
GUI Vorlesung 2019

Übung 10 – Web GUIs Teil 2

Beschreibung

Wir verwenden das HTML der letzten Übung um eine kleine lauffähige Web-Version des Headers zu erzeugen. Als Framework verwenden wir dazu Angular.

Aufgabe 1

Öffnen und starten Sie das Starter-Projekt.

Schritte:

1. Öffnen Sie das Projekt *u10-web2-starter* in der IDE
2. Öffnen Sie eine Konsole im Ordner des Projekts
3. Starten sie den Webserver mit
`npm install`
dann
`npm run ng serve`
4. Starten sie einen Browser und öffnen sie die Anwendung (die URL steht in der Konsole, normalerweise localhost:4200)
5. Ändern Sie den Text in der Datei *index.html*
6. Speichern Sie und beobachten Sie das Resultat im Browser

Aufgabe 2

Erzeugen Sie eine neue Komponente für den Header.

Schritte:

1. Erzeugen Sie die folgenden drei Dateien im Ordner *src/app*
 - a. *header.component.css*
 - b. *header.component.html*
 - c. *header.component.ts*
2. Kopieren Sie die Styles der letzten Übung (*web1.css*) in die neue CSS Datei.
3. Kopieren Sie die HTML-Inhalte des Headers (*web1.html*) in die neue HTML-Datei.
Hinweis: Nur das benötigte `<div>` mit Inhalt, nicht den Rahmen der HTML Datei (kein `<head>`, `<body>`, `<script>`....).
4. Füllen Sie die Datei *header.component.ts* mit einer passenden leeren Komponente. Als Vorlage können Sie dazu die Datei *app.component.ts* benutzen, sollten aber Namen, Pfade und Selektor anpassen.
5. Sie müssen jetzt noch Ihre Komponente registrieren. Machen Sie das, indem Sie den Header analog zur *AppComponent* in der Datei *app.module.ts* aufnehmen.

Hinweis: Import nicht vergessen und die einzelnen Komponenten kommasepariert in *declarations* aufnehmen.

6. Verwenden Sie den Header in der *AppComponent*.

Aufgabe 3

Speichern Sie die Eingabe des Benutzers in einer Modelklasse und binden Sie einen Click-Handler an.

Schritte:

1. Verwenden Sie die Klasse *HeaderModel* im Code der *HeaderComponent* (*import* nicht vergessen)
2. Erzeugen Sie eine neue Instanz der Klasse
3. Binden Sie mind. 3 der Eingabefelder (Text) an die Attribute des Models
4. Geben Sie bei einem Click auf den „Generate Graph“-Button das Model aus

Beispiel:

```
import {HeaderModel} from "../HeaderModel";

export class HeaderComponent {
  ...

  public onGenerateGraph(): void {
    alert(this.model)
  }
}
```

Aufgabe 4 - optional

Machen Sie aus dem Header eine wiederverwendbare Komponente. Verschicken sie beim Click auf den *onGenerateGraph*-Button ein Event an die Elternkomponente und behandeln Sie das Event dort, statt direkt im Header.

Schritte:

1. Importieren Sie in der *HeaderComponent* die benötigten Angular-Elemente:
import { Component, EventEmitter, Output } from '@angular/core';
2. Führen Sie eine *@Output()* Property vom Typ *EventEmitter<HeaderModel>* ein.
3. Erzeugen Sie eine Instanz.
4. Bauen sie die bisherige *onGenerateGraph* Methode so um, dass sie selbst nichts mehr ausgibt, sondern ein Event verschickt.
5. Führen sie in der *AppComponent* eine Methode ein die etwas ausgibt.
6. Passen Sie das Template der *AppComponent* so an, dass bei Events vom Header die neue Methode in der *AppComponent* ausgeführt wird