**Assignment 2**

Start with an RESTful Express API server by creating a new project and installing Express middleware which you think you'll need (the minimum is body-parser). Then, implement an Express application which has the following REST API endpoints:

* GET and POST /posts
* PUT and DELETE /posts/:postId/
* GET and POST /posts/:postId/comments
* PUT and DELETE /posts/:postId/comments/commentId

Some hints and constraints for the lab:

* Create a folder named routes.
* Implement all /posts routes in routes/posts.js and all /posts/:postId/comments routes in routes/comments.js.
* Use module.exports to export the functions created for posts and comments
* Use require to import posts and comments functions
* Create routes/index.js which imports and exports posts.js and comments.js so they can be used with require('routes') in server.js
* The removal of a blog post must remove all its comments.
* Use an in-memory store with the structure similar to this:

let store = {

posts: [

{name: 'Top 10 ES6 Features every Web Developer must know',

url: 'https://webapplog.com/es6',

text: 'This essay will give you a quick introduction to ES6. If you don’t know what is ES6, it’s a new JavaScript implementation.',

comments: [

text: 'Cruel…..var { house, mouse} = No type optimization at all',

text: 'I think you’re undervaluing the benefit of ‘let’ and ‘const’.',

text: '(p1,p2)=>{ … } ,i understand this ,thank you !'

]

}

]

}

In the end, your server.js file will be small and only use require to plug routes for posts and comments. Here's the boilerplate for the comments.js. You need to fill in the blank and implement these four methods:

module.exports = {

getComments(req, res) {

},

addComment(req, res) {

},

updateComment(req, res) {

},

removeComment(req, res) {

}

}

The posts.js file looks similar:

module.exports = {

getPosts(req, res) {

},

addPost(req, res) {

},

updatePost(req, res) {

},

removePost(req, res) {

}

}

Test your server with Postman, or any other HTTP agent or tester. For example, the CRUD commands are

**//posts post data : POST**

"Content-Type: application/json"

'{"name": "Top 10 ES6 Features", "url":"http://webapplog.com/es6", "text": ""}'

"http://localhost:3000/posts"

**//updates post data at specific id : PUT**

'Content-Type: application/json'

'{"name": "Top 10 ES6 Features Every Developer Must Know", "url":"http://webapplog.com/es6", "text": ""}'

"http://localhost:3000/posts/0"

**//gets post data using GET**

"http://localhost:3000/posts"

//deletes post data at specific id using DELETE

"http://localhost:3000/posts/0"