

CSE 3101: Internet Computing II

Lab Session 5

Paul Crawford

Semester 1, Week 10 (29th & 30th October, 2018)

1. Aims

- 1.1. Further understanding of Web applications, including client-side Web Forms (with server-side processing upon submission), server-side PHP functions, and Web Service architectures.
- 1.2. Further increased facility with the ongoing sets of concepts & techniques needed for eventual completion of the Final Project.
- 1.3. **POLICY CHANGE**: Please note that henceforth **all Lab Sessions will be marked** and count as a portion of the overall Lab grade (10%). Completed labwork must be zipped into an archive (filename given later below at the end of § 2.1), and submitted to the Tutor (<pcrawford@mac.com>) by midnight Friday each week, with the Subject line: 'CSE3101 - Lab <N> - <YourGroupName>'.

2. Tasks

2.1. Web API (RESTful Flavour)

- 2.1.1. Download the RESTful API folder archive ([kindly provided by the 2nd Lecturer](#)) from the tutor's Nyanza Software website (<www.nyanzasoftware.com>) > 'Teaching', § 'University of Guyana' > 'CSE3101: Internet Computing II'. Then, **unzip (extract)** it to obtain the 'RESTful API' folder.
- 2.1.2. Set up the test database via the PhpMyAdmin console, by first creating an empty database named 'api_db' ([also beforehand temporarily renaming any existing one from class](#)), and then **importing** the 'api_db.sql' file (located in the folder extracted above).
- 2.1.3. Copy the 'restapi' subfolder (located in the folder extracted above) into your Apache Web Root directory (e.g., 'C:\xampp\htdocs'). Note: Depending on your xAMP[P] configuration, you might then need to edit the 'config/database.php' file to set an empty password (for the 'root' user) near the top.
- 2.1.4. Exercise the Web API functionality (the 'R' or query URIs can be tested via any Web browser, and the 'CRUD' URIs can be tested via either the Tutor's extra web form or a specialised web service such as Postman), and save the various results as follows:

- 2.1.4.1. Navigate to 'localhost/restapi/**category/read.php**', and save the resulting JSON data (there should be 6 entries) to a file named 'Category_Read.json'.
- 2.1.4.2. Navigate to 'localhost/restapi/**product/read.php**', and save the resulting JSON data (there should be 17 entries) to a file named 'Product_Read_Original.json'.
- 2.1.4.3. Navigate to 'localhost/restapi/**product/read_one.php?id=1**', and save the resulting JSON data (there should be just 1 entry) to a file named 'Product_Read_One.json'.
- 2.1.4.4. Navigate to 'localhost/restapi/**product/read_paging.php?page=2**', and save the resulting JSON data (there should be 5 entries) to a file named 'Product_Read_Paging.json'.
- 2.1.4.5. Navigate to 'localhost/restapi/**product/search.php?s=Wall**', and save the resulting JSON data (there should be 2 entries) to a file named 'Product_Search.json'.
- 2.1.4.6. Navigate to 'localhost/restapi/**paulc_extras/cud_client.html**' OR use a web service such as Postman, and create a new product by submitting to the URI 'localhost/restapi/**product/create.php**' the following POST data in JSON format:

For the Client Form (which will auto-convert into JSON):

```
name       : My Test Car
description : Great car designed by me!
price      : 1000
category_id : 3
```

For Postman & similar services (making sure to select the JSON data-type):

```
{
  "name"       : "My Test Car",
  "description" : "Great car designed by me!",
  "price"      : 1000,
  "category_id" : 3
}
```

Make sure that you see the result: 'message: Product was created.'. **{Note: For the Client Form, it will be logged to the Console.}**

- 2.1.4.7. Navigate to 'localhost/restapi/**paulc_extras/cud_client.html**' OR use a web service such as Postman, and update the existing 'Wallet' product by submitting to the URI 'localhost/restapi/**product/update.php**' the following POST data in JSON format:

For the Client Form (which will auto-convert into JSON):

```
id       : 28
name     : Wallet (Improved)
description : You can absolutely use this one!
```

```
price      : 799
category_id : 6
```

For Postman & similar services (making sure to select the JSON data-type):

```
{
  "id"          : 28,
  "name"        : "Wallet (Improved)",
  "description" : "You can absolutely use this one!",
  "price"       : 799,
  "category_id" : 6
}
```

Make sure that you see the result: 'message: Product was updated.'. {Note: For the Client Form, it will be logged to the Console.}

- 2.1.4.8. Navigate to 'localhost/restapi/**paulc_extras/cud_client.html**' OR use a web service such as Postman, and delete the existing 'Amanda Waller Shirt' product by submitting to the URI 'localhost/restapi/**product/delete.php**' the following POST data in JSON format:

For the Client Form (which will auto-convert into JSON):

```
id      : 31
```

For Postman & similar services (making sure to select the JSON data-type):

```
{
  "id" : 31
}
```

Make sure that you see the result: 'message: Product was deleted.'. {Note: For the Client Form, it will be logged to the Console.}

- 2.1.4.9. (Re-)Navigate to 'localhost/restapi/**product/read.php**', and save the resulting JSON data (there should still be 17 entries) to a file named 'Product_Read_AfterCUD.json'.
- 2.1.4.10. Create a zip archive named 'CSE3101_Lab05_<YourGroupName>.zip' from the above **6** result files — i.e.: 'Category_Read.json', 'Product_Read_Original.json', 'Product_Read_One.json', 'Product_Read_Paging.json', 'Product_Search.json' and 'Product_Read_AfterCUD.json' — and submit it as described in § 1.3.

2.2. Overall Considerations

- 2.2.1. For additional information, practice and code samples, you could also explore any of various useful articles, tutorials & references available online. For instance, the **Wikipedia** site (<en.wikipedia.org>) has several detailed articles on Web Services, Web APIs, the REST architecture, etc.