

StartingPoint

Web Development Skills Package
Stratus Data Systems, Inc

Version 2.10, May 2022

Introduction

Thank you for taking the time to complete the StartingPoint Web Development Skills Package!

We're looking to get a sense of your approach to software development and your experience or potential in the realm of browser-based applications. It is not necessary to complete all of the tasks, so please don't feel as if you need to spend more than an hour working on this package. We would like for you to return the package to us within one or two hours of your receipt.

Specific technologies in use:

- JavaScript <https://developer.mozilla.org/en/JavaScript>
- HTML <https://developer.mozilla.org/en/HTML>
- CSS <https://developer.mozilla.org/En/CSS>
- ExpressJS - <https://expressjs.com/>
- LeafletJS - <https://leafletjs.com/>
- React - react-leaflet - <https://react-leaflet.js.org/>
- React - Chakra-UI - <https://chakra-ui.com/>

Important Components

- api/routes/JSONWebService.js
 - A simple web service that loads JSON files and can return one row or many via a REST API request.
- client/src/components/LeafletMap.js
 - The map displayed in this project
- client/src/components/MTARoutes.js
 - A component that displays a list of Subway Routes within the New York MTA.
- client/src/components/MetaEntrances.js
 - A component that displays a list of Subway Entrances within the New York MTA.

Getting Started

Development Tools Required

- A desktop or laptop computer running an operating system capable of software development, such as Microsoft Windows, Mac OS X, or Linux.
- Recent versions of Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge
- A text editor or code editor.
- A JavaScript/CSS/HTML debugger.
- Tools to unzip and zip a folder (built-in to most modern operating systems).

JavaScript/CSS/HTML debugger

- In Firefox, under Tools / Web Developer, click Show Tools in menu bar.
- In Safari, under Preferences / Advanced, click Show Develop menu in menu bar.
- In Chrome or Edge, under the View menu, click Developer tools.

Getting and Running the Package

1. Download and extract the package into location of your choice.
2. Open the Starting Point folder.
3. Install dependencies

```
cd Starting Point
yarn install
```

4. Begin development

```
yarn run start
```

During Development

Some things that are important

- Clean, well organized code.
- Proper indentation.
- Proper use of JavaScript function and variable naming and capitalization conventions.
- Don't abbreviate. Do use verbose names.
- Comment what's important using descriptive comments, but don't over-comment.

When Complete

1. Create a zipped folder of the entire StartingPoint package, without node_modules, including your additions and changes.
 - In Windows, a zipped folder can be created by right-clicking the folder and selecting Send to followed by Compressed (zipped) folder from the menu.
 - In Mac OS, a zipped folder can be created by right-clicking the folder and selecting Compress "StartingPoint".
2. Return the compressed project to us at - <https://www.stratusdata.com/transfer>
3. Reply to the email letting us know that you have completed the assessment.

Tasks

It is not necessary to complete all of the tasks.

Complete as much as you can within the given time. Please return the package within one or two hours of your receipt.

1 - Add route colors to map

The subway routes displayed on the map do not show the correct color. For each GeoJSON in LeafletMap.js set the GeoJSON feature color.

The colors required for this task may be found within the route_color property in the api/routes response.

To accomplish this task, examine the api/routes response and the api/shapes response. Both API endpoints may be required to complete this.

2 - Correct table headers and columns in table view of Subway Entrances

When a user clicks on a subway route, the top 100 subway entrances for this route are displayed in a table below the route links. Improve the display of this table by retrieving a columnset from the API route api/columnsets/entrances. This API endpoint contains a list of property names to assist formatting the table.

- columns - The columns to display
- columns.name - The property name to display
- columns.displayName - The header value (th) for this column

3 - Add entrances to the map

For each entrance displayed in the table, add a marker to the map using the entrance properties latitude and longitude.

When the user clicks or taps the marker, display the cross street $\{(north_south_street) \times (east_west_street)\}$ in an overlay.

- Icons [react-icons/fa](#), FaTrain

4 - Add pagination buttons (next and previous) to entrances

By default the entrances/route route returns only the top 20 entrances. Some routes have hundreds of entrances. Add Next and Previous buttons using chakra-ui buttons below the result display.

The entrances/route API accepts optional query parameters skip, default 0, and top, default 20. Implement the parameter skip to control the page currently displayed within the results table.

5 - Correct race conditions

An artificial random delay (up to 3 seconds) has been added to all responses returned from the API. Selecting different routes quickly may produce incorrect results due to older results returning after the user's last selected action. Depending on how you implemented Next and Previous buttons a race condition will also be possible there.

In either MTAEntrances or MTARoutes correct the race condition by ensuring that whatever the user last selected is displayed.