

Internetprogrammering

DD1386

Föreläsning 11

Innehåll

- Promise
- Hantering av kakor
- Router

Promise

- Ett objekt som representerar resultat av en asynkron operation.
- T.ex man vill hämta data från en enhet/eyetsem (t.ex databas, filsystemet, ...) men man vill inte att resten av applikationen ska blockeras av medan data hämtas då ska man utföra hämtningen asynkront.
- Ett Promise objekt kan ha följande tillstånd:
 - Pending
 - Fullfilled
 - Rejected
- Se exemplet

Promise example

```
new Promise((resolve, reject) => {  
  console.log('Initial');  
  readFromKeyboard(resolve, reject);  
}).then((resolv_answ) => {  
  console.log('resolved', resolv_answ);  
  console.log('Vad roligt att du kommer!!!');  
}).catch((reject_answ) => {  
  console.log('rejected', reject_answ);  
  console.log('Tråkigt');  
}).then(() => {  
  console.log('Vilken film kan vi bestämma sedan!');  
});  
console.log('\njag gör något annat här medan du svarar...');
```

```
function readFromKeyboard(resolve, reject){  
  const readline = require('readline');  
  const { stdin: input, stdout: output } = require('process');  
  const rl = readline.createInterface({ input, output });  
  rl.question('Ska vi på bio på torsdag?', (answer) => {  
    rl.close();  
    if( answer=='nej')  
      reject(answer);  
    else  
      resolve(answer);  
  });  
}
```

Kakor

- För hämtning av kakor används request-objektet.
- För att skapa kakor används response-objektet.
- Modulerna cookie, cookie-parser kan användas för parsning av kakor

Se exempel på canvas!

Expressjs: Router

- Routing innebär hur servern väljer att svara på en URI i en request.
- Objektet router är en middleware som finns i Expressjs.

Se exempel på canvas