



## Fiori App Entwicklung II

### Aufgabe 8 (20 Punkte):

In dieser Aufgabe soll der in Aufgabe 7 erstellte Kleine-Helden Konfigurator weiterentwickelt werden. Hierzu soll auf dem Code aus Aufgabe 7 aufgebaut werden. Ist die Aufgabe abgeschlossen, soll der Entwicklungsstand entsprechend getagged und in das gleiche Repository wie in Aufgabe 7 gepushed werden.

#### Kleine-Helden Konfigurator entwickeln – Teil II

In dieser Aufgabe soll das in Aufgabe 7 entwickelte Frontend an ein bereitgestelltes Backend angebunden werden. Das Backend ist erreichbar unter der URL <https://kleine-helden.octavia.de> und stellt folgende Informationen zur Verfügung:

- <https://kleine-helden.octavia.de/odata/v4/kleine-helden/BookSizes>: Verfügbare Buchgrößen inkl. Preis.
- <https://kleine-helden.octavia.de/odata/v4/kleine-helden/StoryTypes>: Verfügbare Geschichtstypen

1. Implementiert den Kleine-Helden Konfigurator so weit, wie in der Vorlesung gezeigt
  - a. Die verfügbaren Buchgrößen sollen vom Server geladen werden und inkl. Preis angezeigt werden:

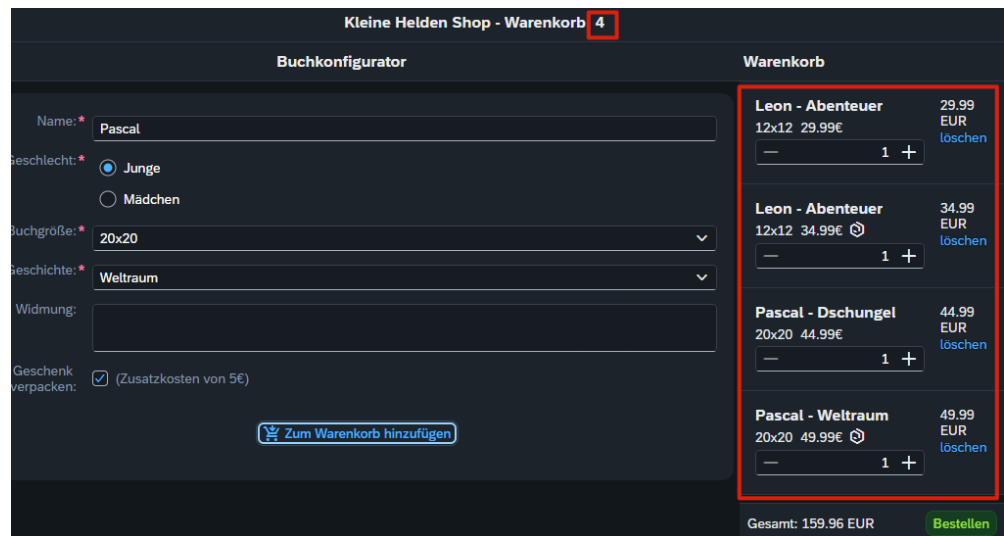
Buchgröße: *	20x20	
Geschichte: *	12x12	29.99€
Widmung:	20x20	44.99€

- b. Die verfügbaren Geschichten sollen vom Server geladen werden und inkl. Preis angezeigt werden:

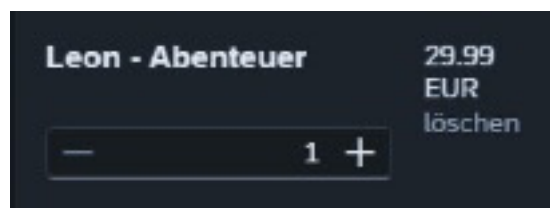
Geschichte: *	Weltraum
Widmung:	Abenteuer Dschungel Unterwasser
Als Geschenk verpacken:	Weltraum

[Zum Warenkorb hinzufügen](#)

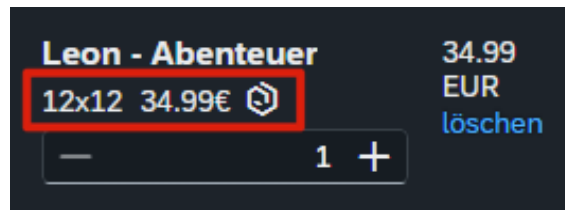
- c. Beim Klicken auf den „Zum Warenkorb hinzufügen“ Button soll:
    - i. Die aktuelle Konfiguration dem Warenkorb hinzugefügt werden
    - ii. Automatisch der Warenkorb auf der rechten Seite geöffnet werden
    - iii. Der Text „Kleine Helden Shop – Warenkorb: <CartCount>“ die aktuelle Anzahl der im Warenkorb befindlichen Elemente anzeigen.



- d. Warenkorb implementieren
- i. Der Warenkorb zeigt alle im Warenkorb befindlichen Elemente in einer Liste an
  - ii. Für jedes Element wird folgendes angezeigt:
    1. <Name> - <Geschichte>
    2. Ein UI Element welches es erlaubt die Anzahl der Elemente im Bereich von 1-9 zu ändern.
    3. Einen „Löschen“ Button um das jeweilige Element aus dem Warenkorb zu entfernen.
    4. Der Gesamtpreis des jeweiligen Elements am rechten Rand.
    5. Die Gesamtsumme des Warenkorbs ganz unten im Warenkorb (links neben dem „Bestellen“ Button) angezeigt werden. Die Berechnung soll im Frontend stattfinden. Werden Artikel dem Warenkorb hinzugefügt oder gelöscht, soll sich die Summe automatisch korrekt aktualisieren.



- e. Warenkorb – Detailinformationen
- i. Für jedes Element wird folgendes angezeigt:
    1. Die Buchgröße
    2. Der Produktpreis
    3. Falls bei dem Produkt eine Geschenkverpackung gewählt wurde, soll das icon "sap-icon://product" angezeigt werden. Sonst soll kein Icon angezeigt werden.



- f. Bestellen Funktion implementieren
  - i. Im Warenkorb soll ein „Bestellen“ Button angezeigt werden (siehe Screenshots oben)
  - ii. Beim Klicken auf den „Bestellen“ Button soll eine Bestellung an den Server gesendet werden. Hierzu soll die „unbound action“ `submitOrder(artikel)` aufgerufen werden. Abhängig von der Antwort des Servers eine passende Meldung angezeigt werden:

204: Bestellung erfolgreich

500: Entschuldigung, es ist ein Fehler bei der Bestellung aufgetreten

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

### Kleine-Helden Konfigurator in Git einchecken

1. Fügt den Tag „aufgabe-8-1“ zu dem Abgabecommit hinzu und pushed diesen:
  - a. `> git tag aufgabe-8-1`
  - 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-23-24/<matrikelnummer>.git  
 \* [new tag]                    aufgabe-8-1 -> aufgabe-8-1

**Wichtig 1: 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.**

**Wichtig 2: Verwendet für die zweite Abgabe bis Donnerstag den Tag „aufgabe-8-2“. Solltet ihre keine Änderungen vorgenommen haben, setzt den Tag trotzdem einfach auf den gleichen Commit wie den Tag „aufgabe-8-1“!**

Abgabeformat: Checkt eure Abgaben im OctaVIA GIT ein

### Deadline:

Montag, 18.12.2023, 14:00 Uhr

Donnerstag, 21.12.2023, 14:00 Uhr