

COMP 4669

Advanced Web Applications with PHP

Assignment 3

Assignment 3

This assignment will test everything covered in lessons 7 – 10, assumes knowledge of lesson 1 – 6, including

- Public API from a service provider
- PHP Third Party SDKs
- MVC pattern
 - o JSON Middleware Presentation
 - o Separation of concerns
 - o Model Hydration
 - o Database Interaction – table joins, pagination and query building

Import pictures from Instagram

This assignment will extend from your Assignment 2. Recall that your Assignment 2 required you to pull data from a database, hydrate your Models, and present the results in a JSON format for the front end SPA. The data was already seeded for you by your starter kit. In this assignment, you will be importing pictures into the database tables from Instagram.

Getting Started

- Use the solution (or your own solution) for Assignment 2 as the starting point.
- Use the slightly modified SPA (`bundle.js`).
- Run the SQL alter statement from `sql.txt` to alter your `pictures` table.

Here are the details

- When a user goes to `/authorize`, they should be redirected to Instagram to ask for authorization.
- When the user approves, they are to be redirected to `/import`. The Controller-Action for `/import` will
 - request for an Access Token from Instagram
 - use the access token to request for media from the current user
 - parse the response and insert the picture URL into the `pictures` tables, in the column `pictures_instagram`
 - any of the 3 URLs is fine – Instagram provides one for each resolution
 - insert the comments for each imported picture, into the `comments` table
 - A simple “done” message for the View is fine.
- An addition to your JSON payloads will now include the key `pictureInstagram`, and the value assigned as the field from the `pictures_instagram` column.
- Do not worry about authors but you're welcome to attempt this.
- Life will be much easier if you use the `php-instagram-api` SDK

Submission

Your instructor will modify the client ID, secret, scope and redirect URI. You just have to worry about getting this working with your developer account.

Don't worry, if you've been paying attention you'll understand that your instructor has no way of accessing your private Instagram account.

There is no need to submit your `vendor/` directory or the `composer.phar` file. You **must** submit your `composer.json` and `composer.lock` files, and the rest of your source code.

Compress and submit your files to the D2L dropbox (<http://learn.bcit.ca>)