

Web Engineering + Design 1

NODE.JS

Michael Gfeller

Die Teilnehmer...

- ... wissen, dass JavaScript auch auf einem Server ausgeführt werden kann
- ... können den Code eines Node.Js Servers interpretieren – im Umfang dieser Vorlesung
- ... erkennen die HTTP Konzepte in einem Node.Js Server
- ... können JSON interpretieren und definieren und erkennen fehlerhaftes JSON.
- ... können ein Favicon korrekt einbinden

Abgrenzungen ...

- ... die Teilnehmer können keinen Node.Js Server implementieren
 - WED2

Inhaltsverzeichnis

- **Node.Js**
 - Demo
- **JSON**
- **Favicon**
- **Ausblick**

DEMO

■ Inhalt der Demo

- «Hello World» Beispiel
- Typische Aufgaben von einem Web-Server
 - Ausliefern von Index.html
 - Ausliefern von Assets (Bilder, ...)
 - Ausliefern von Daten (JSON)
 - Annahme von Parameter über GET
 - POST in WED2 😊
 - 404er
 - Ausliefern von JSON
 - Favicon



Danke für die Interesse an der Party. Melde dich mit deinem Name an:



Danke für deine Anmeldung Michael

Request (IncomingMessage)

- **method**

- GET, PUT, POST, ...

- **url**

- Angefragte URL z.B. /help or help?page=1

Response

■ writeHead

- `response.writeHead(200, {'Content-Length': body.length, 'Content-Type': 'text/plain' });`

■ setHeader

- `response.setHeader("Content-Type", "text/html");`

■ statusCode

- `response.statusCode = 404;`

■ statusMessage

- `response.statusMessage = 'Not found';`

■ write

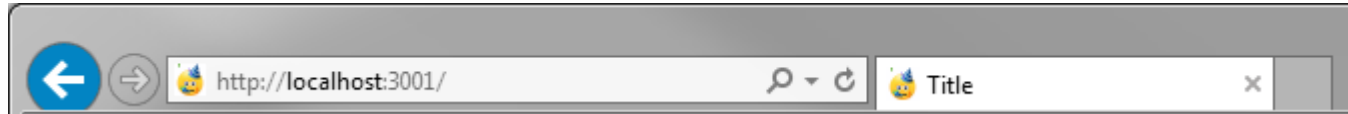
- `response.write("Data")`

■ end

- `response.end("Data")`

- <https://nodejs.org/>
- <https://nodejs.org/api/http.html>
- https://nodejs.org/api/http.html#http_class_http_serverresponse
- https://nodejs.org/api/http.html#http_class_http_clientrequest

FAVICON



■ Kurz für «favorite icon»

- shortcut icon, website icon, tab icon, URL icon oder bookmark icon.

■ Grösse

- Quadratisch: 16x16
- Funktionieren in den meisten Browsern: 32x32, 64x64

■ Format

- *.ico, *.png, *.jpg, *.gif (nicht animiert)

■ Einbinden mit dem Link-Tag

```
<link rel="shortcut icon" type="image/gif" href="facepalm.gif" />
```

■ Oder das File auf dem Server am richtigen Ort platzieren



■ Oder beide Varianten kombiniert

- Rückwärtskompatibilität + Vorteil von Variante 1

JSON

JSON (JavaScript Object Notation)

- **JSON ist ein Daten-Austauschformat.**
- **Wird verwendet um Daten zu senden, speichern (z.B. DOM storages, cookies, HTTP requests).**
- **Hat im Web XML verdrängt**
- **Wird oft mit AJAX verwendet**
- **Unterschied zwischen JSON und JavaScript Object**
 - all identifiers and all strings are written between double quotes
 - the only allowed data types are: String, Number, Boolean, Array, Object and null.
 - NaN, Infinity und -Infinity are converted to null
 - there is no defined representation for Date, Error, Regular Expression, and Function objects.
 - there are no comments
 - trailing commas are forbidden
- **Content-Type: application/json**
- **JSON-Helper: JSON.parse & JSON.stringify**

- **JSON - Array ist Valid sollte aber nicht verwendet da:**
 - **JSON Hijacking**
 - **Erweiterbarkeit**

```
//Gültig aber nicht ideal & gefährlich:  
JSON.stringify([1,2,4,5]);
```

```
//besser:  
JSON.stringify({elements : [1,2,4,5]});
```

- **Validator: <http://jsonlint.com/>**

AUSBLICK AUF WED2

- **Parallelität auf dem Server**
- **Mehr Node.Js**
- **NPM**
- **Express.Js**
- **REST-API Implementieren**
- **... mehr Server...**