

# Inlämningsuppgift i Webbserverprogrammering 1

## Innehåll

- Inledning
  - Bakgrund
  - Frågeställningen
  - Mål
  - Avgränsning
    - Backend/API
    - Frontend
    - Testning med REST Klient
  - Syfte
  - Vad ska ni leverera?
- Projekt Uppgift
  - API med frontend och backend
  - Funktioner som ska finnas
    - CRUD metoder
    - Egenskaper varje TODO ska ha
- Betygskrav
  - Betyg E
  - Betyg C
  - Betyg A
- Sista inlämningstid

## Inledning

## Bakgrund

Ni är nu i slutet på kursen Webbserverprogrammering 1

# Frågeställningen

Hur kan vi implementera ett API på en backend som vi testar med API-tester innan vi bygger vår frontend.

## Mål

I detta projektet ska ni göra en enkel backend med ett RESTapi som kopplas till en frontend samt har automatiska tester i Insomnia Rest Client eller Postman.

Ert API ska hantera TODOs där ni ska kunna skapa, läsa ut, redigera samt radera TODOs enligt CRUD principen. Era TODOs ska existera i en "in memory database".

## Avgränsning

### Backend/API

Javascript på backend med hjälp av NODE.js med de paket ni behöver och avgränsat till de paket vi gått igenom.

### Frontend

React på frontend och användning av JavaScript med de paket ni behöver och avgränsat till de paket vi gått igenom.

### Testning med REST Klient

Användning av Rest Klient för att skapa manuella och automatiska tester.

Tillåtna klienter:

- Insomnia
- Postman

## Syfte

Att ni ska lära er hur man testar ett API samt hur man kopplar ihop frontend med backend.

## Vad ska ni leverera?

En TODO applikation som kan:

- Skapa nya TODOs
- Visa alla TODOs från databasen
- Visa en enskild TODO från databasen
- Uppdatera en TODO i databasen
  - Uppdatera texten
  - Uppdatera om en todo är klar eller inte
- Radera TODOs från er TODO databas

Förväntad funktionalitet:

- Full funktionalitet ska finnas på backend med korrekta API-anrop.
- Full funktionalitet ska finnas på frontend med korrekta API-anrop, CSS så man ser vad som är vad på sidan (behöver inte vara avancerat).
- En konfigurationsfil med era tester som jag kan importera för att kontrollera era tester.

## Projekt Uppgift

### API med frontend och backend

Skapa en backend med API som kan prata med en frontend samt utför manuella och automatiska tester på backend.

### Funktioner som ska finnas

#### CRUD metoder

- Skapa nya TODOs
- Visa alla TODOs från databasen
- Visa en enskild TODO från databasen

- Uppdatera en TODO i databasen
  - Uppdatera texten
  - Uppdatera om en todo är klar eller inte
- Radera TODOs från er TODO databas

### Egenskaper varje TODO ska ha

- `id` (unikt id på varje TODO)
- `text` (Själva todo meddelandet)
- `done` (är uppgiften klar)
- `assignedTo` (vem ska utföra uppgiften)

Ni får addera mer fält om ni vill.

## Betygskrav

Det är funktionaliteten på ert arbete som betygssätts inte hur gränssnittet ser ut. Dock gör det inget om gränssnittet ser polerat ut då det ger erfarenhet, repetition och något som ni skulle kunna ha som "portfolio projekt".

## Betyg E

- Upprätta en enkel projektplan innan uppgiften påbörjas med till exempel:
  - Hur du tänker försöka lösa uppgiften.
  - Pseudokod.
  - Diagram.
- Konsekvent kod stil:
  - Tydlig namngivning av variabler.
  - Strukturerad och enkelt kommenterad källkod med tillfredsställande resultat.
  - **Lättläst kod.**
- Gör en enklare kontroll så att dina funktioner fungerar som de ska.
- Manuella tester

## Dokumentation

- Gör en enkel dokumentation på ditt arbete, till exempel:
  - Beskriv lite olika lösningar du gjort.

- Beskriv något som var besvärligt att få till.
- Beskriv om du fått byta lösning och varför i sådana fall.
- Använd korrekta begrepp och syntax i dina beskrivningar.

## Programmet skall fungera

# Betyg C

Allt i **Betyg E** ska vara uppfyllt samt:

- Upprätta en genomarbetad projektplan
- Ren struktur på din kod
- Arbeta enligt DRY-principen (Do Not Repeat you're self)
- Kontrollera så att du inte får orimliga resultat beroende på inmatning.
- Se till så att det inte finns programlogiska fel

## Dokumentation

Din dokumentation ska utöver det i **Betyg E** innehålla:

- Beskriv hur du felsökt ditt program när det uppstått problem
- Utvärdera din inlämning
  - Vad gick bra
  - Vad gick dåligt
  - Vad har du lärt dig
- Vilka möjligheter ser du med de kunskaper du fått under kursen.

# Betyg A

Allt i **Betyg C** ska vara uppfyllt samt:

- Upprätta en genomarbetad projektplan som revideras vid behov
- Skriv så tydlig kod så kommentarer ofta blir onödiga.
- Där du använder kommentarer ska de **noggranna och utförliga/tydliga**.
- **Väldigt lättläst kod** dvs någon med mindre-/nybörjar-kunskaper ska kunna läsa och förstå er kod.

## Dokumentation

Din dokumentation ska utöver det i **Betyg C** innehålla:

- Motivera varför du valt en specifik lösning.
- Lämna förslag på förbättringar av din kod.
- Lämna exempel på lösningar du valde att **inte** implementera

# Sista inlämningstid

26 September 2021 kl 23:59