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 5 2015 (Monday) 09:00 (T01), 11:00 (T02/03)

Project Deliverables

Write an Express server that provides basic CRUD services for the `restaurants` collection. You **must** deploy your server to IBM Bluemix for assessment.

What and How to submit?

1. Source code uploaded to your personal GitHub account.
2. Server running in IBM Bluemix (for demo)
3. 10-minutes demo in the last lab session (December 5)

Functional Requirements

1. Create user accounts. Each user account has a `userid` and `password`.
   - Put userid in *cookie session*.
2. Create new restaurant documents.
   - Restaurant `name` is mandatory.
   - Restaurant document *may* contain a photo
3. Display document details using EJS.
   - Display photo if it's available.
   - Display a map showing the location of the restaurant if `coord` is available.
4. Allow document *owner* (the user who created the document) to update restaurant document.
5. Allow document owner to delete restaurant document.
6. Rate restaurant. Each restaurant can only be rated by a user once.
   - 0 > `score` >= 10
7. Basic search functions (by *name*, *borough*, *cuisine*)
8. Provide the following RESTful services for the `restaurant` **resource**.

Request	Request Parameters	Response	Remarks

`POST /api/create`	`{`
	`  name: 'xxx',`
	`  address: {`
	`  street: 'yyy',`
	`  …`
	`  }`
	`}`
		`{`
		`  status: ok,`
		`  _id: zzz`
		`}`
		
		`OR`
		
		`{`
		`  status: failed`
		`}`
			Create new restaurant documents.
			
			_id is the objectId

`GET /api/read`	`/name/xxx`
	`/borough/yyy`
	`/cuisine/zzz`
		`[`
		`{ restaurant doc1},`
		`{ restaurant doc2},`
		`…`
		`]`
		
		`OR`
		
		`{}`
			Read restaurant documents with criteria:
			`xxx` - name of restaurant
			`yyy` - borough of restaurant
			`zzz` - cuisine served

Your Express server should use a cloud-based MongoDB server (such as mlab), the nodejs monogdb driver and EJS.

Below is a sample server that implements *some* of the functional requirements. It's not bug-free!

`http://projdemo.mybluemix.net`

- Login as *demo* (no password).

How you'll be assessed?

You 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/read/name/Starbucks`

Expected outcome:

- A JSON string containing the `Starbucks` document.