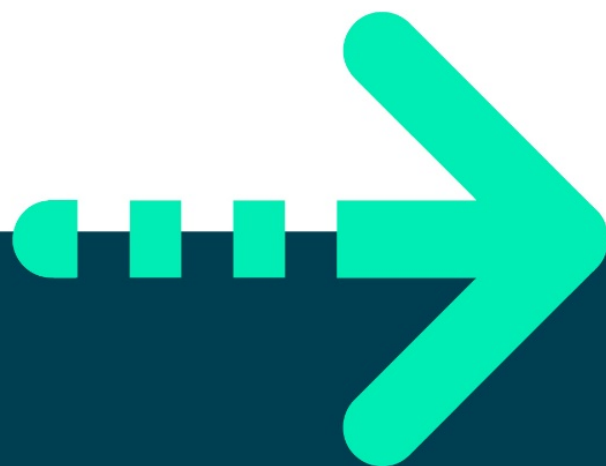




# WEB APPLICATION ARCHITECTURES





# Ajax

## Objectives

In this exercise you will investigate Ajax

## Reference material

This exercise is based on material from the “Ajax” chapter.

## Overview

- In this exercise you will experiment with Ajax using JQuery

## Estimated duration

The estimated duration for this lab is ~45 minutes.

## Completed solution

There is a completed solution for this lab.

## Step by step instructions

### Part-1

#### Investigate Ajax and Web services

In this part you're not expected to write any code.

First of all, you will run a web server on your own computer. To do this:

- Select the start menu, and type “cmd” to open a command window
- Locate the WebService folder within the starters folder for this chapter
- Type “cd” followed by the location of the WebService folder to change into that folder
  - Hint – you can drag the folder into the command window, and the location will be added to the current command
- Type “node index” to run the web server
  - When finished (but not yet!), press Ctrl-C to end the web server

Next, run the client. Load the “client” folder into Code first, open the “index.html” file, and press Alt-B to run it in your web browser. Have a look at what it does.



Spend some time studying the code so that you understand how it works.

Pay particular attention to the use of Ajax in the `getJSON()` method, and note that it automatically de-serializes the JSON data and sends an array of objects into the callback function.

## Part-2

1. Go to <https://api.tfl.gov.uk/Line/Mode/tube/Status> and have a look at the results. Can you recognise that this URL returns some JSON data?
2. Use Google to find a JSON formatter. Copy and paste the contents of the previous web request into a JSON formatter, and then examine the results. What do you think this JSON data is telling you?
3. Create a web page which uses JQuery and AJAX to show the most relevant parts of the JSON data to the user:
  - a. You have seen examples of a button which fires an AJAX request. But in this case, it would be better if the AJAX request was fired as soon as the web page loads – the user shouldn't have to take an extra step to initiate the action.

Therefore, inside the `$(document).ready()` function, you will need to add a call to `$.getJSON()`

- b. Notice that the JSON contains an array. Therefore, you will need some kind of loop to go through each item in the array. An alternative would be to use `$.each()` to go through each item in the array.
  - c. For each item in the array, we are interested in its "name".
  - d. Additionally, each item in the array contains a sub-array called "lineStatuses".

We only need the first item from this sub-array – if there is more than one item in the array, you can just ignore everything except the first one.

From this first item, retrieve and display the "statusSeverityDescription".

- e. Ensure the data is displayed to the user in an appropriate format, so that the user can read it clearly!

