

Web Engineering 3 - Script

Ablauf & Planung

- Ziel des Moduls: Projektarbeit mit einem Beleg und einer Präsentation als finales Ziel
- Projekt sollte einen Großteil der Aspekte der Web Entwicklung abdecken
- Basis für das Projekt: User-Stories, die das Projekt leiten sollen
- Gruppenarbeit möglich
- Beleg mit Code als Abgabe am Ende des Semesters

Anforderungen

Projekt

- Datenbank, Backend, Frontend mit Container Deployment (Docker, Podman, ...)
- Dokumentation der REST API Endpunkte mit OpenAPI o.ä.
- Einige Tests in Front- und Backend. Komplettes Test coverage wird nicht vorausgesetzt
- Einreichung des Repositories (ZIP, Link zu GitHub oder andere VCS)

Beleg

- Seitenanzahl nicht festgelegt. Bewegt sich wahrscheinlich um 20 Seiten, wird aber nicht vorausgesetzt
- Beschreibung, wie Anforderungen aus den User Stories umgesetzt wurden
- Umsetzung beschreiben
- Gründe für Entscheidungen bei der Entwicklung darstellen
- Dokumentation der einzelnen Software Bestandteile

Präsentation bzw. Verteidigung

- Demonstration des finalen Produkts
- Vorstellung der Umsetzung
- Kurzes Zeigen von ausgewählten Programmbestandteilen, die als wichtig angesehen werden

Technologien

Frontend

- Svelte/SvelteKit
- Vue
- React
- Angular
- Astro
- Vanilla
- Rails
- Laravell
- Symphony
- Vite/Webpack
- NodeJS/Bun/Deno/Yarn
- TailwindCSS
- PRIME

Backend

- Tomcat Servlet zu Spring Beans (Aufbau)

- Spring
- Micronaut
- Quarkus
- Kotlin
- Gradle

Database

- PostgreSQL

Testing

- Cypress, Playwright

DevOps

- Docker
- Podman

User Stories

Kernpunkte einer User Story [1]:

- Wer ist der User
- Was will der User machen
- Warum will der User das machen
- Weitere Informationen sind optional

Template [1]:

AS A {user|persona|system}

INSTEAD OF {current condition}

I WANT TO {action} IN {mode} TIME | IN {differentiating performance units} TO {utility performance units}

SO THAT {value of justification}

NO LATER THAN {best by date}

User Stories für das finale Projekt

Debugging

- Lesen von Dokumentation und Stack Traces
- Intelligentes Nutzen von Google
- Logging: Print von unterschiedlichen Daten um Funktionalität sicherzustellen
- Debuggers in IDEs: Pausieren der Ausführung, Durchschreiten von Codestücken
- Reproduzieren des Problems in einer minimalen Umgebung, die im schlimmsten Fall auch mit anderen geteilt werden kann
- Test Driven Development als Strategie Probleme früher zu finden
- Nutzen von Sprachen mit expliziten Types (z.B.: TypeScript statt JavaScript)

Frontend

Browser DevTools

[Firefox DevTools User Docs](#)

HTML/CSS Inspection

- Live HTML und CSS anpassen um Änderungen zu sehen
- Visuelle Darstellungen von Parametern wie Padding, Margins usw.

JavaScript Console

- Log output lesen
- Interagieren mit der Website durch JavaScript

JavaScript Debugger

- Durchschreiten von JavaScript auf der Website

Source Map

- Wichtig für minified JavaScript (zum Beispiel erstellt durch Frameworks)
- Originaler Code bleibt erhalten und kann für das Debugging genutzt werden
- Source Map muss generiert werden
- In der transformierten Datei muss mit einem Kommentar auf die Source Map verwiesen werden

```
1 ///# sourceMappingURL=http://example.com/path/sourcemap.map
```

JS JavaScript

Generierung:

- **Svelte** generiert SourceMaps automatisch. Früher wurde eine Compiler Option benötigt.

```
1 // svelte.config.js
2 const config = {
3   compilerOptions: {
4     enableSourcemap: true
5   },
6 }
```

JS JavaScript

React und **Vue** generieren SourceMaps automatisch. Sie können durch Compiler Option explizit aktiviert oder deaktiviert werden

```
1 /* tsconfig.node.json */
2 {
3   "compilerOptions": {
4     "sourceMap": true
5   }
6 }
```

JSON

Angular generiert SourceMaps automatisch. Es wird durch eine Compiler Option aktiviert.

```
1 /* angular.json */
2 "projects": {
3   "vite-project": {
4     "architect": {
5       "build": {
6         "configuration": {
7           "development": {
8             "sourceMap": true
9           }
10        }
11      }
12    }
13 }
```

JSON

Network Operations

- Auflistung aller Netzwerk Anfragen der Website
- Genaue Untersuchung aller Daten, die zu den einzelnen Anfragen gehören

IDE Debugging**Debugging Extensions****Bibliography**

- [1] J. Schiel, “The Anatomy of a User Story.” ScrumAlliance.