Java Official Joke Web API HashMap Tutorial

Contents

Time required: 60 minutes

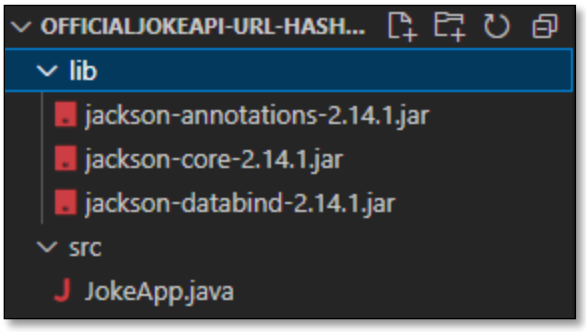
- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.
- Please read all the directions before beginning the assignment.

Jackson JSON Libraries

We will be retrieving data from the web in JSON format. Java doesn't have any native JSON parsing. We are going to add three libraries to parse out the JSON data into Java classes.

1. Download these three files.
    a. jackson-annotations-2.16.1.jar
    b. jackson-core-2.16.1.jar
    c. jackson-databind-2.16.1.jar
2. Create a folder for your **JokeHashMap** program.
3. Create a folder inside **JokeHashMap** named: **lib**
4. Create another folder called **src**
5. Copy and paste the three jar files into the **lib** folder.
6. In the **src** folder, create **JokeApp.java**
7. Double Click **JokeApp.java**

8. You should see the following at the bottom left if you open **Java Projects → Referenced Libraries**



Joke Program

This program which uses the Jackson JSON libraries to map the json data to a Java HashMap object. A HashMap is very much like a Python dictionary. It uses key value pairs to access and retrieve data.

```java
/*
 * Filename:    JokeApp.java
 * Written by:  William Loring
 * Written on:  12/25/2022
 * Revised:
 * Description: Official Joke API using Java HashMap
 * A HashMap is like a Python dictionary using key value pairs
 */
// Catch any io excetions
import java.io.IOException;
import java.net.URI;
//Java libraries to retrieve http information from a URL
import java.net.URL;
// Import Jackson JSON libraries
import com.fasterxml.jackson.core.JsonGenerationException;
import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.JsonMappingException;
import com.fasterxml.jackson.databind.ObjectMapper;
// Libraries to use HashMap
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class JokeApp {
    static Scanner keyboard = new Scanner(System.in);
    static HashMap<String, Object> dataHashMap = new HashMap<String, Object>();
    final static String API_URL = "https://v2.jokeapi.dev/joke/Any?type=twopart&safe-mode";
    static String menu = "y";

    Run | Debug
    public static void main(String[] args) {
        while (!menu.equals("n")) {
            getData();
            // For debugging, print out entire HashMap
            // System.out.println(dataHashMap.toString());

            // Display a single Cat Fact
            System.out.println(dataHashMap.get("setup"));
            System.out.println(dataHashMap.get("delivery"));

            System.out.print("Another Joke (y, n): ");
            menu = keyboard.nextLine();
        }
        System.out.println("Thanks for using the Official Joke App.");
    }
```

```
47      // -------- Get data from the API endpoint ----------------------- //
48      static void getData() {
49          try {
50              // Create URI object
51              URI uri = new URI(API_URL);
52
53              // Create URL object from URI
54              URL url = uri.toURL();
55
56              // Create Jackson JSON object to read the raw JSON data
57              // into POJO (Plain Old Java Objects)
58              ObjectMapper pojoMapper = new ObjectMapper();
59
60              // Read POJO into dataHashMap Object
61              dataHashMap = (HashMap<String, Object>) pojoMapper.readValue(
62                      url,
63                      new TypeReference<Map<String, Object>>() {
64                      });
65
66          } catch (JsonGenerationException | JsonMappingException e) {
67              // JSON Exception handling
68              // e.printStackTrace() will print the exception
69              e.printStackTrace();
70              // Input Output Exception handling
71          } catch (IOException e) {
72              // Handle any other exceptions
73              // e.printStackTrace() will print the exception
74              e.printStackTrace();
75          } catch (Exception e) {
76              // e.printStackTrace() will print the exception
77              e.printStackTrace();
78          }
79      }
80  }
```

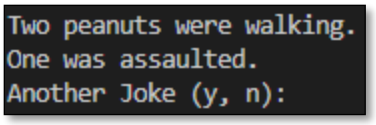
The above code does 3 things.

1. Creates a Jackson object to read JSON into POJO (Plain Old Java Objects).
2. Reads the https JSON into a HashMap object.
3. Displays a Joke.

Time for a Joke

Time to run your new application.

Example run:



Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.