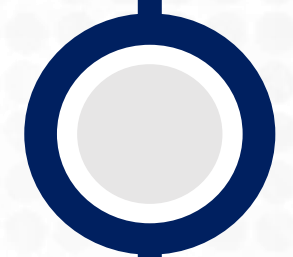


Node.js

Data Handling

04/12/2024

עבודה מול נתונים ב-Express.js



איסוף נתונים מהשתמש באמצעות Query Params

בחלק זה נלמד כיצד נוכל לאסוף נתונים מהשתמש באמצעות שימוש ב-Query Params.
בסיום הנושא תוכלו לענות על השאלות הבאות:

```
router.get("/", (req, res) => {  
  const { name } = req.query;  
  if (!name) return res.json(users);  
  
  const user = users.find(user => user.name === name);  
  
  if (!user) {  
    return res.status(404).json({ message: "User not found" });  
  } else {  
    return res.json(user);  
  }  
});
```

localhost:8080/users/?name=bob

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "name": "bob",  
3   "age": 35  
4 }
```

- מה הם Query Params?
- במה הם שונים מ-Request Params סטנדרטיים?
- מדוע עלינו לייבא את Router לשם כך?
- מה הם היתרונות והחסרונות שבשימוש?
- כיצד נוכל לגשת אל המידע שהשתמש שלח ב-query params?
- כיצד נראית בקשת GET אשר כוללת Query Params?
- מה תפקידו של האופרטור "?" בבקשה לשרת?
- מה תפקידו של האופרטור "=" בבקשה לשרת?
- האם מידע נשמר בשרת בין בקשות?
- מה המשמעות של Stateless?

Query Params - תרגול

צרו תוכנית **Node.js** חדשה עבור הנושא ופתרו את התרגילים לפי הסדר.

```
export const users = [
  {
    name: "John",
    age: 30
  },
  {
    name: "Jane",
    age: 25
  },
  {
    name: "bob",
    age: 35
  }
];
```

תרגיל	תיאור המשימה
Ex-1	צרו רשימת משתמשים בהתאם למפורט בנספח.
Ex-2	צרו תת נתיב עבור משתמשים ובו בקשת GET שתחזיר את המשתמשים אשר קיימים ברשימה שיצרתם.
Ex-3	במידה והמשתמש שולח Query Param בשם "name", חפשו משתמש בעל שם זהה לשם שהמשתמש שלח והחזירו את פרטי המשתמש, במידה והשם לא נמצא ברשימה – החזירו הודעת שגיאה.
Ex-4	צרו בקשת POST שתקבל שם וגיל משתמש ב- Query Params והוסיפו משתמש עם פרטים אלו לרשימה, במידה ולא נשלחו כל הפרטים – החזירו הודעת שגיאה.
Ex-5	בידקו באמצעות בקשת GET שמשתמש שיצרתם אכן נכנס לרשימה. הפסיקו את ריצת השרת והפעילו שוב ושילחו את אותה הבקשה. האם המשתמש שיצרתם עדיין נמצא? מדוע?

איסוף נתונים מהשתמש באמצעות גוף הבקשה

בחלק זה נלמד כיצד נוכל לאסוף נתונים מהשתמש באמצעות שימוש בגוף בקשת HTTP. בסיום הנושא תוכלו לענות על השאלות הבאות:

- מה הוא Request Body?
- איזה סוג של נתונים ניתן לשלוח בגוף של בקשה?
- כיצד ביכולתנו לאפשר שליחה של JSON בגוף הבקשה?
- האם יש הגבלה לכמות הנתונים שניתן לשלוח בבקשה אחת?
- האם ניתן לשנות את הסף של מגבלה זו? כיצד?
- כיצד נוכל לגשת לנתונים שנשלחו בגוף הבקשה?
- מה זה נותן לנו?

```
import express from 'express';
import router from './router/router.js';

const app = express();
const PORT = 8080;

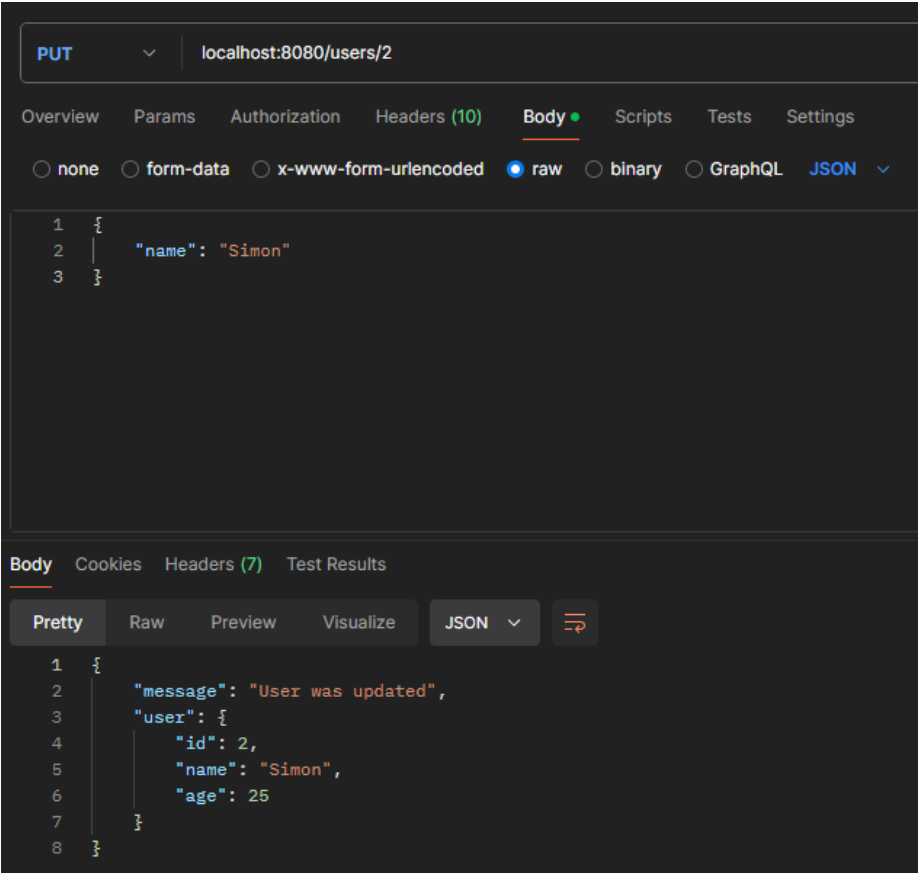
// Add middleware to parse JSON, Maximum request body size is 5MB
app.use(express.json({ limit: '5mb' }));

// Add the router to the app
app.use(router);

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

Request body – תרגול

המשיכו את התרגיל הקודם ופתרו את התרגילים לפי הסדר.



תרגיל	תיאור המשימה
Ex-1	שנו את רשימת המשתמשים כך שיהיה id לכל משתמש.
Ex-2	שנו את נתיב ה-GET כך שידרוש Id ב-Params/Query Params ויחזיר את פרטי המשתמש הרלוונטי, במידה והמשתמש לא נמצא – החזירו הודעת שגיאה.
Ex-3	שנו את נתיב ה-POST כך שיצפה לקבל שם וגיל בגוף הבקשה. צרו id למשתמש כך שיהיה גדול ב-1 מה-Id האחרון ברשימת המשתמשים. הוסיפו את המשתמש לרשימה והחזירו את פרטיו.
Ex-4	שנו את נתיב ה-PUT כך שיצפה לקבל שם ו\או גיל בגוף הבקשה. הנתיב ידרוש Id ב-Params/Query Params. במידה וה-id שהתקבל הוא של משתמש רלוונטי, עדכנו את פרטיו – (במידה ונשלח רק גיל עדכנו רק את הגיל), אם לא קיים משתמש כזה החזירו שגיאה.

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם