

Aufgabenstellung 3: REST-Applikation

Erstellen Sie die Web-Anwendung „HNFanShop“ auf der Basis des REST-Ansatzes, mit der der Online-Kauf von „HN-Fan“-Artikeln simuliert werden kann.

Fachliche Anforderungen

Berücksichtigen Sie bei der Web-Anwendung „HNFanShop“ folgende fachlichen Anforderungen:

- die Bearbeitung beginnt mit der Sicht „Übersicht Fan-Artikel“: die bestellbaren Fan-Artikel werden in übersichtlicher Form angezeigt und der aktuelle Inhalt des Warenkorbs (Anzahl Artikel, Gesamtsumme) ausgewiesen (also: zunächst leer)
 - zu jedem Fan-Artikel werden mindestens die Artikelnummer, die Artikelbezeichnung und der Preis angezeigt
 - zu jedem Fan-Artikel kann eine Detailsicht aufgerufen und angezeigt werden:
 - dazu wird der Fan-Artikel mit einem Maus-Klick ausgewählt
 - die Anzeige wird durch die Bedienung eines Schalters unterhalb der Liste ausgelöst
 - der Schalter wird mit „Fan-Artikel anzeigen“ bezeichnet
 - die angezeigten Fan-Artikel können in den Warenkorb abgelegt werden:
 - dazu wird der Fan-Artikel mit einem Maus-Klick ausgewählt
 - der Vorgang wird durch die Bedienung eines Schalters unterhalb der Liste ausgelöst
 - der Schalter wird mit „ausgewählten Fan-Artikel in Warenkorb legen“ bezeichnet
 - die Anzeige des Warenkorbs wird aktualisiert
 - durch Anklicken der Anzeige des Warenkorbs wird zur „Detaildarstellung Warenkorb“ gewechselt
- in der Sicht „Detaildarstellung Warenkorb“
 - werden die im Warenkorb abgelegten Fan-Artikel angezeigt (Liste entsprechend der Anforderungen oben sowie bei jedem Fan-Artikel die Anzahl) und die Gesamtsumme des Werts der Fan-Artikel
 - können die angezeigten Fan-Artikel aus dem Warenkorb entfernt werden:
 - dazu wird der Fan-Artikel mit einem Maus-Klick ausgewählt
 - der Vorgang wird durch die Bedienung eines Schalters unterhalb der Liste ausgelöst
 - der Schalter wird mit „ausgewählten Fan-Artikel entfernen“ bezeichnet
 - die Anzeige des Warenkorbs wird aktualisiert
 - kann die Anzahl einzelner angezeigter Fan-Artikel verändert werden:
 - dazu wird der Fan-Artikel mit einem Maus-Klick ausgewählt
 - durch die Bedienung des Schalters „+“ unterhalb der Liste wird die Anzahl des ausgewählten Fan-Artikels inkrementiert (maximal: 9)
 - durch die Bedienung des Schalters „-“ unterhalb der Liste wird die Anzahl des ausgewählten Fan-Artikels dekrementiert (minimal: 1)
 - die Anzeige des Warenkorbs wird bei jeder Schalterbedienung aktualisiert
 - wird durch die Bedienung des Schalters „Kaufen“ zur Sicht „Bestätigung Kauf“ gewechselt
 - wird durch die Bedienung des Schalters „Zurück“ zur Sicht „Übersicht Fan-Artikel“ gewechselt
- in der Sicht „Bestätigung Kauf“
 - werden die im Warenkorb vorhandenen Fan-Artikel mit Wert und Anzahl sowie der Gesamtsumme ausgewiesen
 - wird die Eingabe der persönlichen Daten des Bestellers als Pflichteingabe ermöglicht (hier vereinfacht: Name und Vorname) oder es wird ein bereits gespeicherter Besteller ausgewählt
 - wenn neue Bestellerdaten eingegeben werden, werden diese gespeichert, falls sie nicht schon vorhanden sind (Dublekkenkontrolle)
 - wird der Bestellvorgang „durchgeführt“, wenn der Schalter „Verbindlich kaufen“ bedient wird
 - es wird eine Bestätigung angezeigt, die den Namen und Vornamen und die Gesamtsumme ausweist
 - nach Bedienung des Schalters „Weiter“ wird zur Sicht „Übersicht Fan-Artikel“ gewechselt; der Warenkorb wird dabei geleert
 - der Bestellvorgang wird storniert, wenn der Schalter „Stornieren“ bedient wird
 - es wird zur Sicht „Übersicht Fan-Artikel“ gewechselt; der Warenkorb wird dabei geleert
 - wird durch die Bedienung des Schalters „Zurück“ zur Sicht „Detaildarstellung Warenkorb“ gewechselt.

Es gibt ferner eine gesonderte Administrationssicht, die als eigenständige Webseite aufgerufen wird, mit folgenden Möglichkeiten:

- Anzeige der Bestellungen, die bestätigt wurden, mit:
 - den bestellten Artikeln
 - den Daten des Bestellers
- Anzeige der gespeicherten Bestellerdaten mit:
 - den vorgenommenen Bestellungen.

Eine Benutzerverwaltung wird nicht benötigt.

Die Fan-Artikel sind ein vorgegebener Datenbestand, der serverseitig in JSON-Dateien vorgehalten wird. Eine Pflegefunktion für diesen Datenbestand ist nicht erforderlich.

Anforderungen an die Umsetzung

- Webclient:
 - „Single-Page“-Anwendung: die Sichten werden zur Laufzeit ausgetauscht, ohne die Webseite vollständig neu zu laden; Strukturierung des clientseitigen Teils der Anwendung in sinnvolle Einheiten durch:
 - Berücksichtigung des REST-Ansatzes (siehe Anforderungen an den Webserver)
 - Datenaustausch erfolgt im Hintergrund mit dem „AJAX“-Ansatz (Nutzung des XMLHttpRequest-Objekt); verwenden Sie dazu die javascript-Bibliothek jquery (\$.ajax und ggf. weitere Funktionen)
 - Verwendung von „inheritance.js“, um eine objektorientierte Notation in javascript zu simulieren
 - Verwendung von „es.js“ zur Verwendung des „Publish-Subscriber“-Muster
 - Verwendung von „tm.js“ und „te.js“ als clientseitiges Template-System
 - Verwendung HTML5 (XML-konforme Notation)
 - Überprüfung des Markup mit Hilfe der w3c-Validator-Dienste
 - Präsentation mit CSS, ausgelagert in eine externe CSS-Datei
 - die Verwendung von „CSS-Frameworks“ wie „bootstrap“ wird NICHT zugelassen
- Webserver:
 - Verwendung Python (3.3, entsprechend Vorbereitungsstermin)
 - Verwendung Framework „cherrypy“ (3.5, entsprechend Vorbereitungsstermin)
 - Auslieferung statischer Inhalte
 - REST-Interface
 - Bearbeitung der einzelnen Ressourcen
 - Verwendung cherrypy - „Method-Dispatcher“ anstelle des sonst üblichen Object-Dispatching
 - Datenspeicherung mit Hilfe des Python-Modul „json“ (siehe Python-Dokumentation) in einzelnen Dateien des serverseitigen Dateisystems in einem Unterverzeichnis „data“

Der „Single-Page“-Ansatz etc. ist entsprechend auch für die Administrationssicht zu berücksichtigen.

Hinweise zum REST-Interface

Beim REST-Ansatz werden Ressourcen, die durch die HTTP-Standardmethoden bearbeitet werden können, als allgemeines Konzept verwendet. Die Ressourcen werden durch URI eindeutig identifiziert. Ressourcen können nicht nur Dinge repräsentieren, sondern auch Verfahrensweisen.

In der Aufgabenstellung können etwa folgende Ressourcen und Methoden auftreten:

Ressource = URI	HTTP-Methode	Bedeutung
/artikel	GET	Daten aller Artikel bereitstellen
/artikel/:id	GET	Daten des Artikels bereitstellen

/besteller	GET	Daten aller Besteller bereitstellen
/besteller/:id	GET	Daten des Bestellers bereitstellen
/besteller + Daten im Body	POST	neue Besteller-Daten; id wird als Ergebnis geliefert (Dublettenkontrolle!)
/warenkorb/:id	GET	Daten des Warenkorbs bereitstellen (einschließlich Bestellpositionen)
/warenkorb + Daten im Body	POST	neuer Warenkorb; id wird als Ergebnis geliefert
/warenkorb/:id + Daten im Body	PUT	Daten des Warenkorbs aktualisieren
/warenkorb/:id	DELETE	Warenkorb löschen
/bestellposition/:id-warenkorb/:id-artikel + Daten im Body	POST	Artikel in Warenkorb ablegen
/bestellposition/:id-warenkorb/:id-artikel + Daten im Body	PUT	Daten bei Artikel in Warenkorb aktualisieren (z.B. die Anzahl)
/bestellposition/:id-warenkorb/:id-artikel	DELETE	Artikel aus Warenkorb entnehmen
/bestellung	GET	Daten aller Bestellungen bereitstellen
/bestellung + Daten im Body	POST	neue Bestellung; id wird als Ergebnis geliefert
/bestellung/:id + Daten im Body	PUT	Aktualisierung der Daten der Bestellung (z.B. Bestätigung)
/bestellung/:id	DELETE	Stornierung einer Bestellung

Die mit einem Doppelpunkt gekennzeichneten Pfadbestandteile enthalten zur Laufzeit die entsprechenden Identifikationen.

Anforderungen an die Dokumentation

Erstellen Sie eine Dokumentation, die Ihre Lösung beschreibt. Verwenden Sie folgende Gliederung (Hinweise zu den Inhalten sind *kursiv* gesetzt):

1. allgemeine Beschreibung Ihrer Lösung (*wozu dient die Anwendung, welche Ansätze werden bei der Implementierung verfolgt*)
2. Beschreibung des Datenaustauschs zwischen Client und Server in tabellarischer Form
 - *Beschreibung aller Anfragen an den Server*
 - *Beschreibung der erwarteten Verarbeitung auf dem Server*
 - *Beschreibung der erwarteten Antworten an den Client*
3. Beschreibung der clientseitigen Komponenten (*für jede von Ihnen erstellte Komponente*)
 - 3.1 Zweck
 - 3.2 Aufbau (Bestandteile der Komponente)
 - 3.3 Zusammenwirken mit anderen Komponenten (*Kommunikation mit es.js / Publish-Subscribe*)
 - 3.4 API (*öffentlich verfügbare Methoden*)
4. Beschreibung der serverseitigen Komponenten (*für jede von Ihnen erstellte Komponente*)
 - 4.1 Zweck
 - 4.2 Aufbau (Bestandteile der Komponente)
 - 4.3 Zusammenwirken mit anderen Komponenten
 - 4.4 API (*öffentlich verfügbare Methoden*)
5. Durchführung und Ergebnis der geforderten Prüfungen.

Geben Sie einleitend Ihre Gruppenzugehörigkeit, den Aufbau Ihres Team und das Gültigkeitsdatum der Dokumentation an.

Die Dokumentation wird als utf-8 kodierter Text mit der einfachen Auszeichnungssprache „markdown“ erstellt. Mit Hilfe des Werkzeugs „pandoc“ (siehe Quellen) erfolgt die Umsetzung in eine HTML-Datei:

```
pandoc -f markdown -t html5 -s --toc <IhreDatei> -o <IhreHTML5Datei>
```

Das Werkzeug „pandoc“ muss genutzt werden. Die Erstellung der Dokumentation mit anderen „markdown“-Werkzeugen / Editoren wird nicht zugelassen. Die in „pandoc“ verfügbaren Erweiterungen der Auszeichnungssprache „markdown“ müssen genutzt werden, z.B. bei der Verwendung von Tabellen.

Hilfsmittel, Quellen

Hilfsmittel	Quelle	Beschreibung / Hinweise
w3c-Validator-Dienste (Markup)	http://validator.w3.org/	Überprüfung der Korrektheit des Markup
w3c-Validator-Dienste (CSS)	http://jigsaw.w3.org/css-validator/	Überprüfung von CSS-Stilregeln
JSON	http://www.json.org	Beschreibung des Datenformats JSON (die dort angegebenen Implementierungen benötigen Sie nicht; sie sind bereits in Python enthalten)
json	(Python-Standard-Modul)	Umwandeln python-Datenstrukturen <-> JSON
jquery	http://jquery.com/	js-Bibliothek; verwenden Sie die Version >= 2
inheritance.js	siehe Anlage	zur Bereitstellung einer objektorientierten Notation in javascript
es.js	siehe Anlage	Implementierung des Publish-Subscriber-Musters
tm.js und te.js	siehe Anlage	Template-Engine und Template-Manager
pandoc	http://johnmacfarlane.net/pandoc/	Konverter, hier für markdown -> html5

Bewertung / Testat

Zur Bewertung Ihrer Lösung im Hinblick auf die mögliche Erteilung des Testats müssen Sie vorlegen und erläutern:

- den von Ihnen erstellten Quellcode Ihrer Web-Anwendung
- die von Ihnen erstellte Dokumentation.

Sie müssen die Lauffähigkeit Ihrer Lösung und die Durchführung der Validierungen nachweisen.