Homework: JSON Exercise

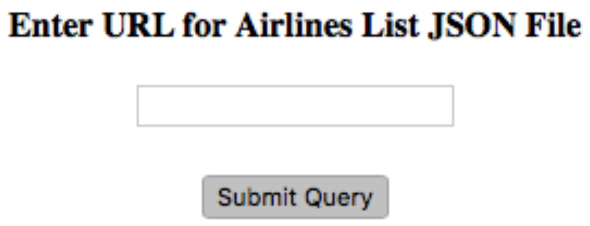
1. Objectives

- Become familiar with the navigating JavaScript JSON objects;
- Use of JSON.parse parser and synchronous XMLHttpRequest;
- Transform the content of a JSON document into an HTML page.

2. Description

You are required to write a HTML/JavaScript program, which takes the URL of a JSON document containing US Airlines information, parses the JSON file, and extracts the list of airlines, displaying them in a table. The JavaScript program will be embedded in an HTML file so that it can be executed within a browser.

- Your program should display a text box to enter the JSON file name as shown below on Figure 1. On clicking the "Submit Query" button, your program should display the table as shown below, Figure 2. If the text box is left empty and Submit Query is clicked, an appropriate error message must be shown.



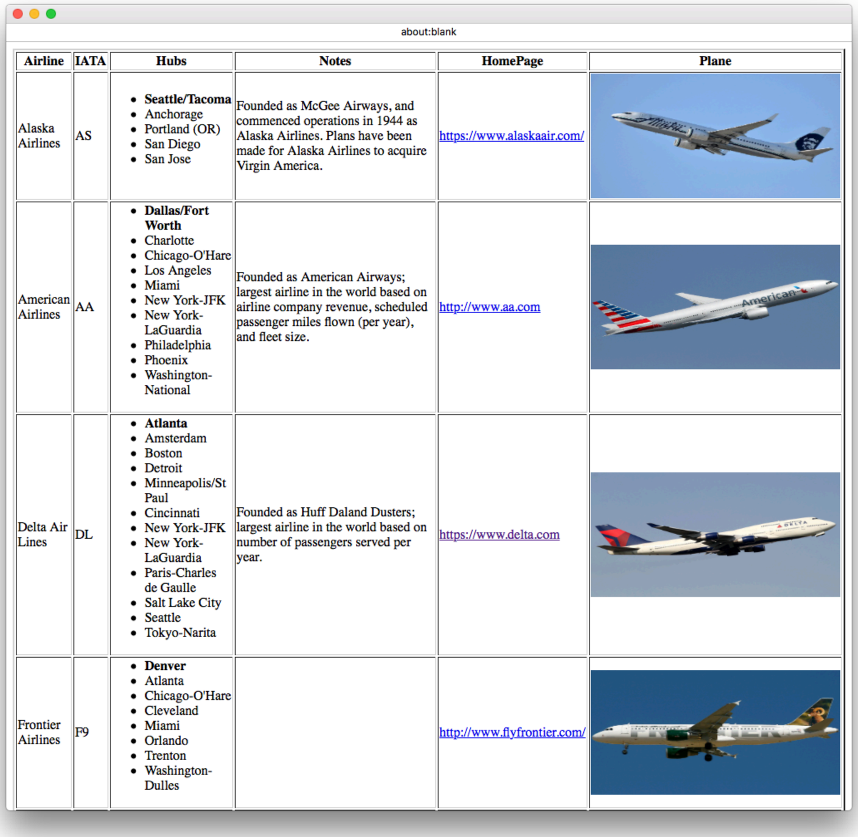
Enter URL for Airlines List JSON File

Figure 1: Initial Page

- Once the JSON file is downloaded, a JavaScript function within the HTML file parses the JSON document that was passed as an input to the popped up window.

- After parsing the JSON document, a table should be displayed consisting of the data for all Airline companies that are contained in the JSON file. For example, given the following XML document:

  http://cs-server.usc.edu:45678/hw/hw4/airlinelist.json the table below should be displayed:



about:blank

Airline	IATA	Hubs	Notes	HomePage	Plane
Alaska Airlines	AS	• **Seattle/Tacoma** • Anchorage • Portland (OR) • San Diego • San Jose	Founded as McGee Airways, and commenced operations in 1944 as Alaska Airlines. Plans have been made for Alaska Airlines to acquire Virgin America.	https://www.alaskaair.com/	
American Airlines	AA	• **Dallas/Fort Worth** • Charlotte • Chicago-O'Hare • Los Angeles • Miami • New York-JFK • New York-LaGuardia • Philadelphia • Phoenix • Washington-National	Founded as American Airways; largest airline in the world based on airline company revenue, scheduled passenger miles flown (per year), and fleet size.	http://www.aa.com	
Delta Air Lines	DL	• **Atlanta** • Amsterdam • Boston • Detroit • Minneapolis/St Paul • Cincinnati • New York-JFK • New York-LaGuardia • Paris-Charles de Gaulle • Salt Lake City • Seattle • Tokyo-Narita	Founded as Huff Daland Dusters; largest airline in the world based on number of passengers served per year.	https://www.delta.com	
Frontier Airlines	F9	• **Denver** • Atlanta • Chicago-O'Hare • Cleveland • Miami • Orlando • Trenton • Washington-Dulles		http://www.flyfrontier.com/	

Figure 2: Table containing airlines from airlinelist.json

Here is a version of the ***airlinelist.json*** file containing the data that is displayed above:

```
{
  "Mainline": {
    "Table": {
      "Header": {
        "Data": [
          "Airline",
          "IATA",
          "Hubs",
          "Notes",
          "HomePage",
          "Plane"
        ]
      },
      "Row": [
        {
          "Airline": "Alaska Airlines",
          "IATA": "AS",
          "Hubs": {
            "Hub": [
              "Seattle/Tacoma",
       "Anchorage",
       "Portland (OR)",
       "San Diego",
       "San Jose"
      ]
     },
     "Notes": "Founded as McGee Airways, and commenced operations in 1944 as
Alaska Airlines. Plans have been made for Alaska Airlines to acquire Virgin America.",
     "HomePage": "https://www.alaskaair.com/",
     "Plane": "http://cs-server.usc.edu:45678/hw/hw4/
Alaska_Airlines,_Boeing_737.jpg"
    },
    {
     "Airline": "American Airlines",
     "IATA": "AA",
     "Hubs": {
      "Hub": [
       "Dallas/Fort Worth",
       "Charlotte",
       "Chicago-O'Hare",
       "Los Angeles",
       "Miami",
       "New York-JFK",
       "New York-LaGuardia",
       "Philadelphia",
       "Phoenix",
       "Washington-National"
      ]
     },
     "Notes": "Founded as American Airways; largest airline in the world based on
airline company revenue, scheduled passenger miles flown (per year), and fleet
size.",
     "HomePage": "http://www.aa.com",
     "Plane": "http://cs-server.usc.edu:45678/hw/hw4/
American_Airlines_Boeing_777.png"
    },
    {
     "Airline": "Delta Air Lines",
     "IATA": "DL",
     "Hubs": {
      "Hub": [
       "Atlanta",
       "Amsterdam",
       "Boston",
       "Detroit",
       "Minneapolis/St Paul",
       "Cincinnati",
       "New York-JFK",
       "New York-LaGuardia",
       "Paris-Charles de Gaulle",
       "Salt Lake City",
       "Seattle",
       "Tokyo-Narita"
      ]
     },
          "Notes": "Founded as Huff Daland Dusters; largest airline in the world based
on number of passengers served per year.",
          "HomePage": "https://www.delta.com",
          "Plane": "http://cs-server.usc.edu:45678/hw/hw4/Delta_B747.jpg"
        },
        {
          "Airline": "Frontier Airlines",
          "IATA": "F9",
          "Hubs": {
            "Hub": [
              "Denver",
              "Atlanta",
              "Chicago-O'Hare",
              "Cleveland",
              "Miami",
              "Orlando",
              "Trenton",
              "Washington-Dulles"
            ]
          },
          "HomePage": "http://www.flyfrontier.com/",
          "Plane": "http://cs-server.usc.edu:45678/hw/hw4/
Airbus_A320-214,_Frontier_Airlines.jpg"
        }
      ]
    }
  }
}
```

Note: The data for the airlines used in the sample files and video has been obtained from Wikipedia:

https://en.wikipedia.org/wiki/List_of_airlines_of_the_United_States

3. Error Handling

An error condition that should be checked for is an JSON file containing NO airline companies. An example of a JSON files which does not contain airline company entries:

```
{
  "Mainline": {
    "Table": {
      "Header": {
        "Data": [
          "Airline",
          "IATA",
          "Hubs",
          "Notes",
          "HomePage",
     "Plane"
    ]
   }
  }
 }
}
```

In addition, you program should handle the case when the JSON file does not exist. A proper message should be displayed.

The “structure” of the input JSON files won’t change. We won’t test the case where the order of “keys” is changed or one of the keys is missed. In other words, every *Row* always contains the keys: *Airline*, *IATA*, *Hubs*, *Notes*, *HomePage*, and *Plane* in the “same” given order. Note that inside the *Hubs* key, there may be ZERO or more *Hub* tags.

If the value of a tag is empty, you should display a blank cell.

You are required to handle the case where the Header data values are different. Please note that the Data tag values might differ but the JSON structure remains the same. For example, the Header can be,

```
</Header>
   "Header": {
    "Data": [
     "US Airline",
     "IATA",
     "Main Hubs",
     "Notes",
     "Home Page",
     "Plane with Logo"
    ]
   }
```

instead of,

```
  </Header>
   "Header": {
    "Data": [
     "Airline",
     "IATA",
     "Hubs",
     "Notes",
     "HomePage",
     "Plane"
    ]
   }
```

No other error conditions need be checked. In all cases if an error is found your program should show an alert box indicating the error was detected.

4. Hints

- *Step 1: Writing Your HTML/JavaScript program - Using the DOM Parser*

  Here's how you could use the Microsoft DOM API and the Mozilla DOM API (used in Firefox) to load and parse an XML document into a DOM tree, and then use the DOM API to extract information from that document.

  ```
  <script type="text/javascript">
  var xmlDoc;
  function loadXML(url) {
      if (window.XMLHttpRequest)
    {
      // code for IE7+, Firefox, Chrome, Opera, Safari
       xmlhttp=new XMLHttpRequest();
    }
   else
    {
    // code for IE6, IE5
     xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
   xmlhttp.open("GET",url,false); //open, send, responseText are
   xmlhttp.send();               //properties of XMLHTTPRequest
   xmlDoc=xmlhttp.responseText;
   return xmlDoc;
   }
  // ....... processing the document goes here
  </script>
  ```

  Now you can parse the JSON file with JSON.parse and generate the HTML table on the fly by navigating through the JSON object. You can assume that every JSON file will have identical Object, Array and key names.

  Your task is to write a program that transforms this type of JSON file into the table as shown above.

- *Step 2: Display the Resulting HTML Document*

You should use the DOM document.write method to output the required HTML.

- *Step 3: Use JavaScript control syntax*

  The only program control statements that you will need to use for this exercise are the “if” and the “for” statements. The syntax of both statements is practically identical to the syntax of the corresponding statement in the C, C++ and Java languages, as in:

```
if (aircraft_keys[j]=="Image") {
      // do stuff
}
for (j=0; j<aircraft_keys.length; j++) {
      // do more stuff
}
```

Please make a note of the following issue:

Cross-Site Scripting (XSS):

JavaScript cannot call the resources from another domain. This is called cross side scripting which is not allowed in browsers. Therefore, you must put your JSON files and your script in the same domain. Please make sure, when you are testing your implementation, to place both the HTML file and the JSON file on your local machine IN THE SAME FOLDER. A sample file can be copied from here:

http://cs-server.usc.edu:45678/hw/hw4/airlinelist.json

Window.open() method must be used to pop up a new window which would display the final widget.

Image files can be either local or remote, as these files do not exhibit the cross-site scripting issue.

USC Disclaimer:

You should avoid displaying the USC disclaimer in your pages. Otherwise, this will result in a 0.5 point penalty.

Scrollable Window:
The popup window should be scrollable so the user can read all records listed in the window.

6. Image Test Files

These image files are provided for testing purpose.

http://cs-server.usc.edu:45678/hw/hw4/spirit_plane.jpg
http://cs-server.usc.edu:45678/hw/hw4/southwest_plane.jpg
http://cs-server.usc.edu:45678/hw/hw4/JetBlue_Airways_Airbus_A321.jpg
http://cs-server.usc.edu:45678/hw/hw4/Airbus_A330-200_Hawaiian.jpg
http://cs-server.usc.edu:45678/hw/hw4/Airbus_A320-214,_Frontier_Airlines.jpg
http://cs-server.usc.edu:45678/hw/hw4/Delta_B747.jpg
http://cs-server.usc.edu:45678/hw/hw4/American_Airlines_Boeing_777.png
http://cs-server.usc.edu:45678/hw/hw4/Alaska_Airlines,_Boeing_737.jpg
http://cs-server.usc.edu:45678/hw/hw4/united_plane.jpg

7. Material You Need to Submit

On your course homework page, your link for this homework should go to a page that looks like the one displayed in the Figure on page 1. **This page should include your entire JavaScript/HTML/CSS program in a single file**. Also, you should submit your source code electronically to the csci571 account so that it can be graded and compared to all other students' code via the MOSS code comparison tool. If your submission is composed of multiple files, 3 points will be deducted.