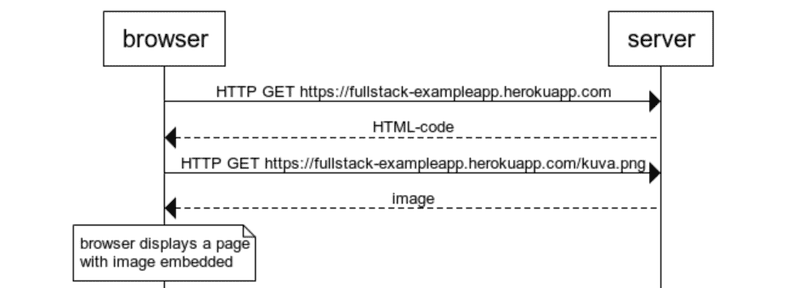
**University of Helsinki –** [**Intro to Modern Web Dev (JavaScript)**](https://fullstackopen.com/en/)

# Part 0 – [**Fundamentals of Web Apps**](https://fullstackopen.com/en/part0)

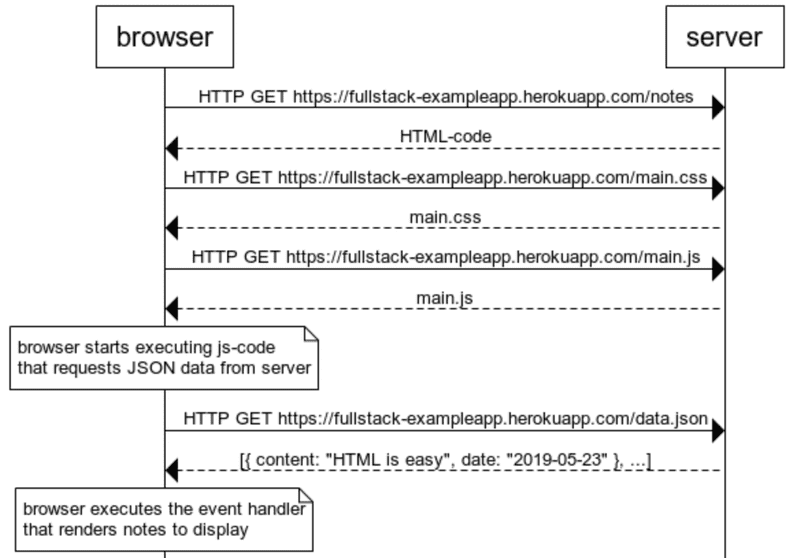
* All based through HTTP apis, GET requests download:
  + Images, html, js, css, json responses
* All requests and responses have headers for authentication, status code
* HTML template strings can also be stored in a database and fetched and rendered with dynamic variables
* You can use JS to render in dynamically generated html elements dependent on JSON responses.

**Event Handlers and Callback functions**

* Actual xhttp request send after function which deals with response
* var xhttp = new XMLHttpRequest()
* xhttp.onreadystatechange = function() {
* // code that takes care of the server response
* }
* xhttp.open('GET', '/data.json', true)
* xhttp.send()
* steadyState == 4 and status == 200 indicates that a request opereation is successfully completed.
* Event Handler functions are called callback functions. Application code does not invoke function, browser does, once event has occurred.

**Document Object Model (DOM)**

* Html takes implicit tree like structure
* DOM is an api to programmatically modify the element trees

**POST requests and forms:**

* http post request appends data to destination and then reloads the page etc.

**Traditional Web Apps vs Single Page App (SPA)**

* Don’t fetch all pages separately but instead use one HTML page and dynamically manipulate contents using browser side JS.
* Don’t need to re-render the page each time something changes

Backend – Node.js runtime environment

Frontend – React and Redux libraries

# Part 1 – [Intro to React](https://fullstackopen.com/en/part1)

## Intro to React