Advanced Programming

Lab Sheet 21 – Consuming Web Services

Introduction

A web service is a common architectural pattern used to separate *front end* (presentation of data to users) and *back end* (processing and storage of data) operations in enterprise-grade systems. In these exercises, you are going to create a simple application that uses a web service, which is normally referred to as consuming a web service. You'll be creating web services in your lab sessions next week.

The government's Food Standards Agency makes the results of restaurants' (and other businesses serving food) hygiene inspections [1] publicly available (see, for example, here). These results are the same one-to-five scores displayed on stickers in establishments' doors and windows. I have collected all ~400,000 ratings available for establishments in England and Wales, and placed them in a MySQL database. I have written a web service using the PHP scripting language that will query the database and return inspection results in JSON format.

The web service supports several different ways of searching for eateries, but in these exercises we'll limit ourselves to searching by (partial) postcode.

The Web Service

To use the web service, you must supply it with the first few (minimum 3) characters of the postcode in which you wish to find establishments. You pass this information as part of the URL, as depicted in the examples below:

```
Hygiene ratings of establishments near campus:
http://sandbox.kriswelsh.com/hygieneapi/hygiene.php?op=search_postcode&postcode=M1+7D

Hygiene ratings of establishments near Buckingham Palace:
http://sandbox.kriswelsh.com/hygieneapi/hygiene.php?op=search_postcode&postcode=SW1A+1AA

Hygiene ratings of establishments in Llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch:
http://sandbox.kriswelsh.com/hygieneapi/hygiene.php?op=search_postcode&postcode=LL6_
```

The web service returns data on up to 10 establishments whose postcode starts with the specified characters in a JSON-based data format, as depicted in Fig. 1, below:

```json
[
    {
        "id": "184292",
        "BusinessName": "Manchester Metropolitan University",
        "AddressLine1": "John Dalton Building",
        "AddressLine2": "Chester Street",
        "AddressLine3": "Manchester",
        "PostCode": "M1 5GD",
        "RatingValue": "5",
        "RatingDate": "2017-06-08",
        "Location": {
            "Latitude": "53.472326",
            "Longitude": "-2.23959"
        }
    },
    {
        "id": "200439",
        "BusinessName": "The Courtyard",
        "AddressLine1": "The Courtyard",
        "AddressLine2": "2 Chester Street",
        "AddressLine3": "Manchester",
        "PostCode": "M1 5GE",
        "RatingValue": "3",
        "RatingDate": "2018-09-27",
        "Location": {
            "Latitude": "53.4720703",
            "Longitude": "-2.2414102"
        }
    }
]
```

Figure 1: JSON Format of Returned Data

TASK 1: RETRIEVING DATA FROM THE WEB SERVICE

Although we eventually want to create an application that allows the user to enter a partial postcode to search for restaurants using the web service, it would make sense to start with the code that is responsible for actually retrieving that data.

Create a class named HygieneServiceClient, and give it a retrieveRatings(String postcode) method. For now, ignore the String parameter, and hard-code the method so that it can retrieve the data from the web service using one of the example URLs (or another postcode, if you prefer). The method should return a List (ArrayList) of Restaurant Objects, containing (at least) a business name, address, and hygiene rating. This will tie together learning from various different topics across the unit so far, including:

1. I/O

2. Data Interchange Formats

3. JSON Parsing

4. Exception Handling

5. Collections

The basic idea here is to create a URL object pointing to the correct web address, call .openConnection() on it, get the InputStream from the returned URLConnection, and then use a BufferedReader to gather all of the returned data into a String. Once completed, you can use the JSON reference parser to obtain the values for each of the properties in your Restaurant class.

To test your work so far, print out the name, address, hygiene rating and inspection date of each of the establishments for a particular (partial) postcode.

Task 2: Allowing Custom Searches using a GUI

To make the application truly useful, you will need to create a UI that allows the user to enter a partial postcode, and that displays the result. I would recommend displaying results in a JTextArea in the first instance, although the extension task below suggests formatting the output more professionally using a JTable. You will need to create a HygieneRatingUI class that creates a JFrame adorned with appropriate widgets to allow the user to find the hygiene ratings of restaurants with their chosen (partial) postcode.

You will need a JTextField to allow the user to enter their chosen postcode, a JButton for the user to click when they are ready to search, as well as your chosen method of returning the results. You will also need to adjust the retrieveRatings() method to construct the correct URL for the postcode entered, instead of using the hard-coded one from earlier.

Finally, because connecting to the web service to perform the search and retrieve results takes a few seconds, you should perform the search in a background thread, instead of locking the entire UI for however long it takes to connect to and query the web service.

Extension Task: Output Formatting

There is significant scope to improve the user interface of your application, so consider replacing the JTextArea with a JTable that displays all of the information returned by the web service about each result, as depicted in Fig. 1. This program will be similar to the work you did last week in lab 20, fetching data from the web service instead of from a database. You'll likely need the JTable to scroll through results too.

You could even use the official FSA hygiene rating images [1] with the green and black rating numbers to illustrate the rating of each establishment.

References

[1] Food Standards Agency, "Food Hygiene Rating Scheme". Available: https://www.food.gov.uk/safety-hygiene/food-hygiene-rating-scheme