



# Introduktion till Webbframverk för JavaScript



# Förväntning och mål

- Förväntningar?
- Genomgång av kursplanering
- Genomgång av inlämningsuppgift
  - Rekommendationer om när ni ska börja med delar i kursplanering
- Frågor



# OBS

- Denna kurs är viktig
- Gör uppgifter (enligt kursplanering)
- Läs inför lektioner (enligt kursplanering)
- Och arbeta under lektionsfria (enligt kursplanering)
- React återkommer i nästan alla kommande kurser



# Försäkran

Vi kommer gå igenom

- Repetera grunder i JavaScript
- Lära oss React enligt “best practice” från grunden
- Göra många olika små applikationer (video och utmaningar)
  - Intervju frågor
- Lära oss om ekosystem i React (bibliotek)
- Kodandet blir mer verkligt

# Chatgpt

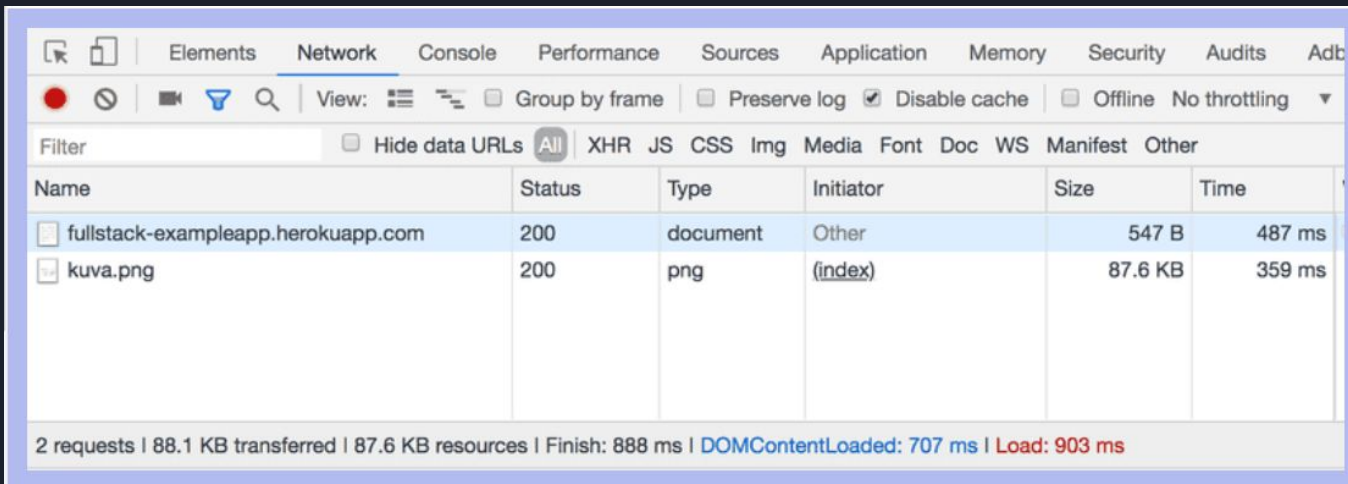
Diskussion om ChatGPT...



# Grunder i webbramverk

Anatomi av en GET request (rita)

`https://studies.cs.helsinki.fi/exampleapp.`



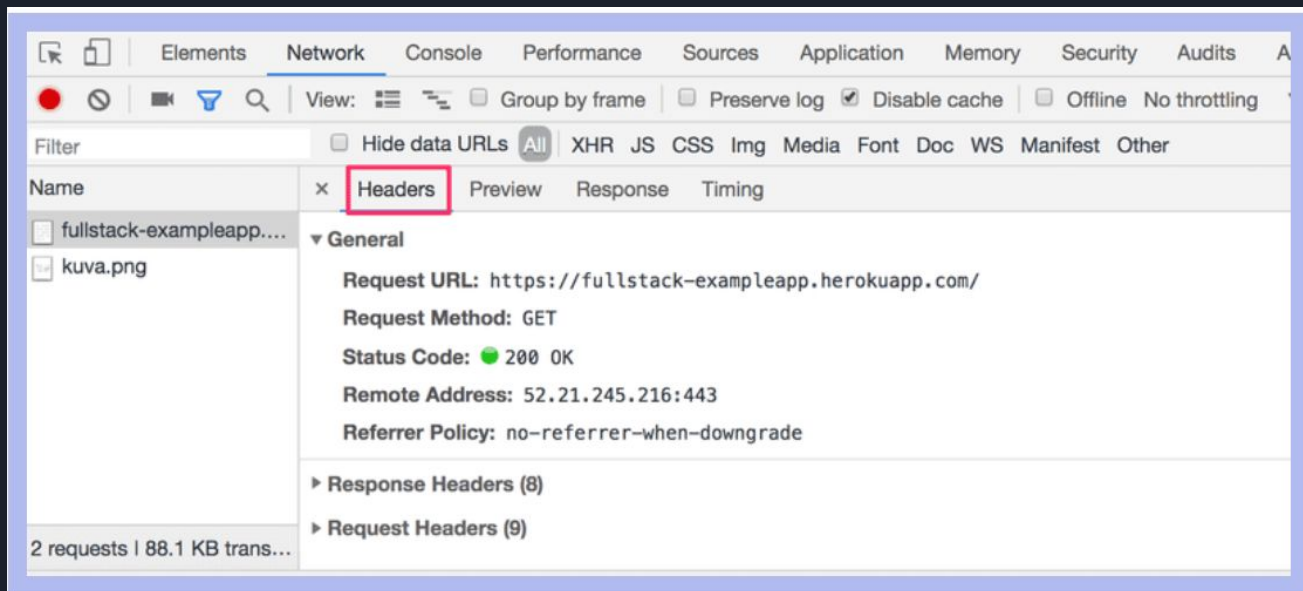
The screenshot shows the Chrome DevTools Network tab. The top bar includes tabs for Elements, Network, Console, Performance, Sources, Application, Memory, Security, Audits, and Address Bar. The Network tab is active, showing a list of requests. The first request is to fullstack-exampleapp.herokuapp.com, which is a document of 547 B, taking 487 ms. The second request is for kuva.png, a 87.6 KB resource, taking 359 ms. The bottom status bar indicates 2 requests, 88.1 KB transferred, 87.6 KB resources, and a total load time of 903 ms.

Name	Status	Type	Initiator	Size	Time
fullstack-exampleapp.herokuapp.com	200	document	Other	547 B	487 ms
kuva.png	200	png	(index)	87.6 KB	359 ms

2 requests | 88.1 KB transferred | 87.6 KB resources | Finish: 888 ms | DOMContentLoaded: 707 ms | Load: 903 ms

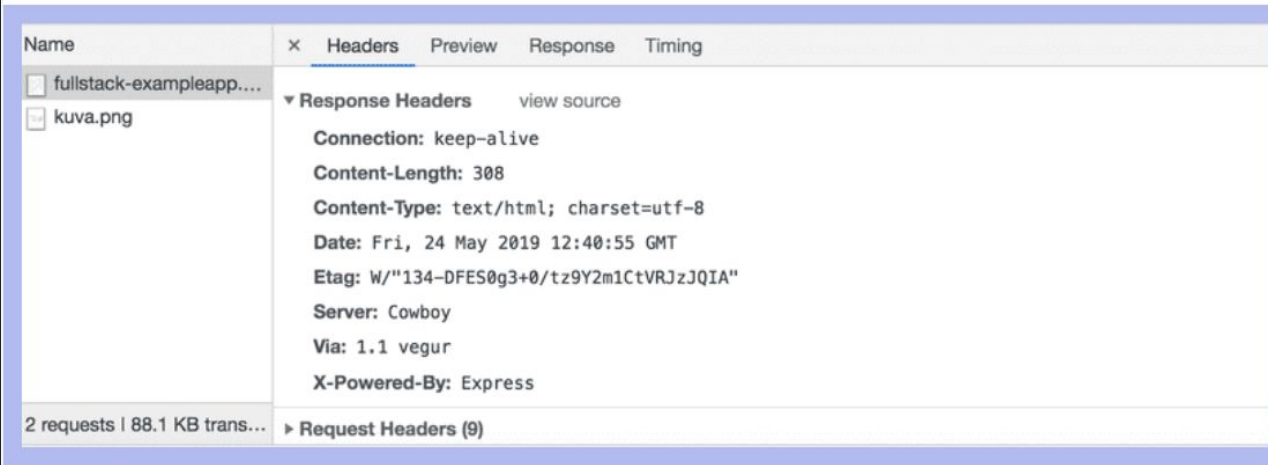
# Grunder i webbapplikationer

Man kan se vad som skickats i request om man tittar på headers



# Grunder i webbapplikationer

## Response headers



The screenshot shows a web browser's developer tools interface. The 'Name' column on the left lists two requests: 'fullstack-exampleapp....' and 'kuva.png'. The 'fullstack-exampleapp....' request is selected, and the 'Headers' tab is active. The 'Response Headers' section is expanded, showing the following information:

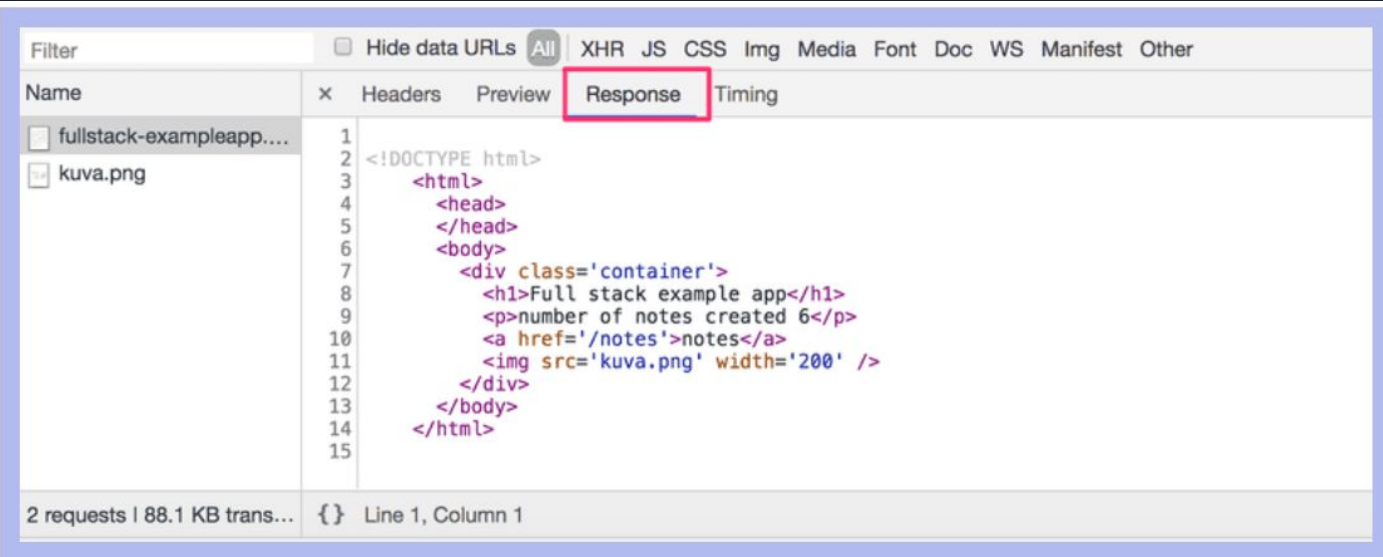
- Connection: keep-alive
- Content-Length: 308
- Content-Type: text/html; charset=utf-8
- Date: Fri, 24 May 2019 12:40:55 GMT
- Etag: W/"134-DFES0g3+0/tz9Y2m1CtVRJzJQIA"
- Server: Cowboy
- Via: 1.1 vegur
- X-Powered-By: Express

At the bottom of the interface, it shows '2 requests | 88.1 KB trans...' and a link to 'Request Headers (9)'.



# Grunder i webbapplikationer

## Response content

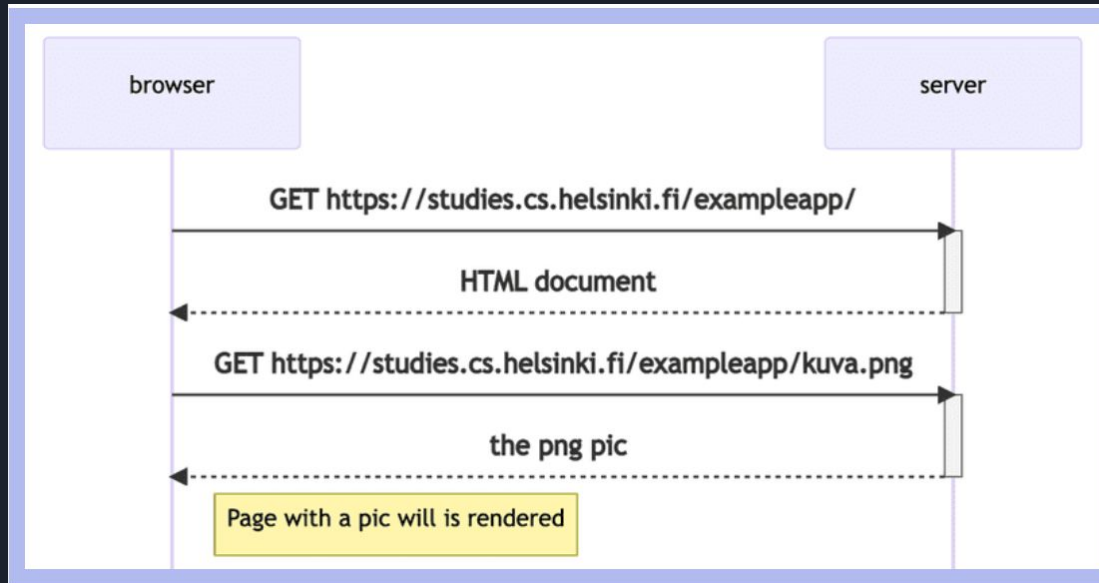


The screenshot shows a web browser's developer tools interface. The 'Response' tab is selected and highlighted with a red rectangle. The left sidebar shows a list of requests, with 'fullstack-exampleapp....' and 'kuva.png' visible. The main pane displays the response content, which is an HTML document. The status bar at the bottom indicates '2 requests | 88.1 KB trans...' and 'Line 1, Column 1'.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <div class='container'>
7       <h1>Full stack example app</h1>
8       <p>number of notes created 6</p>
9       <a href='/notes'>notes</a>
10      <img src='kuva.png' width='200' />
11    </div>
12  </body>
13 </html>
```

# Grunder i webbapplikationer

Hela flödet





# Grunder i webbapplikationer

## Traditionell webbapplikation

Om vi skulle göra en egen sådan server applikation  
i node med olika sidor (inte som bild)

```
const getFrontPageHtml = noteCount => {  
  return `  
    <!DOCTYPE html>  
    <html>  
      <head>  
      </head>  
      <body>  
        <div class='container'>  
          <h1>Full stack example app</h1>  
          <p>number of notes created ${noteCount}</p>  
          <a href='/notes'>notes</a>  
          <img src='kuva.png' width='200' />  
        </div>  
      </body>  
    </html>  
  `;  
}  
  
app.get('/', (req, res) => {  
  const page = getFrontPageHtml(notes.length)  
  res.send(page)  
})
```

# Grunder i webbapplikationer

Om vi går till <https://studies.cs.helsinki.fi/exampleapp/>

## Full stack example app

number of notes created 100

 mongoDB.  Express

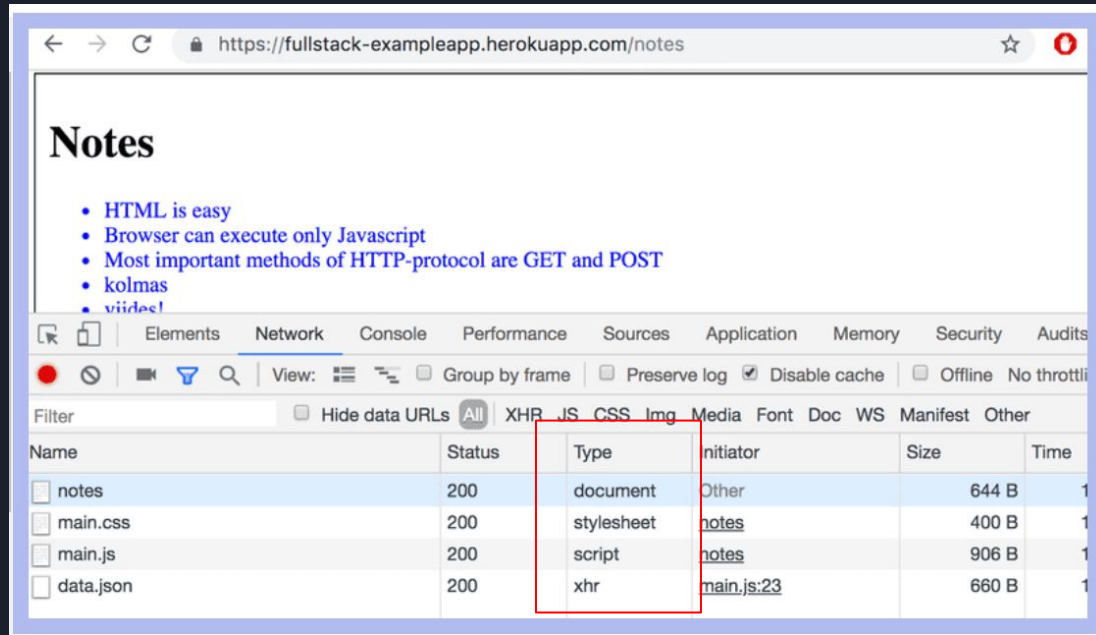
 React  node.js

 Redux  Webpack

[notes](#)

# Grunder i webbapplikationer

Om vi klickar på notes, olika typer av requests



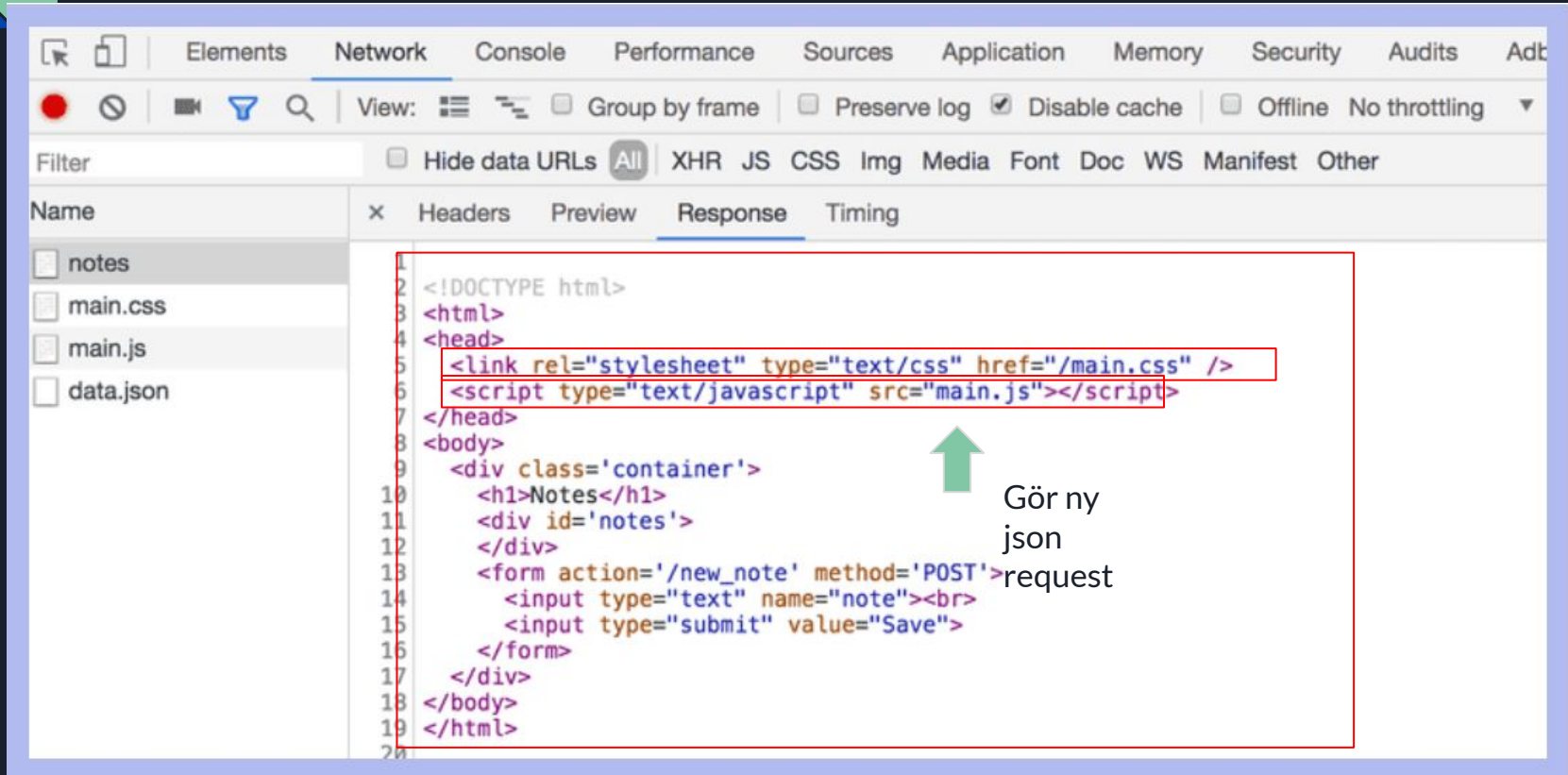
The screenshot shows a web browser at the URL `https://fullstack-exampleapp.herokuapp.com/notes`. The page content is titled "Notes" and contains a bulleted list:

- HTML is easy
- Browser can execute only Javascript
- Most important methods of HTTP-protocol are GET and POST
- kolmas
- viides!

Below the page content, the browser's developer tools are open to the "Network" tab. The "Filter" dropdown is set to "All". The "Type" column in the network log is highlighted with a red box. The network log shows the following requests:

Name	Status	Type	Initiator	Size	Time
notes	200	document	Other	644 B	1
main.css	200	stylesheet	notes	400 B	1
main.js	200	script	notes	906 B	1
data.json	200	xhr	main.js:23	660 B	1

# Grunder i webbapplikationer



The screenshot shows a web browser's developer tools interface. The 'Network' tab is selected, and a list of resources is shown on the left: 'notes', 'main.css', 'main.js', and 'data.json'. The 'notes' resource is selected, and its 'Response' is displayed in the main pane. The response is an HTML document. A red box highlights the following code in the response:

```
<link rel="stylesheet" type="text/css" href="/main.css" />
<script type="text/javascript" src="main.js"></script>
```

A green arrow points from the text 'Gör ny json request' to the highlighted code, indicating that the browser is making a new JSON request.

Elements   Network   Console   Performance   Sources   Application   Memory   Security   Audits   Adb

View: [Icons] [Icons] [Icons] Group by frame [ ] Preserve log [x] Disable cache [ ] Offline No throttling ▼

Filter [ ] Hide data URLs [All] XHR JS CSS Img Media Font Doc WS Manifest Other

Name x Headers Preview **Response** Timing

notes  
main.css  
main.js  
data.json

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" type="text/css" href="/main.css" />
5   <script type="text/javascript" src="main.js"></script>
6 </head>
7 <body>
8   <div class='container'>
9     <h1>Notes</h1>
10    <div id='notes'>
11    </div>
12    <form action='/new_note' method='POST'>
13      <input type="text" name="note"><br>
14      <input type="submit" value="Save">
15    </form>
16  </div>
17 </body>
18 </html>
```

Gör ny  
json  
request



# Grunderna i webbapplikationer

main.css

```
.container {  
  padding: 10px;  
  border: 1px solid;  
}  
  
.notes {  
  color: blue;  
}
```

# Grunder i webbapplikationer

main.js

```
var xhttp = new XMLHttpRequest()

xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    const data = JSON.parse(this.responseText)
    console.log(data)

    var ul = document.createElement('ul')
    ul.setAttribute('class', 'notes')

    data.forEach(function(note) {
      var li = document.createElement('li')

      ul.appendChild(li)
      li.appendChild(document.createTextNode(note.content))
    })

    document.getElementById('notes').appendChild(ul)
  }
}

xhttp.open('GET', '/data.json', true)
xhttp.send()
```

Callback

Caller



# Grunder i webbapplikationer

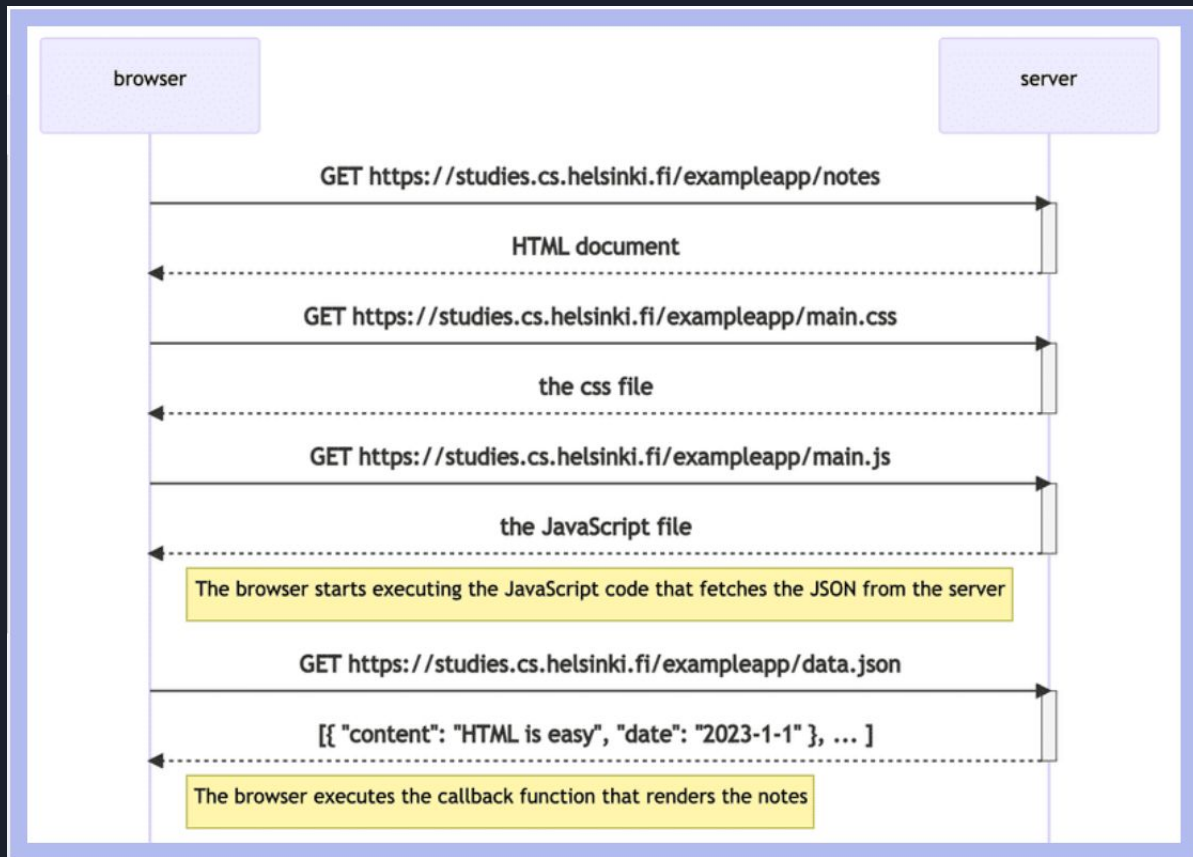
`https://studies.cs.helsinki.fi/exampleapp/data.json`



A screenshot of a web browser window. The address bar shows the URL `https://fullstack-exampleapp.herokuapp.com/data.json`. The page content displays a JSON array of objects, each containing a 'content' field and a 'date' field. The objects are as follows:

- `{"content": "HTML is easy", "date": "2019-05-23T17:30:31.098Z"}`
- `{"content": "Browser can execute only Javascript", "date": "2019-05-23T18:39:34.091Z"}`
- `{"content": "Most important methods of HTTP-protocol are GET and POST", "date": "2019-05-23T19:20:14.298Z"}`
- `{"content": "kolmas", "date": "2019-05-24T12:37:59.879Z"}`
- `{"content": "viides!", "date": "2019-05-24T12:38:14.734Z"}`
- `{"content": "asd", "date": "2019-05-24T12:40:32.200Z"}`
- `{"content": "Homma toimii", "date": "2019-05-24T13:13:24.418Z"}`
- `{"content": "homma ok", "date": "2019-05-24T13:13:44.632Z"}`

# Grunder i webbapplikationer





# Grunder i webbapplikationer

Demo: Gör applikation



# Grunder i webbapplikationer

Övning: Gör likadan applikation mha slides