

# Predavanje

Nedelja 3, cas 6

# Strukturiranje monolitne Express aplikacije po MVC obrascu

- MVC (Model-View-Controller) je arhitektonski obrazac koji deli aplikaciju na tri osnovne komponente: Model, View (Prikaz) i Controller (Kontroler). Ovaj obrazac pomaže u organizaciji koda, omogućava bolju čitljivost, olakšava održavanje i skaliranje aplikacije.
- Ciljevi
  - Razumevanje principa MVC obrasca.
  - Implementacija MVC strukture u monolitnoj Express aplikaciji.
  - Podela odgovornosti između modela, prikaza i kontrolera.

# Strukturiranje monolitne Express aplikacije po MVC obrascu

- Detalji
  - Model: Predstavlja podatke aplikacije i poslovnu logiku. Uključuje validaciju podataka i pristup bazi podataka.
  - View (Prikaz): Odgovoran za prezentaciju podataka korisniku. Može uključivati HTML, CSS i JavaScript za frontend aplikacije.
  - Controller (Kontroler): Posreduje između modela i prikaza. Obrada zahteva, manipulacija podacima i vraćanje odgovora korisniku.

# Strukturiranje monolitne Express aplikacije po MVC obrascu

- Struktura projekta
  - controllers/ – Sadrži kontrolere koji obrađuju zahteve.
  - models/ – Sadrži modele podataka.
  - views/ – Sadrži prikaze za korisnički interfejs.
  - routes/ – Sadrži definisane rute koje povezuju URL putanje sa kontrolerima.

# Kreiranje jednostavne aplikacije koja čuva podatke u memoriji

- Uvod
  - Ova vežba će pokazati kako kreirati jednostavnu aplikaciju koristeći Express framework koja čuva podatke u memoriji. Aplikacija će omogućiti osnovne CRUD operacije (Create, Read, Update, Delete) na kontaktima u adresaru.
- Ciljevi
  - Naučiti osnove korišćenja Express frameworka za razvoj web aplikacija.
  - Implementacija osnovnih CRUD operacija.
  - Korišćenje middleware-a za parsiranje URL kodiranih podataka.

# Kreiranje jednostavne aplikacije koja čuva podatke u memoriji

- Detalji
  - Kreiramo rute za dodavanje novog kontakta (POST), pregled svih kontakata (GET).
  - Podaci o kontaktima se čuvaju u nizu (array) u memoriji servera.

# Uvod u sloj perzistencije, čuvanje podataka u tekstualnim datotekama

- Uvod

- Perzistencija podataka omogućava dugotrajno čuvanje podataka tako da su oni dostupni i nakon restartovanja aplikacije. U ovoj vežbi ćemo modifikovati aplikaciju tako da podatke o kontaktima čuvamo u tekstualnim datotekama koristeći Node.js fs modul.

- Ciljevi

- Razumevanje koncepta perzistencije podataka.
- Naučiti kako koristiti fs modul za čitanje i pisanje fajlova u Node.js-u.
- Implementacija trajne perzistencije podataka korišćenjem tekstualnih fajlova.

# Uvod u sloj perzistencije, čuvanje podataka u tekstualnim datotekama

- Detalji
  - Koristimo fs.promises za asinkrono čitanje i pisanje fajlova.
  - Podaci se čuvaju u JSON formatu u fajlu.
  - Pri svakom zahtevu, podaci se prvo čitaju iz fajla, zatim se ažuriraju i ponovo zapisuju u fajl.



# Kreiranje jednostavnog CRUD-a (POST, GET)

- Uvod

- CRUD operacije su osnovne operacije koje se mogu izvesti nad podacima u aplikaciji: Create (kreiranje), Read (čitanje), Update (ažuriranje) i Delete (brisanje). U ovoj vežbi ćemo implementirati osnovni CRUD sistem koristeći Express framework.

- Ciljevi

- Razumevanje osnovnih CRUD operacija u web aplikacijama.
- Implementacija ruta za različite CRUD operacije.
- Korišćenje middleware-a za parsiranje podataka iz zahteva.

# Kreiranje jednostavnog CRUD-a (POST, GET)

- Detalji

- Create (POST): Ruta za kreiranje novog resursa. Podaci se šalju putem POST zahteva i čuvaju u memoriji ili fajlu.
- Read (GET): Ruta za čitanje resursa. Omogućava pregled svih resursa ili pojedinačnog resursa putem GET zahteva.
- Update (PUT): Ruta za ažuriranje postojećeg resursa. Podaci se šalju putem PUT zahteva i ažuriraju postojeći resurs.
- Delete (DELETE): Ruta za brisanje resursa. Omogućava brisanje resursa putem DELETE zahteva.

# Zaključak

- Ova prezentacija pokriva ključne aspekte razvoja web aplikacija koristeći Express framework i MVC obrazac. Fokus je na osnovnim principima organizacije koda, perzistenciji podataka i implementaciji osnovnih CRUD operacija.