



Fiori App Entwicklung

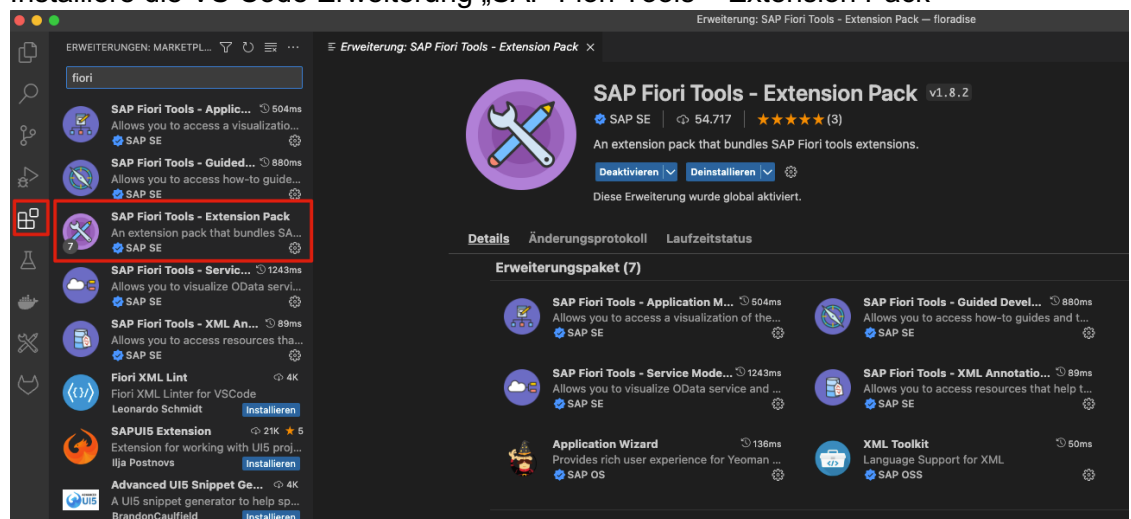
Aufgabe 7 (40 Punkte):

In dieser Aufgabe soll es darum gehen, den ersten Teil des *FitReality*-Tools zu entwickeln.

Entwicklungsumgebung einrichten

1. Richte Node.js ein
 - a. Lade node.js herunter und installiere es auf deinem Rechner:
<https://nodejs.org/>

Tipp: Verwende die neueste (LTS) Version. Zum Zeitpunkt der Erstellung dieser Aufgabe ist das v22.13.0.
2. Richte Visual Studio Code ein
 - a. Lade Visual Studio Code herunter und installiere es auf deinem Rechner:
<https://code.visualstudio.com/download>
 - b. Installiere die VS Code Erweiterung „SAP Fiori Tools – Extension Pack“



3. Richte Git ein. Folgt den Installationsanweisungen auf <https://git-scm.com/>
4. Klont das „fit-reality-template“ Projekt <https://git.octavia.de/uni-kassel/bpe-ws-24-25/fit-reality-template.git> . Dieses Projekt enthält das FitReality Backend, welches die Daten für das Frontend liefert.



FitReality-Tool (Frontend) entwickeln

1. Generiere das UI5 Projekt mit Hilfe von Yeoman wie in der Vorlesung gezeigt. Verwende den Generator „easy-ui5“. Stelle dazu sicher, dass der Generator global installiert ist:

```
npm install -g yo generator-easy-ui5
```

Wichtig: Als generator muss „ts-app“ gewählt werden!

2. Implementiert das Frontend so weit, wie in der Vorlesung gezeigt:
 - a. Name
 - b. Größe in cm
 - c. Gewicht in kg
 - d. Trainingsziel
 - e. Geschlecht (Männlich, Weiblich, Divers)

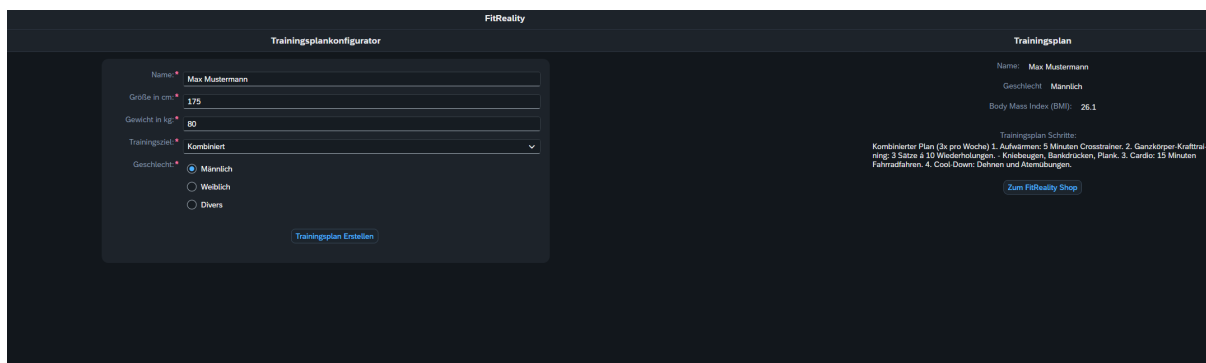
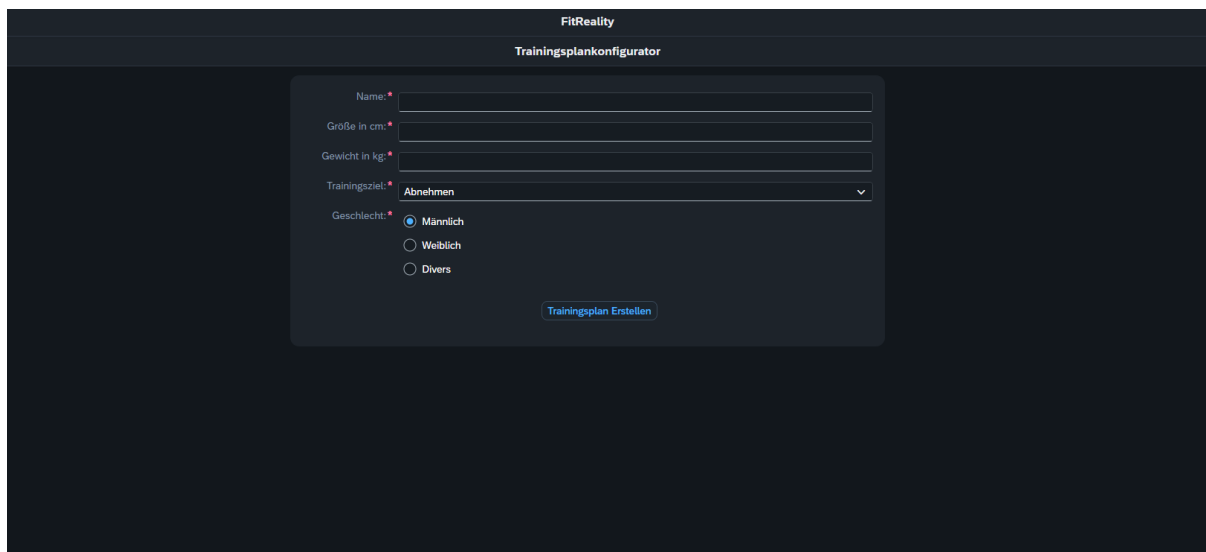
Beachtet:

- **Alle Felder sollen Pflichtfelder sein**
- **Die Felder Größe und Gewicht sollen nur Zahlen akzeptieren**

3. Fügt folgende Funktionalität hinzu:
 - a. Button am Ende des Formulars „*Trainingsplan Erstellen*“
 - b. Beim Klick auf dem Button „*Trainingsplan Erstellen*“ erscheint auf der rechten Seite der Trainingsplan:
 - i. Name
 - ii. Geschlecht
 - iii. Body Mass Index (BMI). Der BMI soll sich wie folgt berechnen:
$$\text{bmi} = \text{Gewicht} / \text{Größe}^2$$
 - iv. Trainingsablauf des ausgewählten Trainingsziels. Der Trainingsablauf ist der Inhalt des Properties description des jeweiligen Trainingsplans.
 - v. Button „*Zum FitReality Shop*“

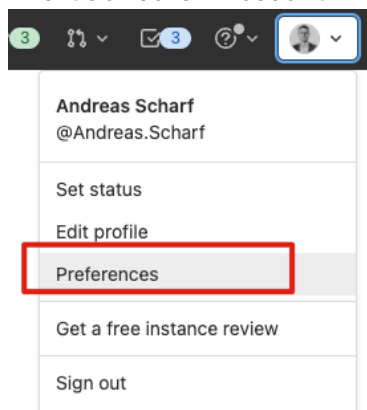


Am Ende sollte das Frontend ähnlich zu folgenden Screenshots aussehen:



FitReality Tool in Git einchecken

1. Loggt euch in dem euch zugewiesenen Git Repository auf <https://git.octavia.de> mit einem Webbrowser ein.
2. Generiert ein Access Token
 - a. Klickt auf euren Account → Preferences:





- b. Klickt auf „Access Tokens“. Wählt einen beliebigen „Token name“ und wählt als Scope „write_repository“

The screenshot shows the GitLab user interface for creating a personal access token. On the left sidebar, the 'Access Tokens' option is highlighted with a red box. The main content area is titled 'Personal Access Tokens' and includes instructions on how to use them. A form is provided to create a new token, with the following fields and options:

- Token name:** A text input field containing 'bpe', highlighted with a red box.
- Expiration date:** A date picker field set to 'YYYY-MM-DD'.
- Select scopes:** A list of checkboxes for different permission levels:
 - api
 - read_api
 - read_user
 - read_repository
 - write_repository (highlighted with a red box)
 - read_registry

- c. Klickt auf „Create personal access token“

WICHTIG: Speichert dieses Token an einem sicheren Ort. Wenn die Seite geschlossen wird, gibt es keine Möglichkeit mehr dieses Token anzuzeigen.



3. Folgt den Anweisungen für
 - a. Git local setup
 - b. Push an existing Git repository.

Beispiel:

Command line instructions

You can also upload existing files from your computer using the instructions below.

Configure your Git identity

Get started [with Git](#) and learn [how to configure it](#).

Local Global

Git local setup

Configure your Git identity locally to use it only for this project:

```
git config --local user.name "Andreas Scharf"
git config --local user.email "andreas.scharf@octavia.de"
```

Add files

Create a new repository

```
git clone https://git.octavia.de/uni-kassel/bpe-ws-24-25/ .git
cd .
git switch --create main
touch README.md
git add README.md
git commit -m "add README"
git push --set-upstream origin main
```

Push an existing folder

```
cd existing_folder
git init --initial-branch=main
git remote add origin https://git.octavia.de/uni-kassel/bpe-ws-24-25/ .git
git add .
git commit -m "Initial commit"
git push --set-upstream origin main
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin https://git.octavia.de/uni-kassel/bpe-ws-24-25/ .git
git push --set-upstream origin --all
git push --set-upstream origin --tags
```



4. Fügt den Tag „aufgabe-7“ zu dem Abgabecommit hinzu und pushed diesen:

- a. `> git tag aufgabe-7`
- b. `> git push origin --tags`
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To <https://git.octavia.de/uni-kassel/bpe-ws-24-25/<matrikelnummer>.git>
* [new tag] aufgabe-7 -> aufgabe-7

Wichtig: Wir kontrollieren diesen Tag. Falls er falsch gesetzt ist (z.B. weil ihr danach noch Commits gemacht habt), werden die Änderungen nicht berücksichtigt. Falls der Tag nicht gesetzt ist, gibt es Punktabzug und wir kontrollieren HEAD.

Stellt sicher, dass sich die Anwendung korrekt starten lässt und die Daten wie in der Vorlesung gezeigt darstellt.

Deadline:

Abgabe: Donnerstag, 23.01.2025, 14:00 Uhr