

Dynamisches Web - Übung 4

25. November 2017

Inhalt der Übung

- Besprechung Übungsaufgabe
- Gemeinsames Refactoring der bestehenden Lösung
- Teamaufgaben
 - Auslagerung Handlebars Templates
 - Umsetzung der fehlenden Features
 - Fortsetzung Refactoring
- Präsentation Teamaufgaben

Übungsaufgabe

Abgabeschluss: **17. Dezember 2017, 22:00 Uhr**

Generelle Vorgaben:

Bitte das JavaScript-File in Moodle mit dem Dateinamen-Muster `ue4-<nachname1>-<nachname2>.zip` hochladen. Weitere Vorgaben:

- Als Basis dient die Serverumsetzung, die als ZIP-Datei auf der E-Learning-Plattform zum Download bereit steht.
- Die Übungsaufgabe ist wieder eine Gruppenaufgabe zu 2 Personen (3 bei ungerader Anzahl der Gesamtgruppenmitglieder).
- Bitte arbeiten Sie wieder in der gleichen Gruppenkonstellation wie in den bisherigen Übungen.
- Im abzugebenden ZIP-File bitte NICHT das `node_modules` Verzeichnis inkludieren, sondern stattdessen `package.json`, das die entsprechenden `dependencies` Deklaration beinhaltet.
- Die Abgabe auf der E-Learning-Plattform erfolgt nur durch EIN Gruppenmitglied
- Alle Nachnamen der 2er-Gruppe (oder ggf. 3er-Gruppe) müssen im Namen des abgegebenen ZIP-Files enthalten sein, sonst können die Bewertungspunkte nicht zugeordnet werden

- Alle zu verwendeten zusätzlichen Bibliotheken sind im package.json File des bereitgestellten Übungsservers deklariert und können mit `npm install` installiert werden.

Aufgabenstellung:

Diesmal schlüpfen Sie in die Rolle eines Webentwicklerteams, das in einem kleinen Software-Haus beschäftigt ist. Ihr Chef kommt mit einer neuen Projektanfrage zu Ihnen. Der Dorf-Fotograf Lutz Maier hat Ihren Chef gebeten, die Entwicklung seines Webshops zu übernehmen.

Wie so oft, können Sie leider nicht auf der „grünen Wiese“ starten. Der Fotograf hat nämlich ursprünglich den Cousin der Bruders eines Bekannten, der sich „ein bisschen auskennt“, mit der Entwicklung betraut. Das ganze Vorhaben hat jedoch kein gutes Ende genommen: „Ich war erstaunt, wie schnell was zu sehen war“, meint der Fotograf und setzt fort: „aber recht bald wurde es immer zäher und jetzt hebt er nicht einmal mehr ab, wenn ich ihn anrufe“.

„Ich wisse nicht was ich will“, berichtet der geläuterte Fotograf von den Vorwürfen des nun nicht mehr greifbaren Entwicklers. Diesmal will er es besser machen und schreibt alles, was es braucht, um mit dem Online-Shop zu starten, in einem E-Mail zusammen. Ihr Chef leitet Ihnen die Nachricht weiter und ergänzt diese um weitere Punkte:

Hallo!

Wie besprochen hier die Anforderungen von Herrn Maier ergänzt um meine Punkte:

1. Refactoring (5P)

- o Die jetzige Lösung ist in einer Wurst hinuntergeschrieben. Wenn wir so weiterwursteln, dann wird das nichts. Bitte bringt den Code in eine saubere Struktur. Aufgrund der eher geringen Größe der Anwendung, eignet sich dafür das MVC-Pattern ganz gut denke ich.
- o Lobenswert ist die simple Lösung mit Handlebars. Das einzige, was mich daran stört sind die noch nicht in eigene Dateien ausgelagerten Templates. Macht das doch bitte gleich am Anfang. Da ohnehin schon express.js verwendet wird, werft mal einen Blick in diese Anleitung <http://expressjs.com/en/advanced/developing-template-engines.html>.
- o Herr Maier hat mir gegenüber auch erwähnt, dass er „Großes“ mit dem Webshop vor hat. Er denkt darüber nach, das Ganze auch als mobile App anzubieten. Deshalb achtet beim Umstrukturieren stets darauf, dass die Models nicht davon abhängig sind, innerhalb einer Weblösung zu laufen. Wenn ein Model

beispielsweise ein express.js Request-Objekt als Parameter verlangt, dann habt ihr etwas falsch gemacht. Beim Extrahieren der Models stellt euch einfach vor, ob ihr die Models mit Unit Tests testen könnt, ohne dafür den Webserver zu starten.

Jetzt kommen die restlichen Anforderungen im Wortlaut des Kunden.

2. Warenkorb bestellen (3P)

- o Der Inhalt des Warenkorbs wird schon richtig angezeigt. Das reicht fürs erste völlig. Was jetzt fehlt ist freilich, dass meine Kunden den Warenkorb bestellen können. Wir brauchen jetzt keine Bezahlungsfunktion. Das werde ich weiterhin per Rechnung machen. Aber irgendwie muss ich mitbekommen, wenn jemand bestellt hat. Damit nicht zu viel Aufwand entsteht, machen wir es einfach so:
 - Der Kunde soll seinen Namen eingeben und seine Adresse, damit ich die Bilder dann verschicken kann. Ach ja, eine Telefonnummer wäre für Rückrufe auch ganz nett. Soll aber optional bleiben.
 - Wenn der Kunde bestellt hat, bitte irgendeine nette Meldung zurückgeben, damit der Kunde weiß, dass es funktioniert hat. Ihr wisst, wie sowas aussehen muss. Ich will nicht, dass ich dann ständig Anrufe von verwirrten Kunden bekomme. Da vertraue ich auf eure Kompetenz.
 - Für mich bitte ein .txt File pro Bestellung in ein Verzeichnis (sagen wir „Bestellungen“) ablegen, das die Kundendaten sowie alle Bilder mit Pfad beinhaltet. Ach ja, bitte im Dateinamen das Bestelldatum mit reinnehmen. Im Text selber soll das Bestelldatum auch irgendwie ersichtlich sein.

2. Bilder löschen (1P)

- o Was fehlt, ist den Warenkorb ändern zu können. Vorerst bitte je Bild im Warenkorb einen „Entfernen“-Knopf anbieten.

3. Warenkorb leeren (1P)

- o Bitte einen Knopf zur Verfügung stellen, damit die Kunden einen Warenkorb auch neu anfangen können. Sprich ein Knopf, der den gesamten Warenkorb einfach löscht.

Sicherlich habe ich kleine Details vergessen. Aber ihr habt ja den Ruf, dass eure Kunden immer sehr begeistert von der Benutzerfreundlichkeit eurer Software sind. Deswegen habe ich euch ja auch engagiert – ich habe einfach nicht die Zeit mir das alles so genau durchzudenken. Ihr seid die Profis!