SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 3

Practical 5 – expressJS and Mongoose

SE3040 – Applications & Frameworks **Semester 1, 2019**

Objective: Teach main features of expressJS and MongooseJS

1. Initialize a nodeJS project directory.

2. Install expressJS module to the nodeJS project.
   npm install express --save

3. Serve HTML page via expressJS.

   Hint: https://expressjs.com/en/starter/static-files.html

```
'use strict';

const express = require('express');

const app = express();

app.use(express.static(__dirname));

app.get('/', (req, res, next) => {
   res.sendFile('index.html');
});

app.listen(3000, err => {
   if (err) {
       console.error(err);
       return;
   }
   console.log('app listening on port 3000');
});
```

4. Create a POST REST route to save a user object in memory array.

   Hint: https://expressjs.com/en/starter/basic-routing.html

   a. User object should contain firstName, secondName and birthday attributes.
   b. Unique id should be generated at the server and should be added into the user object before saving it.
   c. Date.now() method returns the milliseconds elapsed since 1970-01-01 00:00:00. Use this value as the ID.
   d. Save birthday as a date not as a String (Use Date() constructor).
   e. user Array.push() method to add items to users array.

f. In order to accept JSON in request body user the module body parser and add bodyParser.json() method as an express middleware.

```
'use strict';

const express = require('express');

const app = express();

app.use(express.json());

app.use(express.static(__dirname));

app.get('/', (req, res, next) => {
    res.sendFile('index.html');
});

const users = [];

app.post('/users', (req, res, next) => {
    const user = req.body;
    user.birthday = new Date(user.birthday);
    user.id = Date.now();
    users.push(user);
    res.json(user);
});

app.listen(3000, err => {
    if (err) {
        console.error(err);
        return;
    }
    console.log('app listening on port 3000');
});
```

5. Add another 3 routes to;

   Hint: https://expressjs.com/en/starter/basic-routing.html

   a. Get all users.
   b. Get a user by ID (Hint: Use Array.find or Array.findIndex method).
   c. Update a user (Hint: Use Array.find or Array.findIndex method).
   d. Delete a user by ID (Hint: Use Array.find or Array.findIndex method and Array.splice method).

```
'use strict';

const express = require('express');

const app = express();

app.use(express.json());

app.use(express.static(__dirname));

app.get('/', (req, res, next) => {
   res.sendFile('index.html');
});

const users = [];

app.get('/users', (req, res) => {
   res.json(users);
});

app.get('/users/:id', (req, res) => {
   const user = users.find(user => user.id ===
parseInt(req.params.id));
   res.json(user);
});

app.post('/users', (req, res) => {
   const user = req.body;
   user.birthday = new Date(user.birthday);
   user.id = Date.now();
   users.push(user);
   res.json(user);
});

app.put('/users/:id', (req, res) => {
   const user = req.body;
   user.birthday = new Date(user.birthday);
   delete user.id;
   const index = users.findIndex(user => user.id ===
parseInt(req.params.id));
   users[index] = user;
   res.json(users[index]);
});

app.delete('/users/:id', (req, res) => {
   const user = req.body;
   const index = users.findIndex(user => user.id ===
parseInt(req.params.id));
   users.splice(index, 1);
   res.sendStatus(200);
});

app.listen(3000, err => {
   if (err) {
```

```
            console.error(err);
            return;
        }
        console.log('app listening on port 3000');
    });
```

6. Use MongoDB to save users

   Hint: https://expressjs.com/en/starter/basic-routing.html
   Hint: http://mongoosejs.com/docs/

   a. Install mongoose dependency.
      npm install --save mongoose.
   b. Connect to the mongoDB (make sure local mongo instance is up and running).
   c. Create a model for user.

```
'use strict';

const mongoose = require('mongoose');

const UserSchema = new mongoose.Schema({
    firstName: String,
    lastName: String,
    birthday: Date
});

const User = mongoose.model('User', UserSchema);

module.exports = User;
```

   d. MongoDB has auto generated _id field use that instead of previously created id.

7. Change all operations (get all, get one, update and delete) to use mongoDB.

   Hint: https://expressjs.com/en/starter/basic-routing.html
   Hint: http://mongoosejs.com/docs/

```
'use strict';

const express = require('express'),
   mongoose = require('mongoose');

mongoose.Promise = global.Promise;

const app = express();

app.use(express.json());


mongoose.connect('mongodb://localhost:27017/expressjsSample', err
=> {
   if (err) {
       console.log(err);
       process.exit(1);
   }
});

const UserModel = require('./user.model');

app.use(express.static(__dirname));

app.get('/', (req, res, next) => {
   res.sendFile('index.html');
});

app.get('/users', (req, res) => {
   UserModel.find().exec().then(users => {
       res.json(users);
   }).catch(err => {
       console.error(err);
       res.sendStatus(500);
   });
});

app.get('/users/:id', (req, res) => {
   UserModel.findById(req.params.id).exec().then(user => {
       res.json(user || {});
   }).catch(err => {
       console.error(err);
       res.sendStatus(500);
   });
});

app.post('/users', (req, res) => {
```

```
    const user = new UserModel(req.body);
    user.save().then(user => {
        res.json(user);
    }).catch(err => {
        console.error(err);
        res.sendStatus(500);
    });
});

app.put('/users/:id', (req, res) => {
    const user = new UserModel(req.body);
    user.update().then(() => {
        res.sendStatus(200);
    }).catch(err => {
        console.error(err);
        res.sendStatus(500);
    });
});


app.delete('/users/:id', (req, res) => {
    UserModel.remove(req.params.id).then(() => {
        res.sendStatus(200);
    }).catch(err => {
        console.error(err);
        res.sendStatus(500);
    });
});

app.listen(3000, err => {
    if (err) {
        console.error(err);
        return;
    }
    console.log('app listening on port 3000');
});
```