

Objectives of this Test Case

The main objective of this test case is to gain an understanding of your approach to a relevant software development problem. The test case should take approximately **2-3 hours**, but you will be **given one day** to complete the test case. The test case may serve as a discussion point as we proceed through the interviewing process.

Test Case Description

You will be provided with two web API addresses containing REST JSON data.

- BuildingInfos
- AnalyticsData

The BuildingInfos provides information regarding different buildings (e.g., building_id, country, state, city).

The AnalyticsData provides information regarding user purchase history (e.g., device used, building, items purchased).

The data can be fetched from the following URLs:

<http://jobs.mapsted.com/api/Values/GetBuildingData>

<http://jobs.mapsted.com/api/Values/GetAnalyticsData>

You will be required to write the appropriate network calls and store the results in the appropriate classes/objects/models that you will be required to create.

You will then need to process the data and populate appropriate views, which you are also required to create. The choice of UI/view is completely up to you.

Below are the values you need to address (i.e., the values you need to display in your UI/views):

- Total purchase costs for Samsung manufacture devices
- Total number of times item (item_id = 47) was purchased
- Total purchase costs for item's in the category (item_category_id = 7)
- Total purchase costs in Ontario
- Total purchase costs in the United States
- Which building (name or id) has the most total purchase costs?

Please be sure to layout your code as though this was a much larger project (see assessment criteria below). Be sure to comment your code to ensure it can be easily followed.

Note: If you are pressed for time, it is more important to display fewer views correctly than all views incorrectly.

You will be expected to submit your code (in the language of your choosing) along with an image/screenshot of your final view/UI.

Assessment Criteria:

- Class/Model/Object definitions
- Code organization/file structure
- Code clarity (e.g., layout, comments, coding structure)
- Correctness of values displayed in view/UI
- Clarity of view/UI