COMP3322 MODERN TECHNOLOGIES ON WORLD WIDE WEB

Workshop 2: JavaScript and AJAX

Overview

In this workshop, you will develop a simple Web app, which retrieves real-time MTR train data via the MTR Next Train Open Data API. As shown in Fig. 1, the app provides two drop-down selection lists for selecting the MTR line and the target station on that MTR line.

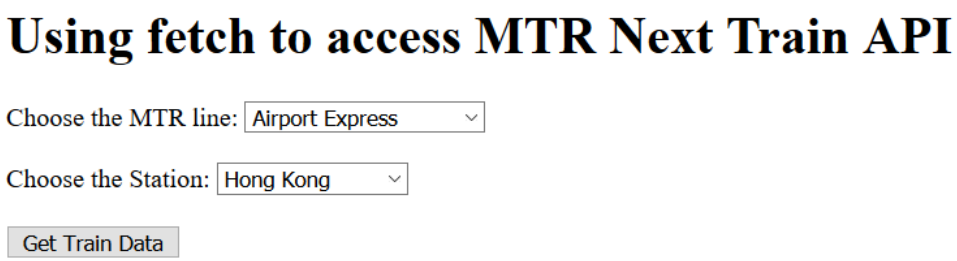

Fig. 1

Instead of listing all stations, you will design a selection menu that lists only the stations on that specific MTR line. For example, Fig. 2. and Fig. 3 show the lists of stations on the Airport Express line and Tseung Kwan O line respectively.

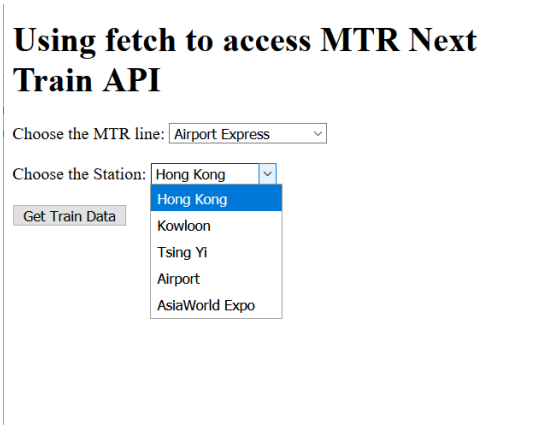

Fig. 2

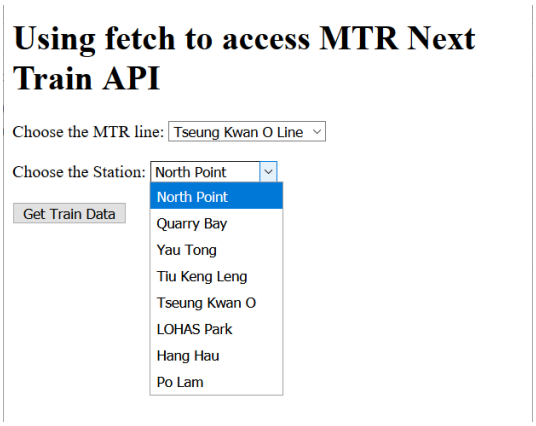

Fig. 3

You will also practice retrieving the data using the fetch() promise call and extract the data from the returned JSON data file. Then present the data beneath the “Get Train Data” button as shown in Fig. 4.

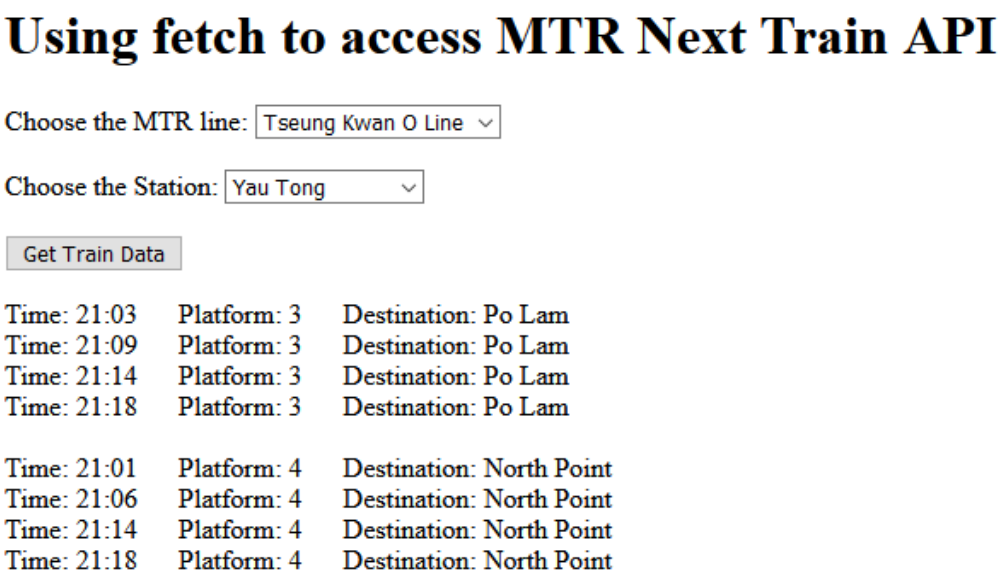

Fig. 4

Task 1: Examine MTR Next Train Open Data

Step1: Visit the following Webpage under the DATA.GOV.HK Website.
https://data.gov.hk/en-data/dataset/mtr-data2-nexttrain-data

Before using the API to access real-time MTR data, please download the two documents: Data dictionary for Next Train API and Next Train API Specification.
https://opendata.mtr.com.hk/doc/Next_Train_DataDictionary_v1.1.pdf
https://opendata.mtr.com.hk/doc/Next_Train_API_Spec_v1.1.pdf

These two documents contain necessary information for you to understand the structure of the returned JSON string and the meanings of various data objects (and terms) carried in the JSON string.

For example, in the API specification document, it gives an example of the response JSON string when the HTTP request is successful (as shown in Fig. 5) and another example of the response when no data is available for that line/station (as shown in Fig. 6).

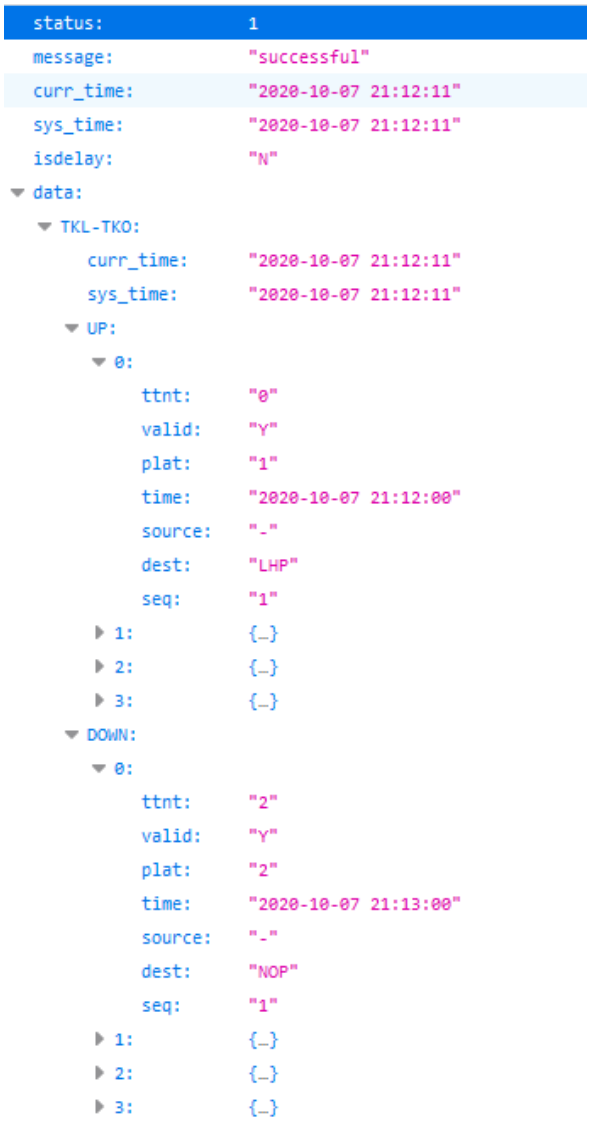

Fig. 5

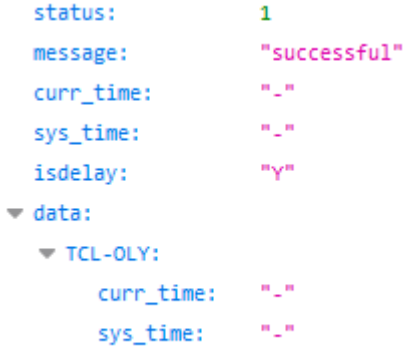

Fig. 6

In the data dictionary document, we can find the required variables for using the API and the meanings of various terms and objects in the returned JSON data. Fig. 7 gives you the screen capture of the description of the UP/DOWN fields.

The below describes the key information within the data array format.		
UP/DOWN	Array format	Indicate the destinations of the train in the specific line. AEL (Airport Express) *UP:* AIR (Airport) / AWE (AsiaWorld Expo) *DOWN:* HOK (Hong Kong) TCL (Tung Chung Line) *UP:* TUC (Tung Chung) *DOWN:* HOK (Hong Kong) WRL (West Rail Line) *UP:* TUM (Tuen Mun) *DOWN:* HUH (Hung Hom) TKL (Tseung Kwan O Line) *UP:* POA (Po Lam) / LHP (LOHAS Park) *DOWN:* TIK (Tiu Keng Leng) / NOP (North Point)

Fig. 7

Step2: Retrieve the MTR data by manually composing the request URL.
Here is the resource URL: https://rt.data.gov.hk/v1/transport/mtr/getSchedule.php
To get the data, you need to provide a query string with two name/value pairs; they are the line parameter and station parameter.

For example, to query the next train arrival schedule of the Austin station on the West Rail line, we need this URL:
https://rt.data.gov.hk/v1/transport/mtr/getSchedule.php?line=WRL&sta=AUS

Copy this URL and paste it to the Chrome browser and then the Firefox browser; you should find that Firefox presents the returned JSON data in a more visually useful way.

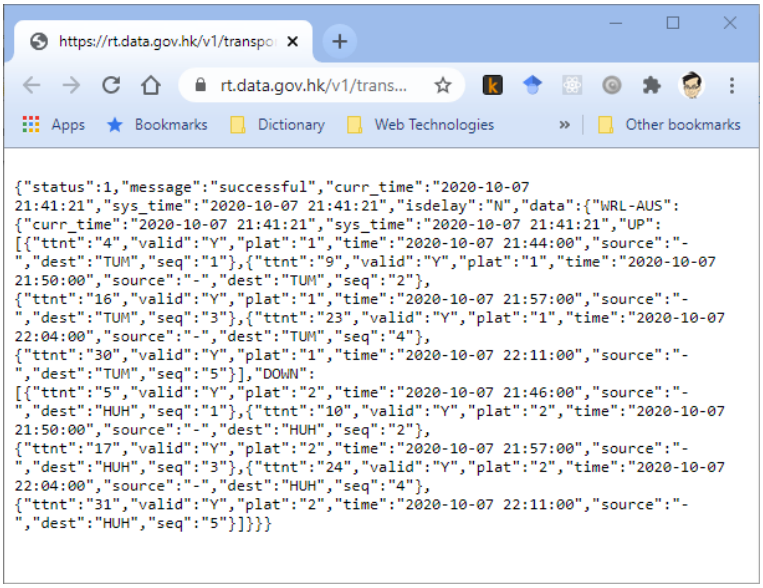

Fig. 8

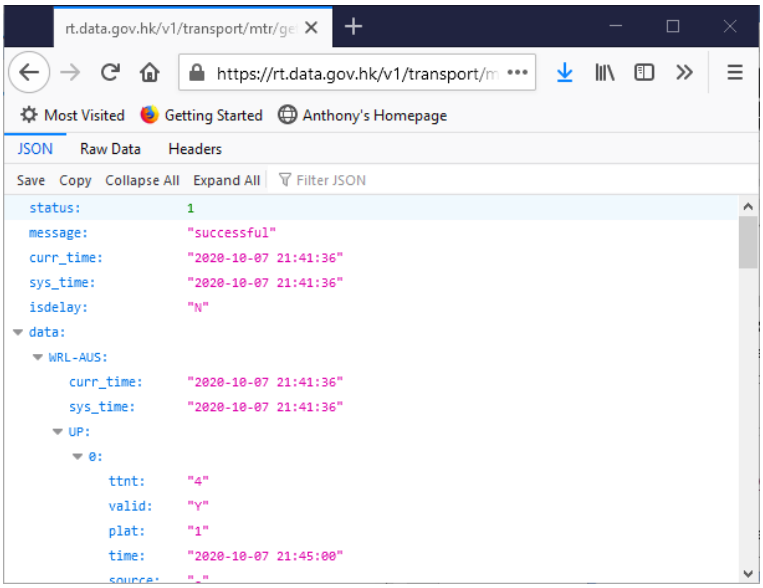

Fig. 9

Use the same mechanism to create more requests to get data on different lines and stations.

Task 2: Implement the program

Download the template file from Moodle- "**index_WS2.html**". Open it with a text editor. You will see it contains the following HTML content:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>Workshop 2</title>
  <style>
  span {
    margin-right: 1.5em;
  }
  </style>
</head>
<body>
  <h1>Using fetch to access MTR Next Train API</h1>
  <label for="line">Choose the MTR line:</label>
  <select name="line" id="line">

  </select>
  <br><br>
  <label for="station">Choose the Station:</label>
  <select name="station" id="station">
```

```
  </select>
  <br><br>
  <button id="bttn">Get Train Data</button>
  <div id="output" style="margin-top: 1rem"></div>

  <script>

  </script>
</body>
</html>
```

When the page is loaded, both the selection lists are empty.

Step 1: Fill in the selection options for selecting the four MTR lines. For example, the following option is for selecting the Airport Express line and it is the default option.

```
<option value="AEL" selected>Airport Express</option>
```

The other options are: 'TCL' for the Tung Chung line, 'WRL' for the West Rail line, and 'TKL' for the Tseung Kwan O line.

Step 2: Fill in the selection options for selecting the five stations on the Airport Express line. For example, the following option is for selecting the Hong Kong station on the Airport Express line.

```
<option class="AEL" value="HOK">Hong Kong</option>
```

To differentiate different stations on different lines, we add the class attribute and set its value to "AEL" for all stations on the Airport Express line.

When done, the two selection lists will look as follows:

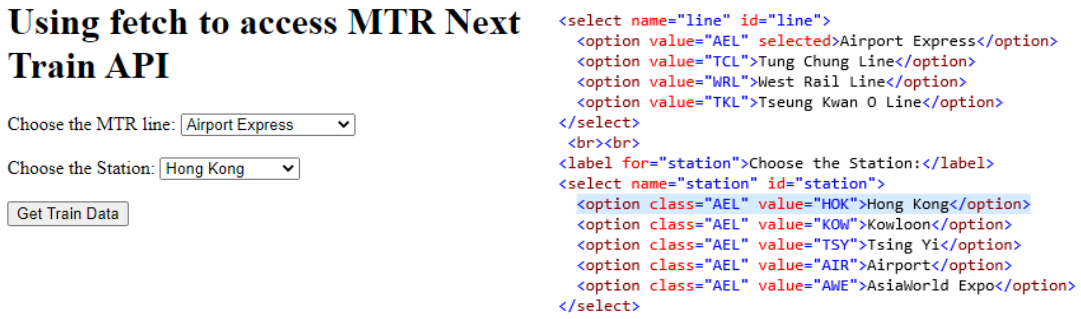

Fig. 10

Currently, we only have the stations on the Airport Express line, we need to use JavaScript to change the set of stations from the AEL set to another set if the user selects another MTR line.

Step 3: We have to build a 'database' which keeps the list of stations associated with each MTR line. This 'database' is an in-memory object with four key/value pairs. Each key maps to the set of station options for each MTR line. Enter the following code to the <script> tags.

```
    const Stations = {
    'AEL': '<option class="AEL" value="HOK">Hong Kong</option> \
    <option class="AEL" value="KOW">Kowloon</option> \
    <option class="AEL" value="TSY">Tsing Yi</option> \
    <option class="AEL" value="AIR">Airport</option> \
    <option class="AEL" value="AWE">AsiaWorld Expo</option>',
    'TCL': '<option class="TCL" value="HOK">Hong Kong</option> \
    <option class="TCL" value="KOW">Kowloon</option> \
    <option class="TCL" value="OLY">Olympic</option> \
    <option class="TCL" value="NAC">Nam Cheong</option> \
    <option class="TCL" value="LAK">Lai King</option> \
    <option class="TCL" value="TSY">Tsing Yi</option> \
    <option class="TCL" value="SUN">Sunny Bay</option> \
    <option class="TCL" value="TUC">Tung Chung</option>',
    'WRL': '<option class="WRL" value="HUH">Hung Hom</option> \
    <option class="WRL" value="ETS">East Tsim Sha Tsui</option> \
    <option class="WRL" value="AUS">Austin</option> \
    <option class="WRL" value="NAC">Nam Cheong</option> \
    <option class="WRL" value="MEF">Mei Foo</option> \
    <option class="WRL" value="TWW">Tsuen Wan West</option> \
    <option class="WRL" value="KSR">Kam Sheung Road</option> \
    <option class="WRL" value="YUL">Yuen Long</option> \
    <option class="WRL" value="LOP">Long Ping</option> \
    <option class="WRL" value="TIS">Tin Shui Wai</option> \
    <option class="WRL" value="SIH">Siu Hong</option> \
    <option class="WRL" value="TUM">Tuen Mun</option>',
    'TKL': '<option class="TKL" value="NOP">North Point</option> \
    <option class="TKL" value="QUB">Quarry Bay</option> \
    <option class="TKL" value="YAT">Yau Tong</option> \
    <option class="TKL" value="TIK">Tiu Keng Leng</option> \
    <option class="TKL" value="TKO">Tseung Kwan O</option> \
    <option class="TKL" value="LHP">LOHAS Park</option> \
    <option class="TKL" value="HAH">Hang Hau</option> \
    <option class="TKL" value="POA">Po Lam</option>' };
```

Step 4: Use JavaScript to check whether the user has selected a different MTR line. When detecting there is a change of the selection option, an event handler will be executed to switch the list of stations from the current set to the target set of stations according to the change.

```
    let currentClass="AEL"; //assume Airport Express line initially
    let line = document.getElementById('line');
```

```
    line.addEventListener('change', (evt) => {
      let select = line.value;
      if (select != currentClass) { //there is a change
        let station = document.querySelector('#station');
        station.innerHTML = Stations[select];
        currentClass=select;
      }
    });
```

Based on the key, we access the in-memory database to get the new set of station options and replace the previous set. Be remembered to keep track of the MTR line.

Load the index_WS2.html file to the browser, you should see that the two selection lists working as shown in Fig. 2 and Fig. 3.

Step 5: Work on the AJAX request using fetch().

Register an event handler for the "Get Train Data" button. Upon clicking on the button, a fetch() request will be sent to the server with the correct 'line' and 'sta' parameters. When the response returns, check whether the response status is successful; if it is successful, extract the data from the returned JSON string and output the data to the block with id="output".

```
    const stnList = { "HOK": "Hong Kong", "KOW": "Kowloon", "TSY": "Tsing Yi", "AIR":
    "Airport", "AWE": "AsiaWorld Expo", "HOK": "Hong Kong", "KOW": "Kowloon", "OLY":
    "Olympic", "NAC": "Nam Cheong", "LAK": "Lai King", "TSY": "Tsing Yi", "SUN": "Sunny
    Bay", "TUC": "Tung Chung", "HUH": "Hung Hom", "ETS": "East Tsim Sha Tsui", "AUS":
    "Austin", "NAC": "Nam Cheong", "MEF": "Mei Foo", "TWW": "Tsuen Wan West", "KSR": "Kam
    Sheung Road", "YUL": "Yuen Long", "LOP": "Long Ping", "TIS": "Tin Shui Wai", "SIH":
    "Siu Hong", "TUM": "Tuen Mun", "NOP": "North Point", "QUB": "Quarry Bay", "YAT": "Yau
    Tong", "TIK": "Tiu Keng Leng", "TKO": "Tseung Kwan O", "LHP": "LOHAS Park", "HAH":
    "Hang Hau", "POA": "Po Lam" }

    let bttn = document.getElementById('bttn');
    bttn.addEventListener("click", fRequest); //register the handler fRequest

    function fRequest() {
      let line = document.getElementById('line').value; //get the MTR line
      let station = document.getElementById('station').value; //get the station

fetch(`https://rt.data.gov.hk/v1/transport/mtr/getSchedule.php?line=${line}&sta=${station}`)
      .then( response => {
        if (response.status == 200) {  //receive response successfully
          response.json().then( schedule => {
            let output = "";

            if (schedule.status == 0) {  //Special Train Service
```

```
            //will handle this later

          } else {
            if (schedule.isdelay == 'Y') {  //Data Absence
              //will handle this later

            } else {  //Normal response
              let dataUP = schedule.data[line+'-'+station].UP;
              let dataDN = schedule.data[line+'-'+station].DOWN;

              if (dataUP) {  //has the UP data
                for (let train of dataUP) {
                  let time = train.time.substr(11,5);
                  output += '<span>Time: '+time+'</span>';
                  output += '<span>Platform: '+train.plat+'</span>';
                  output += '<span>Destination: '+stnList[train.dest]+'<br></span>';
                }
                output += '<br>';
              }
              if (dataDN) {  //has the DOWN data
                for (let train of dataDN) {
                  if (Object.keys(train).length) {  //May not have data - Last Train
                    let time = train.time.substr(11,5);
                    output += '<span>Time: '+time+'</span>';
                    output += '<span>Platform: '+train.plat+'</span>';
                    output += '<span>Destination: '+stnList[train.dest]+'<br></span>';
                  }
                }
              }
            }
            //Write the data beneath the button
            document.getElementById('output').innerHTML = output;
          });
        } else {
          console.log("HTTP return status: "+response.status);
        }
      });
    }
```

We have created an associative array (a JavaScript object) stnList to store the mapping between station code to the station name.

Load the index_WS2.html file to the browser and test the program by selecting an MTR line and a station on that line, then click on the “Get Train Data” button to retrieve the data.

Step 6: Complete the program by handling those special cases.

When received a successful response but with the returned JSON status code equals zero, the MTR line is in special arrangement status or your query parameter(s) is/are not correct. Then output the returned message and URL (if available).

```
output += schedule.message;
if (schedule.url)
  output += `<br><a href='${schedule.url}'>${schedule.url}</a>`;
```

When received a successful response but with the returned JSON data ‘isdelay’ field equals “Y”, this indicates the absence of data. Then output the following message.

```
output = "No data is available";
```

Now the program should be able to retrieve the next train arrival timings of the stations along the four MTR lines. Also, it should be able to handle special cases.

Upload and Test

Use any FTP Client to connect to the server i7.cs.hku.hk and create a new directory Workshop2 under public_html. Then upload the index_WS2.html file to the Workshop2 directory.

Browse and test your webpage at:
http://i7.cs.hku.hk/~[your_CSID]/Workshop2/index_WS2.html