Міністерство освіти і науки України  
Івано-Франківський національний технічний університет нафти і газу

Кафедра програмного   
забезпечення автоматизованих систем

Лабораторна робота №5

«Розроблення RESTful API для сайту»

Виконав: ст. гр. ПІ-13-1

Бабала Василь

Перевірив:   
Броновський І. В.

м. Івано-Франківськ

2016

***Завдання:*** Розробити API для сайту використовуючи методи POST, GET, PUT, PATCH, DELETE Додати можливість додавання товарів (книжок, людей, згідно вашої теми). Додати можливість додавання, видалення, редагування. Розробити сторінку з документацією доступних методів і їх приклади використання.

***Виконання***

Для реалізації необхідного *api*, потрібно описати моделі даних (GitHub: /models), а також прописати методи для зчитування, запису і оновлення даних в моделях. Робота з простими та більш складнішими структурами даних була організована за допомогою id певного Json-файла. *Приклад:*

//Отримання певної книги за id

module.exports.getBookById = function (id, callback, limit) {

Book.findById(id, callback);

};

//Оновлення книги

module.exports.updateBook = function (id, book, options, callback) {

var query = {\_id: id};

var update = {

title: book.title,

genre: book.genre,

description: book.description,

author: book.author,

publisher: book.publisher,

pages: book.pages,

image\_url: book.image\_url,

buy\_url: book.buy\_url

}

Book.findOneAndUpdate(query, update, options, callback);

};

Також для обробки даних потрібно описати необхідні для робити api в «головному» серверному файлі – *app.js*, де також відловлюються базові помилки:

*аpi-функції:*

app.get('/', function (req, res) {

res.send('Please use /api/books or /api/genres');

});

app.get('/api/genres', function (req, res) {

Genre.getGenres(function (err, genres) {

if(err){

throw err;

}

res.json(genres);

});

});

app.post('/api/genres', function (req, res) {

var genre = req.body;

Genre.addGenre(genre, function (err, genre) {

if(err){

throw err;

}

res.json(genre);

});

});

app.put('/api/genres/:\_id', function (req, res) {

var id = req.params.\_id;

var genre = req.body;

Genre.updateGenre(id, genre, {}, function (err, genre) {

if(err){

throw err;

}

res.json(genre);

});

});

app.post('/api/books', function (req, res) {

var book = req.body;

Book.addBook(book, function (err, book) {

if(err){

throw err;

}

res.json(book);

});

});

app.get('/api/books', function (req, res) {

Book.getBooks(function (err, books) {

if(err){

throw err;

}

res.json(books);

});

});

app.get('/api/books/:\_id', function (req, res) {

Book.getBookById(req.params.\_id, function (err, book) {

if(err){

throw err;

}

res.json(book);

});

});

app.put('/api/books/:\_id', function (req, res) {

var id = req.params.\_id;

var book = req.body;

Book.updateBook(id, book, {}, function (err, book) {

if(err){

throw err;

}

res.json(book);

});

});

app.delete('/api/genres/:\_id', function (req, res) {

var id = req.params.\_id;

Genre.removeGenre(id, function (err, genre) {

if(err){

throw err;

}

res.json(genre);

});

});

app.delete('/api/books/:\_id', function (req, res) {

var id = req.params.\_id;

Book.removeBook(id, function (err, book) {

if(err){

throw err;

}

res.json(book);

});

});

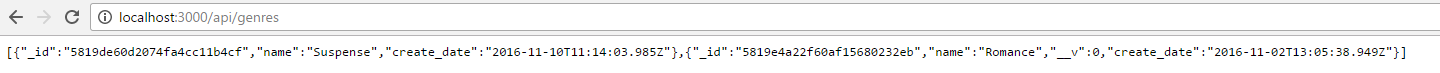
***Приклад роботи:***

Для того, щоб отримати доступ до Json-колекцій, необхідно після запуску рішення (*node app*), в браузері прописати «шлях» до api:

* /api/books – доступ до всіх книг
* /api/genres – доступ до наявних жанрів

Доступ до api з користувачами заблокований в mongoose в цілях безпеки.

Також було протестовано оновлення інформації за допомогою програмного додатка *Postman:*

******

******

***Висновок:*** на цій лабораторній роботі було розроблено RESTful Api для сайту, дані передаються у JSON.