



# Web Engineering 1

## Labor zur Vorlesung

„Dynamische Webseiten mit JavaScript“

Prof. Dr. Schneider

## 1. Erweitern der Laborumgebung

### 1.1. GitHub – Repository und Webstorm

Um verschiedene Softwarestände abzuspeichern, kommen sogenannte Software Versionsverwaltungstools (Software Configuration Management SCM) zum Einsatz. Ein sehr weit verbreitetes SCM-Tool ist [git](#). Ein auf [git](#) basierender ebenfalls kostenfreier Cloud-Service namens [GitHub](#), ermöglicht eigens erstellte Software in der Cloud zu speichern und mit einem Team oder einer ganzen Community zu teilen.

- (1) Erstellen Sie sich einen kostenfreien Account auf GitHub. Sie werden hier aufgefordert eine 2-Faktor-Authentifizierung einzurichten.
- (2) Erstellen Sie sich dann in Ihrem GitHub Account unter [Settings • Developer Settings • Personal Access Tokens](#) einen Access Token.
- (3) Wechseln Sie zurück in Ihre Webstorm IDE und wählen dort unter dem Menüpunkte [Git • GitHub](#) den Menüpunkt "[Share Projekt on GitHub](#)".
- (4) Ihre Projektdateien werden jetzt in das GitHub-Repository geladen.

### 1.2. NodeJS – Webserver

Um die Möglichkeiten von clientseitigen Web-Applikationen zu steigern (z.B.: Zugriff auf Dateien im Filesystem), können Sie sich lokal auf Ihrem Client die Software [node.js](#) installieren. [node.js](#) ist eine [JavaScript-Laufzeitumgebung](#). Sie ermöglicht JavaScripts auch außerhalb des Webbrowsers ausführen zu können.

Damit ein Browser auf die [node.js](#)-Laufzeitumgebung zugreifen kann, installieren Sie einen einfachen [node.js](#)-Webserver.

Aktualisieren Sie zuerst Ihre Software auf Ihrem Ubuntu-Webserver

```
$sudo apt update
```

Anschließend können Sie [node.js](#) mittels des folgenden Befehles installieren

```
$sudo apt install nodejs
```

Überprüfen Sie, ob Ihre Installation erfolgreich war, indem Sie die JavaScript-Engine mit der Option `-v` aufrufen

```
$nodejs -v
```

Um node.js mit [Bibliotheken](#) zu erweitern, installieren Sie sich den sogenannten [Node Package Manager](#) (kurz [npm](#)) installieren.

```
$sudo apt install npm
```

Auch die npm-Installation können Sie durch den Versions-Aufruf auf Funktionsfähigkeit prüfen

```
$npm -v
```

Um Zugriff auf Dateien im Filesystem mittels **nodejs** zu erhalten, können Sie sich das Modul „node:fs“ installieren.

```
$npm install fs
```

Dateien können dann beispielsweise wie folgt im Filesystem gelesen werden

```
const fs = require('fs');

fs.readFile('./user.json', (err, data) => {
  if (err) {
    console.log(err)
    return
  }
  //Daten in der Datei ausgeben
  console.log(data)
})
```

## 2. Erstellen von dynamischen Webseiten

Erstellen Sie jetzt eine Webseite, die ihren Benutzern eine Webseite zum Thema Web-Engineering zur Verfügung stellt. Ihr Benutzer muss sich auf ihrer Willkommenseite

zuerst anmelden. Besitzt er keinen Benutzer wird ihm ein Registrierungsformular zur Verfügung stellt. Nach erfolgreicher Anmeldung wird eine Webseite geöffnet die Information zum ausgewählten Interessenschwerpunkt des angemeldeten Benutzers enthält.

## 2.1. Registrierungs- und Login-Formular

1. Erstellen Sie ihre Startseite. Ihre Startseite besteht aus einem Begrüßungstext und einem Textfeld für den Benutzernamen, einem Passwort-Feld für das Passwort und einem [login](#)-Button. Beim Drücken des Login-Buttons überprüft ein JavaScript-Programm anhand einer JSON-Datei, ob die Kombination aus Passwort und Benutzername richtig ist. Die Startseite enthält weiterhin einen Link auf die Registrierungsseite.
2. Die Registrierungsseite enthält ein Registrierungsformular. Die minimalen Anforderungen an das Registrierungsformular sind

1. Textfelder für Vornamen, Nachnamen und einem selbst gewählten Benutzernamen. Die Eingaben dürfen nur aus Groß- und Kleinbuchstaben und einem "-" oder "\_" bestehen. Der Vorname und Nachname und das Passwort darf aus höchstens 20 Zeichen bestehen.
2. Passwortfeld, Passwort muss mindestens 8 Zeichen lang sein und muss mindestens eine Zahl und ein Sonderzeichen aus "!#,+\_-?" enthalten.
3. Radioboxen für das Geschlecht.
4. Select Box die verschiedenen Interessenskategorien auflistet (3 Stück)
5. Eine Textarea für Kommentare (max. 100 Zeichen, optional)
6. Eine Checkbox für das akzeptieren der Nutzungsbedingungen.
7. Einen Button Registrierung.

3. Drückt der Benutzer den Button Registrierung überprüfen Sie, ob alle Felder richtig ausgefüllt wurden. Wenn nein, geben Sie dem Benutzer ein geeignetes Feedback, wenn ja beglückwünschen Sie den Benutzer mit Anrede, Vorname und Nachname und geben ihm die Möglichkeit sich auf der Startseite anzumelden. Die Registrierungsdaten speichern Sie in einer serverseitigen Datei im JSON-Format.

4. Meldet sich der Benutzer über die Login-Seite an, überprüfen Sie seine Eingabe gegen die JSON-Datei.
5. Ist die Kombination aus Benutzernamen und Passwort richtig, starten Sie eine Webseite die einen Inhalt zum Interessengebiet des Benutzers zeigt. Die Webseite ist so aufgebaut, das Sie mehrere HTML-Elemente zu unterschiedlichen Themen (HTML5, CSS, JavaScript, FlexBox, GridBox) enthält. Nur das HTML-Element, dass dem Themengebiet des Benutzers entspricht wird angezeigt (**Überlagerung** mit **verschiedenen Ebenen mit z-Index**), die anderen HTML-Elemente sind unsichtbar.
6. Auf der Antwortwebseite zeigen Sie zudem an geeigneter Stelle die Geokoordinaten des Benutzers und die aktuelle Systemzeit an.
7. Gestalten Sie ihre Webseite mit CSS-Stylesheets ansprechend. Jedes Themengebiet besitzt eine andere Hintergrundfarbe.