

# Coding task: World of Airports

We'd like you to create an application that pulls data about airports from a Cloudant database (listed below) and displays their location in a simple list. The application should accept a user provided radius and a user provided lat/long point. The result should be a list of airports (with their locations) which are within the input radius of the input point, sorted by distance (sorting can be done client side). When you implement the calculation of the distance please keep in mind the Earth is a globe.

The task should be carried out by using a language of your choice, although you are encouraged to use a language which is supported by the existing Client libraries:

<https://console.bluemix.net/docs/services/Cloudant/libraries/supported.html#supported-clientlibraries>

The goal of this whole task: we would like you to show us that you can write code that is simple, clean, well structured and easy to understand. We're not looking for a snazzy UI, nor are we looking for any extensions in functionality beyond what is stated in the first paragraph.

We need to be able to run your code, so please use nothing but standard libraries and possibly a Cloudant client library. Provide clear documentation as to how to install/build/run your code. Host your app's code on a source code hosting service (e.g. Github or Bitbucket) where we can access it.

---

The database is world readable, at: <https://mikerhodes.cloudant.com/airportdb>. Explore with Curl or similar. This means you need not worry about credentials in your code.

We have set up a search index to return data about airports in a given bounding box. Here is an example query URL:

```
https://mikerhodes.cloudant.com/airportdb/_design/view1/_search/geo?q =lon:[0%20TO%2030]%20AND%20lat:[0%20TO%205]
```

When not URL encoded, this query is `lon:[0TO30]ANDlat:[0TO5]`. This query returns data about airports in the box latitude: 0.0 > 5.0 longitude: 0.0 > 30.0. Further information on search indexes is at <https://console.bluemix.net/docs/services/Cloudant/api/search.html>. The returned data is JSON:

```
{
  "total_rows": 32,
  "bookmark": "g1AAAAEPeJzLYWBgYm1gTmGQSULKzi9...", "rows": [
    {
      "id": "a0487237c4362b941f7332d7eb78cbc6", "order": [
        1.4142135381699,
        3 ],
      "fields": {
        "lat": 4.157639, "lon": 21.650917, "name": "Kotakoli"
      }
    }, ...
  ]
}
```