Predavanje

Nedelja 3, cas 5

Uvod u Express Framework

- Šta je Express?
  - Express je minimalan i fleksibilan Node.js web framework koji pruža robusne funkcije za izgradnju web i mobilnih aplikacija.
  - Omogućava jednostavno upravljanje rutama, srednjim softverom (middleware), zahteva i odgovora.
- Zašto koristiti Express?
  - Jednostavan za upotrebu i proširenje.
  - Omogućava lako rukovanje HTTP metodama i URL rutama.
  - Pruža moćan skup middleware-a za upravljanje raznim zadacima.

Instalacija Express-a

- Instalacija Express-a se vrši preko npm (Node Package Manager):
  - npm install express

Definisanje Ruta

- Šta su rute u Express-u?
    - Rute su osnovne putanje koje određuju kako će aplikacija reagovati na HTTP zahteve određene metode (GET, POST, PUT, DELETE) na određene krajnje tačke.
- Definisanje osnovnih ruta:
    - Primer definisanja rute koja odgovara na GET zahtev:

Definisanje Ruta

- const express = require('express');
- const app = express();

- app.get('/', (req, res) => {
- res.send('Hello World!');
- });

- app.listen(3000, () => {
- console.log('Server radi na portu 3000');
- });

Pozivanje Hendlera

- Šta su hendleri?
  - Hendleri su funkcije koje se pozivaju kada određena ruta odgovara na zahtev.
  - Hendler funkcije prihvataju dva argumenta: objekat zahteva (req) i objekat odgovora (res).

Pozivanje Hendlera

- Primer hendlera:

```
app.get('/about', (req, res) => {
        res.send('Ovo je About stranica');
});
```

Korišćenje Mogućnosti Express-a kao što su Query i URL Parametri

- Query parametri:
    - Koriste se za slanje dodatnih podataka kroz URL.
    - Primer: /search?query=express

```
app.get('/search', (req, res) => {
  const query = req.query.query;
  res.send(`Rezultati pretrage za: ${query}`);
});
```

Korišćenje Mogućnosti Express-a kao što su Query i URL Parametri

- URL parametri:
  - Definišu dinamične segmente u URL rutama.
  - Primer: /users/:userId

```
app.get('/users/:userId', (req, res) => {
  const userId = req.params.userId;
  res.send(`Korisnički ID: ${userId}`);
});
```

Pisanje osnovnog Middlewarea

- Primer osnovnog middleware-a:
  - Middleware koji loguje informacije o svakom zahtevu:

```
app.use((req, res, next) => {
 console.log(` ${req.method} zahtev na ${req.url}` );
  next();
});
```

Middleware za rukovanje greškama

- Middleware funkcija za rukovanje greškama ima četiri argumenta: err, req, res, next.

```
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).send('Nešto je pošlo po zlu!');
});
```