rr

**МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп’ютерних систем**

**Лабораторна робота №8**

**з дисципліни “ Основи web-програмування”**

**тема “Створення REST API сервера”**

|  |  |  |
| --- | --- | --- |
| **Виконав**  **студент II курсу**  **групи КП-81**  **Римаренко Ілля Русланович** |  | **Перевірив**  **“\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.**  **викладач**  **Гадиняк Руслан Анатолійович** |

**Мета роботи**

Реалізувати JSON REST API для доступу до ресурсів веб-сервера.

**Постановка завдання**

У серверній частині додати JSON REST API для виконання CRUD операцій над ресурсами сервера (Users, {Entities}, {Composites}).

1. Отримувати об'єкти можна одиночно по ідентифікатору, або декілька списком. Отримання об'єктів списком дозволяє використовувати URL аргументи для фільтрації результату і обов'язково має мати пагінацію (не повертати всі результати одразу, а розбивати їх на підсписки фіксованої довжини - "сторінки").
2. Всі URL для доступу до API мають починатись з /api/v1 (наприклад, /api/v1/users, /api/v1/users/13 і т.д). При моделюванні шляхів використовувати загальні рекомендації.
3. Запит GET /developer/v1 має повертати HTML сторінку, у якій коротко задокументувати реалізований REST API версії v1:
   * Доступні ресурси, їх URL, методи доступу до них (які параметри мають, який результат повертається)
   * Спосіб авторизації для доступу до ресурсів
   * Формат і види помилок

**Приклад коду програ**

|  |
| --- |
| **Samples of API routes** |
| router.post('/v1/links', (req, res) => {      if (!mongoose.Types.ObjectId.isValid(req.body.eventId) || !mongoose.Types.ObjectId.isValid(req.body.calendarId))          res.status(400).send({ message: "Bad request" });      else         Promise.all([calendar.getCalendarById(req.body.calendarId), event.getEventById(req.body.eventId)])              .then(resault => {                  if (!resault[0] || !resault[1])                      return undefined;                  else                      return calendar.insertEventToCalendarApi(resault[0], req.body.eventId);              })              .then(success => {                  console.log(success);                  if (success)                      res.status(201).send();                  else                      res.status(400).send({ message: "Bad request" })              })              .catch(err => {                  res.status(500).send({ message: "Server error" });                  console.log(err);              });  })  router.delete('/v1/links', (req, res) => {      if (!mongoose.Types.ObjectId.isValid(req.body.eventId) || !mongoose.Types.ObjectId.isValid(req.body.calendarId))          res.status(400).send({ message: "Bad request" });      else          Promise.all([calendar.getCalendarById(req.body.calendarId), event.getEventById(req.body.eventId)])              .then(resault => {                  if (!resault[0] || !resault[1])                      return undefined;                  else                      return calendar.deleteEventFromCalendarApi(resault[0], req.body.eventId);              })              .then(success => {                  if (success)                      res.status(204).send();                  else                      res.status(400).send({ message: "Bad request" })              })              .catch(err => {                  res.status(500).send({ message: "Server error" });                  console.log(err);              });  }) |

**Приклади роботи програми**

****

**Висновок**

Дізнався основи роботі REST API та basic auth.