

Introduction

You'll work in pairs (i.e. group of **two**) to complete this project assignment. Pay attention to the submission deadline, deliverables and how your project will be assessed.

Submission Deadline

December 4 2017 (Monday) 09:00 (T01/T02), 11:00 (T03)

Project Deliverables

1. An **Express server** that provides basic **CRUD** services for the restaurants collection. The server **must** be running on **IBM Cloud (Bluemix) or Heroku**.
2. Full source code uploaded to **GitHub.com**
3. **Demo** on **December 4, 2017**

Functional Requirements

1. Create user accounts
 - Each user account has a `userid` and `password`.
 - Upon successful login, `userid` must be stored in **cookie session**
2. Create new restaurant documents
 - Restaurant documents may contain the following attributes:
 - i. `restaurant_id`
 - ii. `name`
 - iii. `borough`
 - iv. `cuisine`
 - v. `photo`
 - vi. `photo mimetype`
 - vii. `address`
 1. `street`
 2. `building`
 3. `zipcode`
 4. `coord`
 - viii. `grades`
 1. `user`
 2. `score`
 - ix. `owner`
 - `name` and `owner` are mandatory; other attributes are optional
3. Update restaurant documents
 - A document can **only** be updated by its owner (i.e. the user who created the document)
4. Rate restaurant. A restaurant can only be rated **once** by a user.
 - `score > 0` and `score <= 10`
5. Display restaurant documents
 - Show `photo` *if* it's available
 - Show a link to Google Map *if* `coord` is available
6. Delete restaurant documents
 - A document can **only** be deleted by its owner
7. Search / Filtering
 - by `name`, `borough`, `cuisine`, `borough`, etc.

8. Provide the following RESTful services.

Request	Parameters	Response	Remarks
POST /api/restaurant/create	<pre>{ name: 'xxx', address: { street: 'yyy', ... } ... }</pre>	<pre>{ status: ok, _id: zzz }</pre> <p>OR</p> <pre>{ status: failed }</pre>	<p>Create new restaurant documents.</p> <p>_id refers to the ObjectID</p>
GET /api/restaurant/read	<pre>/name/xxx /borough/yyy /cuisine/zzz</pre>	<pre>[{ restaurant doc₁ }, { restaurant doc₂ }, ...]</pre> <p>OR</p> <pre>{}</pre>	<p>Read restaurant documents with criteria:</p> <p>xxx - name of restaurant</p> <p>yyy - borough of restaurant</p> <p>zzz - cuisine served</p>

Non-Functional Requirements

1. Use EJS to render all UI
2. Use MongoDB driver (Mongoose is slow!)
3. Use npm package `express-fileupload` to upload photos

A Demo Server

A sample server can be found at <http://projdemo.mybluemix.net>

- Login as user demo (no password)
- **This server does not contain all project features!!!**
- **This server is not bug-free!!!**

Assessment Criteria

1. Marks will be deducted (up to 50%) for any of the conditions below:
 - a. Your server is not running in the 'cloud'
 - b. Incomplete source code on GitHub
2. Your server is required to pass **five** test cases. Here's a *sample* test case.

Precondition:

- An empty restaurant collection

Steps:

- Login as *demo*
- Create a restaurant document with name = 'Starbucks '
- Submit GET /api/restaurant/read/name/Starbucks

Expected outcome:

- A JSON string containing the Starbucks document.