

Modul 151: Fullstack-Webanwendungen

Praktiken vom Freitag, den 18.11.2022

Rechtliche Hinweise

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Alle Rechte vorbehalten

© 2022 Jörg Bergmann

Ablaufplan der FWA-Vorlesungen

1	07.10.	Projektinitiative, Stakeholder, Designmodelle, Architektur Überblick
2	14.10.	Risikomanagement, 8 Fallacies, C/S-Anforderungen, System Kontext Diagramm
3	21.10.	Komponentenmodell (funktional und physisch) mit Notationen, Node.js-Module
4	28.10.	HTTP-Server, ECMA, Grundfunktionalitäten (neu, ändern, auflisten, löschen, ausführen)
5	04.11.	Express, CSS, Template-Engines, FrontEnd-Frameworks (Vue.js, Nest.js, TypeScript)
6	11.11.	Logisches und phys. Datenmodell, ERD, Data Dictionary, Anbindung von Datenbanken
7	18.11.	Sicherheitsaspekte (ITIL) Benutzergruppen, Rollen und Verantwortlichkeiten, Standorte, HTTPS-Modul Authentifizierung(Passport) & Verschlüsselung, Session-Handling
8	25.11.	REST Server (Postman, cURL), Open-API-Spezifikation und Dokumentation (Swagger), XML-/JSON-Datenformate, Present.-, Exec.- und Data-Deployment Units
9	02.12.	Echtzeit Webappl., Websocket, Socket.IO, jQuery-Framework, Skalierbarkeit, Load-Balancing, Performance, u.a. bei Regulären Ausdrücken
10	09.12.	Asynchrone Programmierung, Fork von Prozessen, Promises-Konzept
11	16.12.	RxJS Asynchrone Kommunikation, API Socket-Server (TCP/IP)
12	23.12.	Streams-Konzept, Arbeiten mit Dateien in der CS-Kommunikation, Gulp-Build-Process
13	06.01. 2023	C/S-Qualitätssicherung mit dem Viability Assessment, Unit- und Integrationstest, Test-Werkzeuge, Projekt-Abnahme und Übergabe an Wartungs- und Betriebsorganisation
14	13.01.	Software-Packaging (npm, Docker) und -Deployment (Gulp), Implementierungsplan, Stage-Konzept
15	20.01.	Betriebsmodell (Operations Model) funktional und physisch, Service-Level-Messungen und -Analyse, ITIL Prozessbereiche, Disaster Recovery

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Laden der für die Übungen benötigten Node.js-Module

Die Übung sollen den Umgang mit Node.js-Modulen HTTPS und passport trainieren.

7.1 Praktiken FWA - HTTPS & passport

Laden der für die Übungen benötigten Node.js-Module

Einzelschritte dazu:

- Bevor Sie Pakete installieren können, sollten Sie Ihre C/S-Anwendung korrekt initialisieren. Dies geschieht mit dem Kommando: `npm init`.
- Laden sie das Modul `formidable` für z.B. Datei-Upload-Funktionalität mit dem Befehl: `npm install formidable`.
- Laden sie das Modul `pug` für z.B. HTTP-Template-Funktionalität mit dem Befehl: `npm install pug`.
- Laden sie das Modul `morgan` für Logging-Funktionalität mit dem Befehl: `npm install morgan`.
- Laden sie das Modul `cheerio` für HTML-Parser-Funktionalität mit dem Befehl: `npm install cheerio`.
- Laden sie das Modul `express` für die Web-Application-Framework- Funktionalität mit dem Befehl: `npm install express`.
- Laden sie das Modul `passport` für Authentifizierungs- und Session-Handling-Funktionalität mit dem Befehl: `npm install passport`.

7.1 Praktiken FWA - HTTPS & passport

Laden der für die Übungen benötigten Node.js-Module

Fortsetzung:

- Laden sie das Modul `sqlite3` für den relationale Datenbank SQLite-Treiber mit dem Befehl: `npm install sqlite3`.
- Laden sie das Modul `express-session` für das benutzer-Session-Handling mit Express-Framework mit dem Befehl: `npm install express-session`.
- Laden sie das Modul `connect-ensure-login` für das Absichern von Ressourcen für nur bestimmte Benutzer mit dem Befehl: `npm install connect-ensure-login`.
- Laden sie das Modul `passport-local` für die local-Strategy mit dem Befehl: `npm install passport-local`.
- Laden sie das Modul `body-parser` für HTML-Body-Parser-Funktionalität mit dem Befehl: `npm install body-parser`.
- Laden sie das Modul `crypto` für die Verschlüsselung von Benutzer-Passwörtern mit dem Befehl: `npm install crypto`.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Erzeugen der Verschlüsselungszertifikate

Die Übung soll die für HTTPS benötigte Verschlüsselung trainieren.

7.2 Praktiken FWA - HTTPS

Erzeugen der Verschlüsselungszertifikate

Starten Sie dazu ihre Linux-Shell. Einzelschritte dazu:

- Erstellen Sie mit `openssl`-Zertifikate. Schauen Sie sich die möglichen Optionen mit `openssl-h` an.
- Laden sie die erzeugten Dateien für die späteren Übungen auf die Ebene ihre C/S Anwendungsverzeichnis unter ihrem Home-Directory.

Für das Übungsbeispiel können Sie ein selbstsigniertes Zertifikat mit dem Werkzeug `openssl` wie folgt einmal erzeugen und dann für alle Übungen verwenden:

```
openssl genrsa -out localhost.key 2048
```

```
openssl req -new -x509 -key localhost.key -out  
localhost.cert -days 9999 -subj /CN=localhost
```

Die erstellten `localhost.key` und `localhost.cert` legen Sie in das C/S-Anwendungsverzeichnis ab.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

HTTPS-C/S-Anwendung mit Verschlüsselung

Die Übung soll die Verschlüsselung einer HTTPS-C/S-Anwendung trainieren.

7.3 Praktiken FWA - HTTPS-C/S-Anwendung mit Verschlüsselung

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_http26_gesicherte_kom` aus dem GitHub als Vorlage: `https://github.com/herbfrisch/csa/`
Dateiverzeichnis: `05_HTTP`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Konfiguration von Node.js Modul passport

Die Übungen soll die Konfiguration der Authentifizierung mit dem Node.js-Modul passport trainieren.

7.4 Praktiken FWA - „passport“

Konfiguration von Node.js Modul passport

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession01_konfigauth` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/>
Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Einbindung der passport „local“-Strategy

Die Übungen sollen den Umgang mit der „local“-Strategy des passport-Node.js-Moduls trainieren.

7.5 Praktiken FWA - „passport“ Einbindung der passport „local“-Strategy

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession02_localstrategy` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/>
Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Authentifizierung - Anmelden

Die Übungen sollen den Umgang mit Authentifizierung, dem Anmelden eines Benutzers an der C/S-Anwendung trainieren.

7.6 Praktiken FWA - „passport“

Authentifizierung - Anmelden

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession03_anmelden` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/>
Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Bauen sie den Code für die Benutzer der Open Alexa um.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Absichern von Ressourcen

Die Übungen sollen den Umgang mit dem Absichern von Ressourcen in einer C/S-Anwendung trainieren.

7.7 Praktiken FWA - „passport“ Absichern von Ressourcen

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession05_absichernressourcen` aus dem GitHub als **Vorlage**: `https://github.com/herbfrisch/csa/` Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Legen Sie die zu sichernden Ressourcen für Open Alexa fest und bauen sie den Code entsprechend um.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

User Model

Die Übungen sollen den Umgang mit der User Modellierung in einer C/S-Anwendung trainieren.

7.8 Praktiken FWA -

User Model

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession09_usermodel` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/> Dateiverzeichnis: `05_HTTP`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Bauen Sie den Code für Ihre Benutzer der Open Alexa um. Erstellen Sie dazu ein User Modell für folgende Anforderungen:
 - Minderjährige (0-6 Jahre) dürfen nicht geschäftsfähig sein.
 - Beschränkt geschäftsfähige Minderjährige (7-17)
 - dürfen für gewöhnliche regelmäßige Geschäftsabschlüsse tätigen.
 - Für alle anderen Geschäftsabschlüsse benötigen sie die Zustimmung der Eltern.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Benutzerauthentifizierung gegen eine Datenbank

Die Übungen sollen den Umgang mit der Benutzer-Authentifizierung einer C/S-Anwendung gegen eine Datenbank Session-Handling trainieren.

7.9 Praktiken FWA - Benutzerauthentifizierung gegen eine Datenbank

Zunächst müssen Sie die SQLite Datenbank um eine Benutzertabelle erweitern und zumindest einen Datensatz eingeben. Passen Sie die Butzer für Ihre Open Alexa Anwendung um. Ergänzen Sie das Geburtsdatum:

```
CREATE TABLE `Users` (  
  
  id INTEGER PRIMARY KEY,  
  
  firstname TEXT,  
  
  lastname TEXT,  
  
  username TEXT,  
  
  password TEXT);  
  
INSERT INTO `Users`(`firstname`, `lastname`, `username`, `password`) VALUES  
('Hans', 'Mayer', 'hmayer', '6f4b726238e4edac373d1264dcb6f890');
```

7.9 Praktiken FWA - Benutzerauthentifizierung gegen eine Datenbank

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession10_authdatenbank` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/>
Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Bauen Sie den Code entsprechend Ihrer Open Alexa Anwendung um.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Zugriffsbeschränkungen

Die Übungen sollen den Umgang mit Zugriffsbeschränkungen auf eine C/S-Anwendung trainieren.

7.9 Praktiken FWA - Benutzerauthentifizierung gegen eine Datenbank

Zunächst müssen Sie Ihre Movietabelle Ihrer SQLite Datenbank um ein zusätzliches Feld erweitern, Ändern Sie die Datenfelder entsprechend Ihren Anforderungen der Open Alexa Anwendungen:

```
ALTER TABLE `Movies` ADD COLUMN `user` INTEGER;
```

```
ALTER TABLE `Movies` ADD COLUMN `public` INTEGER;
```

7.10 Praktiken FWA - Zugriffsbeschränkungen

Starten Sie dazu ihre node.js- Anwendung und verwenden Sie einen Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_authsession11_zugriffsbeschraenkung` aus dem GitHub als **Vorlage**: `https://github.com/herbfrisch/csa/` Dateiverzeichnis: `09_Authentifizierung_und_Sessionhandling`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Ändern Sie die Zugriffsbeschränkungen für Ihre Open Alexa Anwendung.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Praktiken Fullstack Web-Anwendungen – Authentifizierung, Session-Handling, Sicherheitsaspekte

Authentifizierung, Session-Handling, Sicherheitsaspekte

Die Übungen sollen den Umgang mit Authentifizierung, Session-Handling, Sicherheitsaspekte trainieren.

7.11 Praktiken FWA - „Adressbuch“

Authentifizierung, Session-Handling, Sicherheitsaspekte

Starten Sie und dazu ihre node.js- Anwendung und verwenden Sie eine Browser als Client! Einzelschritte dazu:

- Holen Sie sich die Dateien zu `app_http08_indexjs_einstieg` aus dem GitHub als Vorlage: <https://github.com/herbfrisch/csa/>
Dateiverzeichnis: `05_HTTP`
- Laden sie die Dateien in ein eigenes neues Programmverzeichnis unter ihrem Home-Directory.
- Inspizieren Sie den Code mit dem `nano`-Terminaleditor oder Sublime Text.
- Starten Sie die Anwendung im Terminalfenster.
- Starten sie ihren Browser mit dem Link auf `http://localhost:8080` .
- Stoppen Sie die Anwendung mit `<ctrl>-c` im Terminalfenster.

Ende der Praktiken

Vielen Dank für Ihre Aufmerksamkeit!

Link Verzeichnis

- 1 Programmvorlagen: <https://github.com/herbfrisch/csa>
 - 2 Dokumentation zur den Kernmodulen: <https://nodejs.org/api/>
-
-
-
-
-
-
-
-
-
-