

Chrome 文件 编辑 视图 历史记录 书签 个人资料 标签页 窗口 帮助

hackerrank.com/test/fccaonsb53/questions/3i8rbip6cng

1h 44m left

BETA Can't read the text? Switch theme

3. REST API: Relevant Food Outlets

ALL

A REST API contains information about food outlets across multiple cities. Given the city name, and maximum cost for 2 persons, the goal is to use the API to get the list of food outlets that belong to this city and have an estimated cost less than or equal to given cost. The API returns paginated data.

To access the information, perform an HTTP GET request to:
`https://jsonmock.hackerrank.com/api/food_outlets?city=<city>&page=<pageNumber>`
where `<city>` is the city to get the food outlets for and `<pageNumber>` is an integer that denotes the page of the results to return.

For example, a GET request to
`https://jsonmock.hackerrank.com/api/food_outlets?city=Seattle&page=2`
returns data associated with city `Seattle`, and on the second page of the results.

Similarly, a GET request to
`https://jsonmock.hackerrank.com/api/food_outlets?city=Houston&page=1`
returns data associated with city `Houston`, and on the first page of the results.

Language: java 7

```
1 > import java.io.*; ...
10
11 class Result {
12
13     /*
14      * Complete the 'getRelevantFoodOutlets' function below.
15      *
16      * URL for cut and paste
17      * https://jsonmock.hackerrank.com/api/food_outlets?city=<city>&page=<pageNumber>
18      *
19      * The function is expected to return an array of strings.
20      *
21      * The function accepts a city argument (String) and maxCost argument(Integer).
22      */
23
24     public static List<String> getRelevantFoodOutlets(String city, int maxCost)
25     {
26
27     }
28 }
29
30 > public class Solution { ...
```

Test Results Custom Input Run Code Run Tests Submit

Chrome 文件 编辑 视图 历史记录 书签 个人资料 标签页 窗口 帮助

hackerrank.com/test/fccaonsb53/questions/3i8rbip6cng

1h 44m left

The response to such a request is a JSON with the following 5 fields:

- `page`: The current page of the results.
- `per_page`: The maximum number of records returned per page.
- `total`: The total number of records in the database.
- `total_pages`: The total number of pages with results.
- `data`: Either an empty array or an array of outlet objects. Each object has the following schema:
 - `city`: city we queried for where the outlet is located [STRING]
 - `name`: name of the outlet [STRING]
 - `estimated_cost`: estimated cost for 2 persons [INTEGER]
 - `user_rating`:
 - `average_rating`: average rating of the outlet [FLOAT]
 - `votes`: total votes for the outlet [INTEGER]
 - `id`: unique identifier of the outlet [INTEGER]

Below is an example of an outlet object:

```
{
  "city": "Houston",
  "name": "Cocca Tree",
  "estimated_cost": 18,
  "user_rating": {
    "average_rating": 4.5,
    "votes": 10
  }
}
```

Language: java 7

```
1 > import java.io.*; ...
10
11 class Result {
12
13     /*
14      * Complete the 'getRelevantFoodOutlets' function below.
15      *
16      * URL for cut and paste
17      * https://jsonmock.hackerrank.com/api/food_outlets?city=<city>&page=<pageNumber>
18      *
19      * The function is expected to return an array of strings.
20      *
21      * The function accepts a city argument (String) and maxCost argument(Integer).
22      */
23
24     public static List<String> getRelevantFoodOutlets(String city, int maxCost)
25     {
26
27     }
28 }
29
30 > public class Solution { ...
```

Test Results Custom Input Run Code Run Tests Submit

