

Java Assessment

Write a REST 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 of a valid request must be a JSON document that contains the following information:

1. For both locations, the postal code, latitude and longitude, both in decimal degrees.
2. The distance between the two locations in kilometers.
3. A fixed string 'unit' that has the value "km".

For postal codes lookup use the data provided on the following website.

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

You are free to use a database, as long as you give instructions on how to set it 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, use Spring.

Be prepared to explain every bit of code/configuration you submit.

The submitted solution should include a simple way to build (maven is preferred) and run on either the command line or from within IntelliJ IDEA or as a standalone application.

Bonus Features

You do not have to do the following points, but each one will give you more opportunities to show off your knowledge and give us more to talk about.

- Unit tests.
- Updating postal codes' coordinates.
- Request logging (log the two post codes in the request; preferably in a way that we can later aggregate and report easily).
- Authentication - restrict the service to those who has a username/password combination.

Useful Code

This bit of Java code computes (an approximation of) the distance between two points on the planet, given as longitude/latitude pairs in decimal 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 degree1, double degree2) {
    return square(Math.sin((degree1 - degree2) / 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 assessment.