

Web-Technologien, Sommersemester 2024

Teilleistung 2

Abgabe der Lösungen

- *Abgabetermin.* **26. Mai 2024, 23:55 Uhr**
- *Abgabemodus.* Online-Einreichung des aus Git exportierten Teilleistungs-Ordners als ZIP-Datei im Virtuellen Campus im entsprechenden VC-Kurs des Moduls. Bei einer Bearbeitung im Team, genügt die Abgabe durch ein einzelnes Teammitglied. Die zuletzt hochgeladene Abgabe wird bewertet.
- *Dateiformat der Abgabe.* Die Abgabe erfolgt durch den Upload des ZIP-Archivs, das aus dem bereitgestellten Git-Repository exportiert werden soll. Das Archiv soll alle notwendigen Dateien beinhalten. Verändert die Benennung des Archivs, die beim Export aus Git vergeben wird, nicht.
- *Update der Lösung.* Bis zur oben angegebenen Deadline könnt Ihr Eure Lösung beliebig oft durch eine neue (korrigierte) Fassung ersetzen. Zuvor hochgeladene Lösung werden dabei überschrieben. Wir haben keine Möglichkeit, alte Fassungen wiederherzustellen!

Hinweise

- Die vollständige und korrekte Bearbeitung einer Aufgabe ergibt die volle Punktzahl dieser Aufgabe.
- Für diese Teilleistung erwarten wir einen Bearbeitungsaufwand von insgesamt etwa 45 Stunden, da wir von vier Personen pro Team ausgehen. Dies ist als Richtwert zu verstehen und keinesfalls als Unter- oder Obergrenze.
- In Verdachtsfällen behalten wir uns vor, Plagiate von der Bewertung auszunehmen. Wir werden in einem solchen Fall Kontakt mit den Betroffenen aufnehmen.
- Programmieraufgaben sollen in nachvollziehbar kommentierter Form im Quelltext abgegeben werden. Dabei dürfen keine anderen als die angegebenen Technologien verwendet werden. Nicht lauffähige Programme können nicht bewertet werden. Der Abgabe soll eine README-Datei hinzugefügt werden, die beschreibt, wie man die Lösung ausführen kann.
- Für die Bearbeitung der Teilleistung wird das Versionsverwaltungstool Git verwendet. Der gute Umgang mit Git fließt in die Bewertung der Teilleistung ein.
- Bei der Auswahl der Medien sind keine Urheberrechte zu berücksichtigen, da eine Veröffentlichung der Teilleistung nicht vorgesehen ist.

Aufgabe: Erstellung einer Website für eine Roboterliga mittels HTML, CSS und JavaScript (30 Punkte)

Nachdem es sehr viele gute Vorschläge für das Uni.fest gab, wurde entschlossen einen davon in größerem Stil umzusetzen. Es soll einen Wettkampf geben, bei dem Roboter gegeneinander kämpfen, die von Teams entwickelt und gebaut werden. Dieses Event wird über den Rahmen des Uni.fest hinausgehen, und soll daher einen eigenen, unabhängigen Webauftritt erhalten.

Die neue Webanwendung soll eine **Single-Page-Application** sein, d. h. nur eine einzige HTML-Datei umfassen, auf der alle Informationen zum Wettbewerb zu finden sein sollen. Sie soll neben allgemeinen Informationen zur Liga einen Zeitplan und eine Übersicht über die antretenden Roboter geben. Die Daten zu den Robotern und den Teams wurden im Vorfeld online gesammelt und können über eine API abgerufen werden.

Zudem wurde bereits ein Ablaufplan der Kämpfe erstellt und in einer JSON-Datei gespeichert. Diese ist in der bereitgestellten Vorlage (*template-tl2.zip*), welche für die Bearbeitung verwendet werden muss, enthalten.

Abbildung 1 zeigt die Struktur der Single-Page-Application. Die weiteren Anforderungen an Inhalt, Design, Layout und Funktionalität werden im Folgenden aufgelistet.

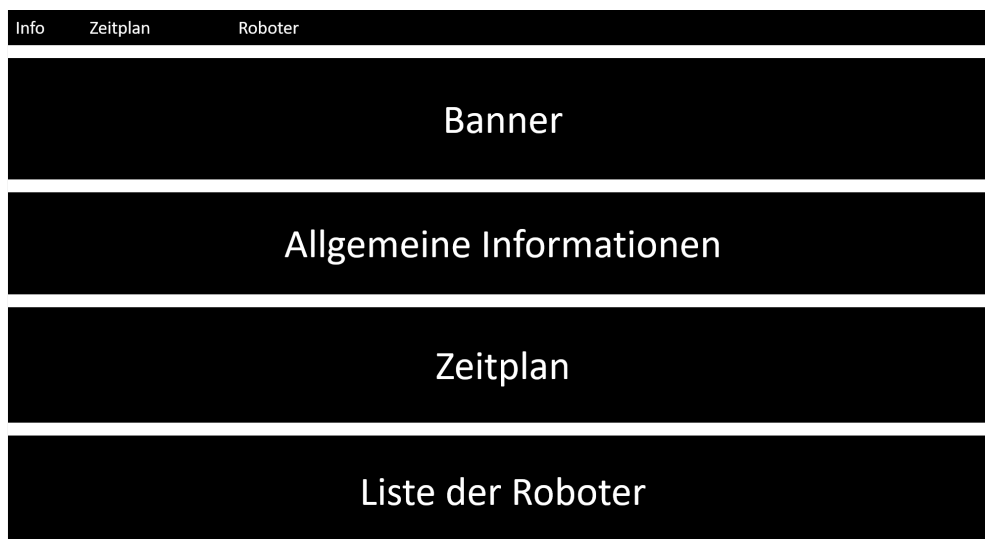


Abbildung 1: Struktur der Single-Page-Application

Zugriff auf die API

Auf die API kann mithilfe von GET-Requests zugegriffen werden. Die Informationen zu den Robotern sind unter <https://robots-api.funkhaus-franken.de> verfügbar.

Um sich einen Überblick über die Informationen der API zu verschaffen, kann es hilfreich sein, die API-Antwort zunächst in Firefox oder auch mit Tools wie Postman¹

¹<https://www.postman.com>

oder *Insomnia*² anzusehen. Diese Tools ermöglichen es, die Antwort strukturiert darzustellen und erleichtern Euch so das Verschaffen eines Überblicks.

Weitere Anforderungen

- Oben auf der Seite soll ein Banner mit einer Überschrift und einem kurzen Slogan zu sehen sein. Denkt euch gerne einen kreativen Namen für die Roboterliga aus. Unterhalb davon sollen die drei Abschnitte (Informationen, Zeitplan, Roboter) folgen.
- Es soll eine Navigationsleiste geben, über die man mithilfe von HTML-anchors zu den drei Abschnitten der Website springen kann. Diese Leiste soll immer am oberen Bildschirmrand angezeigt werden, egal wo man sich auf der Webseite aufhält.
- Der erste Abschnitt soll allgemeine Informationen bzw. Werbung für das Event enthalten. Hier könnt ihr kreativ werden, um die Zuschauer bestmöglich anzulocken und zu informieren.
- Der nächste Abschnitt soll den Zeitplan der Kämpfe an den jeweiligen Tagen darstellen. Hierfür sollen die Daten aus der JSON-Datei in der bereitgestellten Vorlage (*template-tl2.zip*) verwendet werden. Die Vorlage enthält eine HTML-Datei und eine JS-Datei, welche bereits korrekt in die HTML-Datei eingebunden ist. Die JS-Datei enthält zudem bereits eine Funktion, die beim Aufruf der Seite einen Eintrag aus der JSON-Datei einliest und auf der Konsole ausgibt.

- Die **Kampfzeit** ist im *ISO-8601*-Format³ angeben und kann mittels des `Date()`⁴ Konstruktors von JavaScript verarbeitet werden. Dazu hier ein Beispiel:

```
const date = new Date("2024-07-05T12:00");
const formattedDate = date.toLocaleDateString("de-DE", {
  minute: "numeric",
  hour: "numeric",
  day: "numeric",
  month: "long",
  year: "numeric",
});
console.log(formattedDate) // 5. Juli 2024 um 12:00
```

- Ein Mockup für den Zeitplan ist in Abbildung 2 dargestellt.
- Der letzte Abschnitt soll die antretenden Roboter in einer Liste vorstellen. Alle von der API zur Verfügung gestellten Daten sollen dazu mithilfe der Fetch API abgerufen und im Webaufttritt angezeigt werden. Dazu gehört neben Namen und Eigenschaften auch das Bild, das in einer angemessenen Größe dargestellt sein soll. Auch hier wurde als Orientierung ein grobes Mockup erstellt³.
- Zudem soll es die Möglichkeit geben die Liste nach Name und Stärke der Roboter zu ordnen. Dabei soll man zwischen aufsteigender und absteigender Sortierung wählen können.

²<https://insomnia.rest>

³https://de.wikipedia.org/wiki/ISO_8601

⁴https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/Date

Tag 1			Tag 2		
Roboter 1	Roboter 2	Uhrzeit	Roboter 1	Roboter 2	Uhrzeit

Abbildung 2: Mockup des Zeitplan-Abschnitts

- Für den Fall, dass die API nicht erreichbar ist, soll eine sinnvolle und verständliche Fehlermeldung an passender Stelle angezeigt werden.
- Während Daten von der API abgefragt werden, soll eine Ladeanzeige dargestellt werden. Die Ladeanzeige soll ausschließlich mit CSS implementiert werden. Ein paar Beispiele sind unter <https://codesandbox.io/s/luoos?file=/loaders/circular-trails/circular-trails.css> zu finden.
- Der Webaufttritt soll ein klares, übersichtliches und einheitliches Design besitzen. Buttons sollen also bspw. leicht als solche erkennbar sein.
- Die Single-Page-Application soll ein klares, übersichtliches und einheitliches Layout besitzen. Inhalte sollen also bspw. übersichtlich angeordnet werden.



Name: Robot2-Fight2
 Teamname: Star Fights
 Klasse: Feuer
 Gewicht: 42kg
 .
 .
 .

Sortierung:

- Name (asc)
- Name (asc)**
- Name (desc)
- Stärke (asc)
- Stärke (desc)



Name: Zermet the Killer
 Teamname: Cereal Street Squad
 Klasse: Elektrizität
 Gewicht: 20kg
 .
 .
 .

Abbildung 3: Mockup der Roboter-Liste. Obacht: Im Mockup werden noch nicht alle verfügbaren Daten dargestellt. Die fertige Single-Page-Application soll jedoch alle Daten anzeigen.

**Wir wünschen Euch viel Erfolg
bei der Bearbeitung der Teilleistung!**