



## Coding Challenge

### Your task

You have been asked to develop a travel planning mobile app that allows users to search for public transport service departures.

The app should include the following functionality:

- Retrieve a list of departure results for various modes of transportation (train and bus ) from the provided API endpoint.
- Allow users to filter search results by 'departure time' and 'mode of transportation'.
- Render the departure results as markers on a map.

We expect candidates to come up with their own design and logical solutions for this code challenge including: UI, screen navigations, network service layer and search criteria.

Although you are working with a single endpoint it is important to ensure the API client is extensible and separate URL components. You can use any third-party libraries or pods, however please provide justification.

## Instructions

Please provide relevant source code for your implementation and any documentation or additional assumptions that you feel are appropriate. Please use the following information to assist you in completing the challenge successfully:

- The intent of the challenge is to provide us with an opportunity to judge your problem solving, design and development skills. It is important to provide a solution that highlights your skills in these areas.
- Develop the mobile app in ReactNative or using standard iOS / Android code
- Your solution should be high quality, well annotated, and include tests.
- The simplest solution is often the best. You will be given at least 48 hours notice to complete this task, but it is recommended that no more than 8 hours is spent on the problem.
- If you can't complete the task in 8 hours re-prioritise the functional requirements and deliver a working version of what you have managed to solve.
- The application must run and be easy to build from source. It also must be easy to execute for us to determine if the application meets the above requirements.

## API Endpoint Information

The API endpoint provides departure information for various forms of transportation. The API returns a JSON payload containing trains and bus services that operates at specified locations.

The API endpoint is protected using an API Key. This API key will need to be supplied as a HTTP header with every request made to the API endpoint.

Header name: x-api-key

Header Value

(API Key): 6PSJVe1Jr733JRzxP0uAe3TuqBxRqlbE9f4ua8Wf

HTTP Method: GET

URL:

<https://reypjgj1c.execute-api.ap-southeast-2.amazonaws.com/dev/transportation>

Query Parameter	Description
departureTimeMin (optional)	<p>Filters by date/time; the API endpoint will only return departure results that occur after the specified value.</p> <p>Date/time value is a string value in format 'Y-M-D hh:mm:ss' Eg. 2019-06-19T05:51:31.566Z</p> <p>NOTE: This value must be URL Encoded. Eg. 2019-06-19T05:51:31.566Z becomes 2019-06-19T05%3A51%3A31.566Z</p>

departureTimeMax <b>(optional)</b>	<p>Filters by date/time; the API endpoint will only return departure results that occur before the specified value.</p> <p>Date/time value is a string value in format 'Y-M-D hh:mm:ss'  Eg. 2019-06-19T05:51:31.566Z</p> <p>NOTE: This value must be URL Encoded.  Eg.  2019-06-19T05:51:31.566Z  becomes  2019-06-19T05%3A51%3A31.566Z</p>
typeId <b>(optional)</b>	<p>Filter by mode of transportation; the API endpoint will only return departure for the specified mode of transportation:  0 = Train  1 = Bus</p> <p>typeId is an integer value.</p>

**Sample Request:**

<https://reypig1c.execute-api.ap-southeast-2.amazonaws.com/dev/transportation?departureTimeMin=2024-07-02T23%3A10%3A00.000Z&departureTimeMax=2024-08-08T06%3A00%3A00.000Z&typeId=0>

**Sample Response:**

```
{
  "statusCode": 200,
  "message": "Listing possible transportation available at this time.",
  "result": {
    "transportation": {
      "modes": [
        {
          "typeId": 1,
          "departureTime": "2024-07-07T13:50:00.000Z",
          "route": "624",
          "name": "Queen Victoria Market",
          "latitude": -37.806718,
          "longitude": 144.9574589,
          "hasMyKiTopUp": true
        },
        {
          "typeId": 1,
          "departureTime": "2024-07-07T18:50:00.000Z",
          "route": "625",
          "name": "Carlton Gardens",
          "latitude": -37.8049684,
          "longitude": 144.9572112,
          "hasMyKiTopUp": true
        },
        {
          "typeId": 1,
          "departureTime": "2024-07-08T14:55:00.000Z",
          "route": "820",
          "name": "Chadstone Shopping Centre",
          "latitude": -37.8862515,
          "longitude": 145.0807788,
          "hasMyKiTopUp": true
        }
      ]
    }
  }
}
```

