法在两列上同时应用索引)
- GeoHash表示的并不是一个点，而是一个矩形区域
- GeoHash编码的前缀可以表示更大的区域。例如wx4g0ec1，它的前缀wx4g0e表示包含编码wx4g0ec1在内的更大范围。这个特性可以用于附近地点搜索

Geohash算法 - 特点

编码越长，表示的范围越小，位置也越精确。因此可以通过比较GeoHash匹配的位数来判断两个点之间的大概距离。



Geohash算法 - 注意

1. 边缘问题 - 找附近所有八个点
2. 编码相近的可能距离很远 - 实际计算距离

Demo

1. http://geohash.co/ - 9q8yy7
2. Geohash Visualization