WCC

Congratulation for reaching this stage of the recruitment process!

The next step will be completing and discussing a Java test.

Below you’ll find the test and as well as an explanation of the test.

Java test challenge:

Write a REST (-y) service that will return the geographic (straight line) distance between two postal codes in the UK.

Arguments to a request are two UK postal codes (you may decide how these arguments are provided).

Result to a valid request must be a json document that contains the following information:

For both locations, the postal code, latiude and longitude (both in degrees);

The distance between the two locations (in kilometers);

A fixed string 'unit' that has the value "km";

For postal codes lookup: use the following data.

http://www.freemaptools.com/download-uk-postcode-lat-lng.htm;

http://www.freemaptools.com/download/full-postcodes/postcodes.zip;

http://www.freemaptools.com/download/full-postcodes/fullukpostcodes.zip.

(You are free to use a database, as long as you give instructions on how to set one up. You may also use the csv file if you find that more convenient).

TECHNOLOGY REQUIREMENTS:

Use any technology you like, as long as it runs on the JVM. Obviously, if you want to showcase your knowledge of spring, you'll want to use spring. Be prepared to explain every bit of code/configuration you submit.

The submitted solution should include a simple way to build (maven preferred) and run on either the command line or from eclipse.

BONUS FEATURES:

* Unit tests!

* Updating postal codes - add REST calls to query and updated the postal-codes -> coordinates mapping. Obviously, these need to be persisted;

* Request logging (log the two post codes in the request; preferably in some way so we can later aggregate and report easily);

* Authentication - restrict the service to those who know a username/password combination;

(You don't have to do this -- but it will give you more opportunities to show off your knowledge, and give us more to talk about).

USEFUL CODE:

This bit of Java code computes (an approximation of) the distance between two points on the planet, given as long/lat pairs (in degrees).

```
Private final static double EARTH_RADIUS = 6371; // radius in kilometers

Private double calculateDistance(double latitude, double longitude, double latitude2, double
longitude2) {

    // Using Haversine formula! See Wikipedia;

    double lon1Radians = Math.toRadians(longitude);

    double lon2Radians = Math.toRadians(longitude2);

    double lat1Radians = Math.toRadians(latitude);

    double lat2Radians = Math.toRadians(latitude2);

    double a = haversine(lat1Radians, lat2Radians)
      + Math.cos(lat1Radians) * Math.cos(lat2Radians) * haversine(lon1Radians, lon2Radians);

    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
```

```
        return (EARTH_RADIUS * c);

    }

    private double haversine(double deg1, double deg2) {
        return square(Math.sin((deg1 - deg2) / 2.0));

    }

    private double square(double x) {
        return x * x;

    }
```

If you have any questions, please contact us!

We wish you success in finalizing the test.