# Lesson 07 – Introduction to AJAX with JQuery

# Session 2 Worksheet

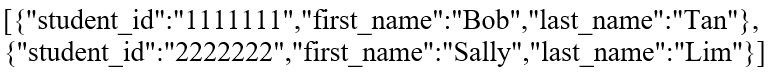
**Exercise 3a:**

Create a PHP file called 3a\_getStudents.php

* Retrieve all data from the database table student
* Test it by executing the following:

<http://localhost/C273_L07Ajax/3a_getStudents.php>

You should get the following JSON response:



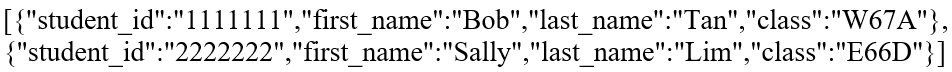
**Exercise 3b**

Create a PHP file called 3b\_getStudentsByModule.php

* Retrieve all data from the database table student based on the module (hint: You need to write a SQL join between tables student and allocation.)
* Test it by executing the following:

<http://localhost/C273_L07Ajax/3b_getStudentsByModule.php?module_code=c207>

You should get the following JSON response:



**Exercise 3c**

Create a PHP file called 3c\_getStudentDetails.php

* Retrieve all data from the database table student based on the student id
* Test it by executing the following:

[http://localhost/C273\_L07Ajax/3c\_getStudentDetails.php?student\_id=1111111](http://localhost/C273_L07Ajax/3c_getStudentDetails.php?student_id=1111111%20)

You should get the following JSON response:



## AJAX with jQuery

**Resources:** <http://api.jquery.com/jquery.ajax/>

**The $.ajax(*settings*) or $.ajax(*url*, *settings*)**

Performs an asynchronous HTTP (Ajax) request. The *settings* is an object of key-value pairs:

* *url*: The request URL
* *type*: GET or POST. The default is GET
* *data*: Request parameters (name=value pairs). Can be expressed as an object (e.g., {name:"peter", msg:"hello"}), or query string (e.g., "name=peter&msg=hello").
* *dataType*: Expected response data type, such as text, xml, json, script or html.
* *cache*: If set to false, it will force requested pages not to be cached by the browser.
* *headers*: an object for request header key-value pairs
* *success*: A function to be called if the request succeeds. The data returned from the server, formatted according to the dataType parameter. If the datatype is JSON, a JavaScript object will be returned.
* *error*: A function to be called if the request fails. The function receives three arguments: The jqXHR (in jQuery 1.4.x, XMLHttpRequest) object, a string describing the type of error ("timeout", "error", "abort", and "parsererror") that occurred and an optional exception object, if one occurred. When an HTTP error occurs, errorThrown receives the textual portion of the HTTP status, such as "Not Found" or "Internal Server Error."

**Exercise 4a**

Open 4a\_showFruit.html and see how the jQuery code makes an ajax call to 1a\_getFruit.php to retrieve the fruit details and replaces the content in the <div>.



Line 6 – Link current document to the jQuery library

Line 8 – All jQuery code is inside a document ready event. This is to prevent any jQuery code from running before the document is finished loading (is ready).

Line 15 – In this case the JSON response is {"fruit\_name":"watermelon",….}, thus in order to retrieve the value "watermelon", you need to access **response.fruit\_name**. The value will replace the contents in the element of id "fruitName" using the html() method.

Line 20 – The error message is preferably displayed in the console for debugging purpose. Otherwise it can be displayed to the user using the alert function.

Run 4a\_showFruit.html and you should see the fruit details displayed on the browser.

**Exercise 4b**

Open 4b\_showColour.html and write the jQuery code to make an ajax call to 1b\_getColour.php and display the random colour in the “container”.

**Exercise 4c**

Open 4c\_circle\_calculator.html and write the jQuery code to

* Make an ajax call to 1c\_getCircleResults.php passing the “radius” and “type” as parameters (hint: use the “data” attribute in $.ajax.   
  Eg, to pass radius=3 and type=area, we write **data: "radius=3&type=area"**.  
  But radius and type depend on the choice of the user, how do we get the values and replace the data attributes?)
* Display the results from the JSON response in #results.

