



Repetition




Agenda

- useEffect
 - repetition
 - setInterval och setTimeout
 - övning
 - useEffect med Server övning
- Server med DB
 - repetition
 - övning
- Genomgripande övningar
 - Todo applikation
 - Signup med Login


useEffect





useEffect - repetition

```
useEffect(() => {  
  // [] Gör så att effekten bara kör första rendering  
}, []);
```




useEffect - repetition

```
useEffect(() => {  
  // Tom gör så att effekten bara kör första rendering  
});
```



useEffect

```
useEffect(() => {  
  // [dep] gör så att effekten kör första rendering och när dep ändras  
}, [dep]);
```



useEffect - repetition

```
useEffect(() => {  
  // Denna funktion körs när komponenten försvinner helt (städar upp efter)  
  return () => {  
    }  
  }, []);
```



setInterval och setTimeout

Vi kan städa upp lyssnare med clearTimeout och clearInterval

```
const id = setTimeout(() => console.log("1 time"))
clearTimeout(id);
```

```
const id = setInterval(() => console.log("Many times"), 1000);
clearInterval(id);
```




useEffect övning

Gör en nedräknare från 10 - 0 (sekunder)

- Skapa en komponent Timer med prop (startValue) och ett state (timeRemaining)
- I App.jsx rendera Timer komponenten (med startValue = 0)
- I Timer.jsx anropa setInterval för att uppdatera timeRemaining varje sekund
 - Gör detta i en useEffect, och anropa den bara 1 gång när Timer renderar första gången
- Stoppa timer när komponenten har renderat klart (använd clearInterval)



useEffect med Server övning

Server

- Skapa en post route till vilken man kan skicka en text som sparas i en variabel (text)
 - Ge värdet "hello world" i början
- Skapa en get route från vilken man kan hämta värdet i text variabeln

Klient

- Skapa en komponent (Text) som hämtar värdet i en useEffect från servern och visar i en div
- Lägg till ett formulär med vilken kan skicka en text till server
 - onSubmit
- Prova att du kan stänga av klienten och sätta på igen och att värdet är samma

Server med DB





Server med DB repetition

Demo av 2 olika servrar med DB

- Skapa en databas som håller reda på ett tal (get och post)
- Skapa en todo server



Server med DB övning

Låt nu värdet av text sparas i en databas (mongoose)

- Lägg in i förra övningens server del en databas med hjälp av mongoose
- Prova att du kan stänga av både klient och server och att texten är oförändrad

Genomgripande övningar



Todo applikation - övning

Todo List

Add Todo

Add new todo

Add

my first todo

☒ ☐

my second todo

☒ ☐

my third todo

☒ ☐

my fourth todo

☒ ☐

Todo applikation - övning


Plan

- Gör först statiska komponenter för utseende
- Gör sedan server(get, post, put, delete)
 - Testa server
- Introducera Databas
 - Testa server igen
- Låt olika komponenter hämta, skapa, ändra och ta bort genom server




The screenshot shows a web application titled "Todo List". At the top, there is a section labeled "Add Todo" containing a text input field with the placeholder "Add new todo" and a blue "Add" button. Below this, there is a list of four todo items, each in a white box with a light gray border. The items are "my first todo", "my second todo", "my third todo", and "my fourth todo". To the right of each item are two small square buttons: a green one with a white checkmark and a red one with a white 'x'. The green checkmark button for "my third todo" is highlighted with a green border.

Signup och Login övning



Member Login



Register

Signup och Login övning

Plan

- Gör först statiska komponenter för utseende
 - 2 komponenter Login, Register
- Gör sedan server(get, post, put, delete)
 - Testa server
- Introducera Databas
 - Testa server igen
- Låt olika komponenter hämta, skapa, ändra och ta bort genom server

