

# Dokumentation Schnäppchen-News App

---

*Web-basierte Anwendungen 2 – Phase 2*

Von:  
Sven Bock                      Matrikel-Nr. 11074903  
Florian Wolf                  Matrikel-Nr. 11077878

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>1-2</b>
<b>1 Projektvorfeld (IST-Analyse) .....</b>	<b>1-3</b>
<b>2 Projektidee - Schnäppchen-News App .....</b>	<b>2-3</b>
2.1 Problemstellung .....	2-3
2.2 Zielsetzung .....	2-3
2.3 Umsetzung .....	2-3
<b>3 Konzeptioneller Meilenstein - Kommunikationsabläufe und Interaktionen .....</b>	<b>3-4</b>
3.1 Ressourcen mit zu übermittelnden Daten .....	3-4
3.2 Interaktionen .....	3-7
<b>4 Projektbezogenes XML Schema .....</b>	<b>4-8</b>
<b>5 Ressourcen und die Semantik der HTTP-Operationen .....</b>	<b>5-9</b>

# 1 Projektvorfeld (IST-Analyse)

Die zweite Phase des Moduls „Webbasierte Anwendungen 2“ soll dazu dienen, ein von uns definiertes Konzept eines interaktiven Systems zu verwirklichen. Hierzu bedarf es erst einmal einer Problemfindung, die es in einem Projekt zu lösen gilt.

Anschließend müssen somit Kommunikationsabläufe und Interaktionen definiert werden, bevor weiterführende Aufgaben wie XML-Schemata, RESTful Webservice, XMPP und der Client erstellt und bearbeitet werden können.

## 2 Projektidee - Schnäppchen-News App

### 2.1 Problemstellung

Es ist absehbar, dass in naher Zukunft ein vorhandenes Gerät erneuert oder ausgetauscht werden muss, ein Vertrag ausläuft, eine Reise gebucht werden möchte oder eine Neuanschaffung geplant ist.

### 2.2 Zielsetzung

Eine für den Benutzer orientierte Schnäppchen-News App soll den potentiellen Käufer auf ihn zutreffende Benachrichtigungen zusenden.

### 2.3 Umsetzung

Für den Benutzer interessante Kategorien können ausgewählt werden, für diese dann eine Push-Benachrichtigung gesendet wird, sobald neue Schnäppchen verfügbar sind (asynchron).

Öffnet man die Applikation, werden alle Schnäppchen angezeigt. Diese sind nach Aktualität sortiert. Kategorien sind zusätzlich auswählbar (synchron).

Benutzer haben die Möglichkeit Schnäppchen zu melden, die dann in der Applikation veröffentlicht werden, sobald diese von den Administratoren geprüft werden (synchron).

### 3 Konzeptioneller Meilenstein - Kommunikationsabläufe und Interaktionen

#### 3.1 Ressourcen mit zu übermittelnden Daten

Im Folgenden werden die von uns definierten Ressourcen mit den dazugehörigen Daten, die übermittelt werden, definiert.

Als erste Überlegung haben wir folgende Ressourcen definiert und für jede eine XML-Datei angelegt:

##### **Schnäppchen**

Das Schnäppchen soll alle wichtigen Daten enthalten, die der Benutzer benötigt, um sich für das Schnäppchen zu interessieren. Da es nicht für jedes Schnäppchen unbedingt möglich oder notwendig ist ein Bild mit anzugeben, ist diese Funktion optional. Ebenso ist es möglich Kommentare zu verfassen, aber nicht zwingend erforderlich. Es wäre auch möglich gewesen den Preis und UVP im Text zu verfassen, sie werden jedoch als einzelne Elemente einheitlicher und besser hervorgehoben. Der UVP soll optional sein, da dieser nicht zwingend erforderlich sein muss. Er dient eher zum Vergleich. Oft existiert ein ungefährer Preis des Produktes schon in den Köpfen der Verbraucher. Kategorien besitzen vordefinierte Auswahlmöglichkeiten. Das hat den Hintergrund, dass nur Kategorien die auch zur Verfügung stehen, ausgewählt werden dürfen sobald man ein Schnäppchen anlegt.

- Titel
- Datum/Uhrzeit
- Text
- Bild (optional)
- Kategorie
- Preis
- UVP (optional)
- Kommentare (optional)

## **Schnäppchen melden**

„Schnäppchen melden“ soll Benutzern die Möglichkeit geben auch ohne Benutzerkonto ein Schnäppchen zu melden. Sie können bei Bedarf Bilder und/oder eine URL zu den Schnäppchen mitteilen. Als Beispiel für ein Bild könnte ein Foto eines Marktprospektes dienen oder für eine URL einen Link eines Onlinehändlers.

- Name
- Email
- Betreff
- Nachricht
- Bild (optional)
- URL (optional)

## **Push-Benachrichtigung**

Die Push-Benachrichtigung soll die unten aufgelisteten Informationen an den Client senden. Mehr Informationen würde der Benutzer anfangs nicht benötigen, da er für mehr Interesse die Applikation aufrufen wird und ihm somit das gesamte Schnäppchen zur Verfügung steht.

- Titel
- Datum/Uhrzeit
- Preis

Weitere Überlegungen und die Beratung mit den Betreuern haben uns jedoch dazu veranlasst aus diesen drei Ressourcen lediglich eine zu erstellen. Das liegt an der Tatsache, dass alle benötigten Informationen anhand der XML-Datei „Schnäppchen“ übermittelt werden können. Da somit die Komplexität des Projektes nicht ausreicht, haben wir uns für zwei weitergehende Ressourcen entschlossen (Benutzer und Märkte).

## **Schnäppchen**

Für das Schnäppchen wurden aufgrund der neuen Überlegungen einige Elemente hinzugefügt. Demnach besitzt nun jedes Schnäppchen eine ID, es ist einem bestimmten Benutzer zugehörig, sowie einem Markt o.ä. in dem es zu erwerben ist. Der UVP und der Preis sind in dem Element Preise zusammengefasst worden. Ansonsten entspricht die obige Beschreibung in der ersten Überlegung noch den Tatsachen.

- ID
- Titel
- Benutzer

- Kategorie
- Bild (optional)
- Markt
- Preise
- Text
- Datum
- Kommentare (optional)

## **Benutzerprofile**

Der Benutzer kann unter seinem Benutzernamen Schnäppchen melden. Das Passwort wird benötigt, damit er sich anmelden kann. Die Email ist notwendig, um mit dem Benutzer in Kontakt zu treten. Auf dem Benutzerprofil kann man ebenso erkennen, wie viele Schnäppchen diese Person bereits hinzugefügt hat und ihn Bewerten für gute oder schlechte Meldungen. Das ermöglicht anderen Personen die Seriosität eines Nutzers zu begutachten. Eine Überlegung eines weiteren Elementes war ein Counter, der umgesetzt werden könnte. Diese Idee haben wir aber nach kurzer Diskussion wieder verworfen.

- Benutzer\_ID
- Benutzername
- Passwort
- Name
- Vorname
- Email
- Anmeldedatum
- Adresse
- Anzahl Schnäppchen
- Gemeldete Schnäppchen
- Bewertung (optional)

## **Märkte**

Die Märkte sind so aufgeteilt, dass man zuerst den Kettenamen definiert. Diese Kette hat dann mehrere Märkte, die unterschiedliche ID's und Adressen haben. Zudem kann es sein, dass nicht jeder Markt dieser Kette die gleichen Schnäppchen anbietet, deshalb sind die Schnäppchen Marktabhängig. Die Adresse soll optional sein, da es auch online Unternehmen gibt, die Schnäppchen anbieten und diese nur versenden. Dafür ist keine Anschrift nötig.

- Kette
  - Name
  - ID
  - Adresse (optional)
  - SchnäppchenXYZ

## 3.2 Interaktionen

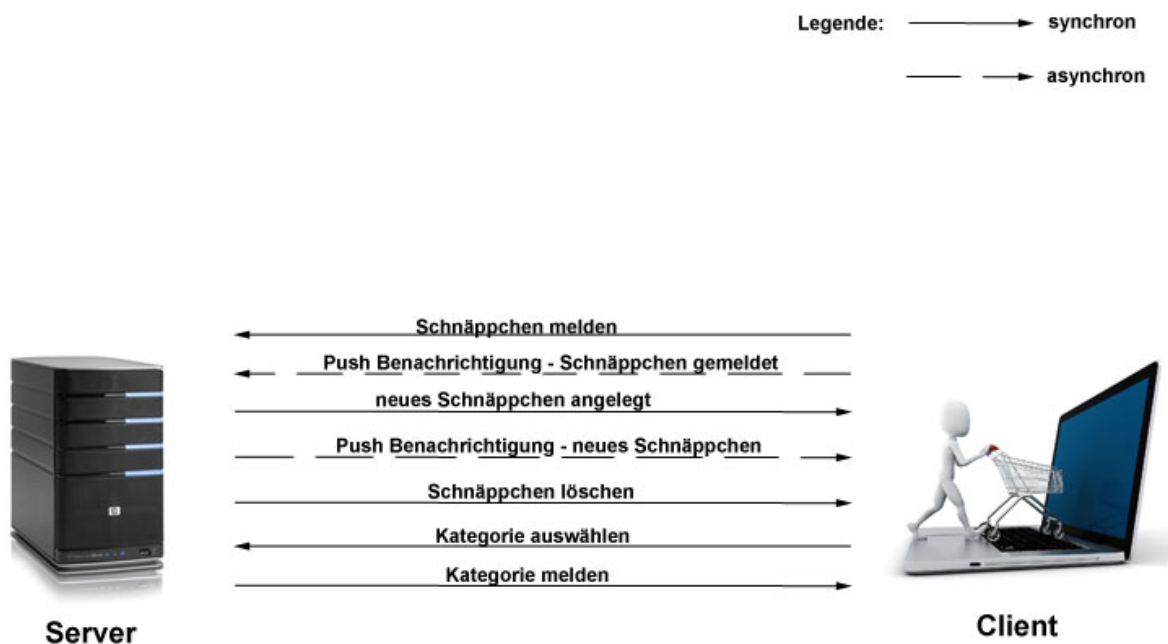


Abb. 1 - Kommunikationsabläufe und Interaktionen

### **Schnäppchen melden (synchron)**

Der Benutzer übermittelt über den Client ein zuvor ausgefülltes Schnäppchen an den Server.

### **Push-Benachrichtigung - Schnäppchen gemeldet (asynchron)**

Nachricht für den Server, dass ein neues Schnäppchen vom Client gemeldet wurde.

### **Neues Schnäppchen angelegt (synchron)**

Der Server legt ein neues Schnäppchen an.

Alternativ hier zu hatten wir noch angedacht, dass ein gemeldetes Schnäppchen durch einen Administrator freigegeben wird. Haben uns jedoch für ein Community basiertes System entschieden haben und somit verfiel auch der Gedanke, dass der Administrator die Informationen über das Schnäppchen selber einträgt oder nur gemeldete Schnäppchen der Benutzer freigibt.

### **Push-Benachrichtigung - neues Schnäppchen (asynchron)**

Nachricht beim Client, dass ein neues Schnäppchen eingetroffen ist.

### **Schnäppchen aktualisieren (lokal beim Client)**

Normale Aktualisierungsfunktion im Client.

**Schnäppchen löschen (synchron)**

Server teilt dem Client die gelöschten Schnäppchen mit.

**Bestimmte Kategorien anzeigen (lokal beim Client)**

Der Client abonniert bestimmte Kategorien beim Server und bekommt nur für diese dementsprechend Push-Benachrichtigungen. Fraglich ist hier noch ob in der Applikation selber alle oder auch nur die bestimmten Kategorien angezeigt werden.

## 4 Projektbezogenes XML Schema

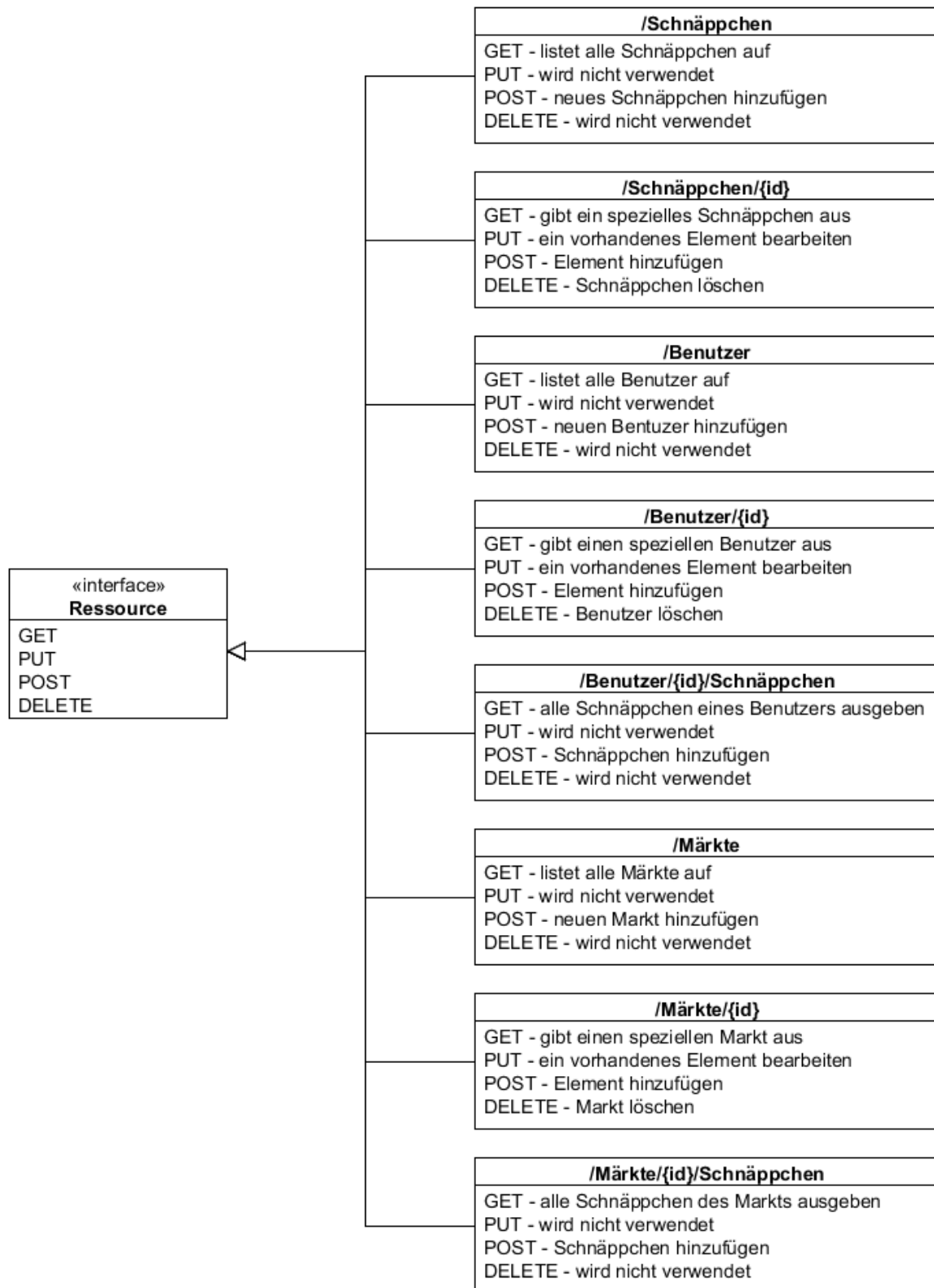
Zu diesem Meilenstein sollen die Planung und Konzeption für das Projekt nötiger, valider XML Schemata durchgeführt werden. Hinsichtlich der späteren Verwendung von JAXB sollte hier in jedem Fall darauf geachtet werden, dass die XML Schemata valide sind.

Bei der Entwicklung in XML



## 5 Ressourcen und die Semantik der HTTP-Operationen

Es sollen projektbezogene Beschreibungen der für das Projekt benötigten Ressourcen und Operationen inklusive Begründung der Entscheidungen erstellt werden.



Unsere Überlegung für die Wahl der geeigneten Ressourcen haben wir in der obigen Grafik festgehalten. Die drei ursprünglichen Ressourcen „Schnäppchen“, „Benutzerprofile“ und „Märkte“ sind bei dieser Aufgabe spezialisiert worden.

### **Schnäppchen**

Bei der Ressource „/Schnäppchen“ werden nur 2 der 4 Http-Operationen verwendet. Dies resultiert daraus, dass unter dieser Ressource alle Schnäppchen aufgelistet werden (GET) oder neue Schnäppchen hinzugefügt werden können (POST). Die anderen Operationen machen hier keinen Sinn, da kein Schnäppchen ausgewählt ist um es zu bearbeiten oder zu löschen. Hingegen bei der Ressource „/Schnäppchen/id“ wird nur ein spezielles Schnäppchen betrachtet (GET). Es können neue Elemente innerhalb des ausgewählten Schnäppchens hinzugefügt (POST) oder bearbeitet (PUT) werden. Ebenso ist es möglich das spezielle Schnäppchen zu löschen (DELETE).

### **Benutzer**

Die Operationen für „/Benutzer“ sind ähnlich die von „/Schnäppchen“. GET listet hier alle Benutzer auf und mittels POST können Neue hinzugefügt werden. PUT und DELETE werden auch hier nicht benötigt.

Demnach verhält sich die Ressource „/Benutzer/id“ äquivalent zu „/Schnäppchen/id“. Man kann sich einen speziellen Benutzer anzeigen lassen (GET), vorhandene Elemente des Benutzers bearbeiten (PUT) oder hinzufügen (POST) oder aber einen speziellen Benutzer löschen.

„/Benutzer/id/Schnäppchen“ gibt alle Schnäppchen eines bestimmten Benutzers aus (GET) oder fügt ein Schnäppchen für einen Benutzer hinzu (POST). PUT und DELETE werden auch hierbei nicht verwendet.

Die Überlegung, ob eine weitere Ressource behandelt wird, die ein bestimmtes Schnäppchen eines speziellen Benutzers ausgibt haben wir wieder verworfen. Dies würde keinen eindeutigen Qualitativen Mehrwert für das Projekt geben, lediglich mehr an der Quantität.

### **Märkte**

Innerhalb der Ressource „/Märkte“ kann man mittels GET alle Märkte auflisten und mit POST neue Märkte hinzufügen. Die beiden anderen Operationen werden auch hier nicht verwendet.

„/Märkte/id“ gibt einen speziellen Markt aus (GET), löscht einen Markt (DELETE), bearbeitet ein vorhandenes Element dieses Marktes (PUT) oder fügt ein neues Element hinzu (POST). Bei „/Märkte/id/Schnäppchen“ kann man sich alle Schnäppchen anzeigen lassen, die in

einem gewählten Markt verfügbar sind (GET) oder diesem Schnäppchen hinzufügen (POST). Auch hier werden PUT und DELETE nicht verwendet, da kein spezielles Schnäppchen eines Marktes ausgewählt ist.

Ebenso wie bei Benutzer war auch hier angedacht, die Ressource für das spezielle Schnäppchen eines Marktes zu definieren. Aber aus dem gleichen Gründen wie oben beschrieben, wird dies nicht durchgeführt.