

Project Milestone #3: API Design and Implementation

Re-submit Assignment

Due Mar 27 by 11:59pm **Points** 40 **Submitting** a file upload

Some code to help you get started with milestone 3: <https://github.com/jerodw/tweeter-samples/tree/server-sample> [_](https://github.com/jerodw/tweeter-samples/tree/server-sample)(<https://github.com/jerodw/tweeter-samples/tree/server-sample>). Here is an explanation of the sample code: https://byu.instructure.com/courses/5934/pages/milestone-3-sample-code?module_item_id=438918

Note 1: This is a different repository than was used for milestone 2 sample code, and that it includes some changes to the client code.

Note 2: Your backend is to be implemented in **Java**, unless you have permission (from your instructor) to use something else.

Overview

In this Milestone, do the following:

1. **Design and implement the Web API your Twitter client will use to call your Twitter backend.**
Implement the endpoints in AWS API Gateway and create a deployment that your client can access.
2. **Connect each of your endpoints to an AWS Lambda.** Each of these endpoints will return dummy data, rather than connect to a database, but are otherwise complete. (Connecting to the database will wait until milestone 4).
3. **Modify your Android client to call each of your Web API endpoints**, rather than use dummy data in the client as you did for milestone 2. Implement pagination in all user interface views (and the associated Web API endpoints) that require it.

Design and implementation

- The client code should use *AsyncTask* and service classes to handle calls to the backend services.
- Your backend (which is deployed to AWS Lambda) should have the following layers:
 - The Lambda handler
 - A services layer to which the handler delegates each request
 - A DAO layer with the dummy data (and which will interact with the database in milestone 4)
- Your implementation is to meet the "user and session management" requirements in the [project overview](#), but for milestone 3 you can use hard coded credentials and a hard coded authentication token.

In Fall 2019, the TAs graded this milestone and gave the following feedback about students' API designs: [M3 API Endpoints Design Suggestions](#)

Testing

Include (at a minimum) tests for each of your Lambda handler functions (which will result in your services being tested as well).

Web API Endpoint Specifications

Your Web API will need to provide capabilities such as:

1. Sign up a user
2. Sign in a user
3. Follow a user
4. Determine if a user is currently following another user
5. Get a user's followers (must be paginated)
6. Send a status
7. Etc

This list includes about half of the necessary endpoints. You will need to figure out the other half on your own. When defining your API in the API Gateway please do the following:

- Provide a description for each API **resource** and **method** that you define. (To do this, click on the "gray book" icon.)
- For each method that has a request body, specify a request body model.
- For each method, ensure you have integration responses for all relevant HTTP status codes (eg, 200, 400, 500).

Submission

- Pass off your project with a TA by the due date at 5pm
- Submit to Canvas a zip file containing your project in its current form. *If your project is too big to upload to Canvas, you are probably including files that shouldn't be there. Only include files that you would version.*
- Submit to Canvas an exported YAML or JSON swagger file that describes your API. *To do this export, click on your stage, then the export tab. You will see the option to "export as swagger".*
- Submit to Canvas a PDF file containing a UML sequence diagram demonstrating what happens when a user sends a status, including both client and server side objects.

Rubric

- [30 points] The project overview defines 10 features you are to implement. Each feature will be graded out of 3 for a total of 30 feature points, as follows:

- 1 point for correct functionality (with your client now calling the backend) including paginating as needed
- 1 point for implementing the backend functionality with three separate layers (as described above)
- 1 point for unit tests for handler classes
- [5 points] For your JSON/YAML swagger file describing your API. For full points your API must meet the requirements described above in the "Web API Endpoint Specifications" section.
- [5 points] For your UML sequence diagram

Milestone #3 Rubric (Just for TA convenience!)

Criteria	Ratings		Pts
Sign Up Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Sign In Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Sign Out Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Post Status Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
View Feed Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
View User Story Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
View User Followers Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
View User Following Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts

Criteria	Ratings		Pts
Follow Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Unfollow Feature - Is functional - Layered Architecture - Has Unit Tests	3.0 pts Full Marks	0.0 pts No Marks	3.0 pts
Swagger JSON/YAML Export describing your API For full points your API must meet the requirements described above in the "Web API Endpoint Specifications" section.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
UML Sequence Diagram for sending a status UML sequence diagram demonstrating what happens when a user sends a status, including both client and server side objects.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Late Points -10% per day	0.0 pts Full Marks	0.0 pts No Marks	0.0 pts
Total Points: 40.0			