

## Übungsaufgaben Blatt 9 - jQuery und AJAX

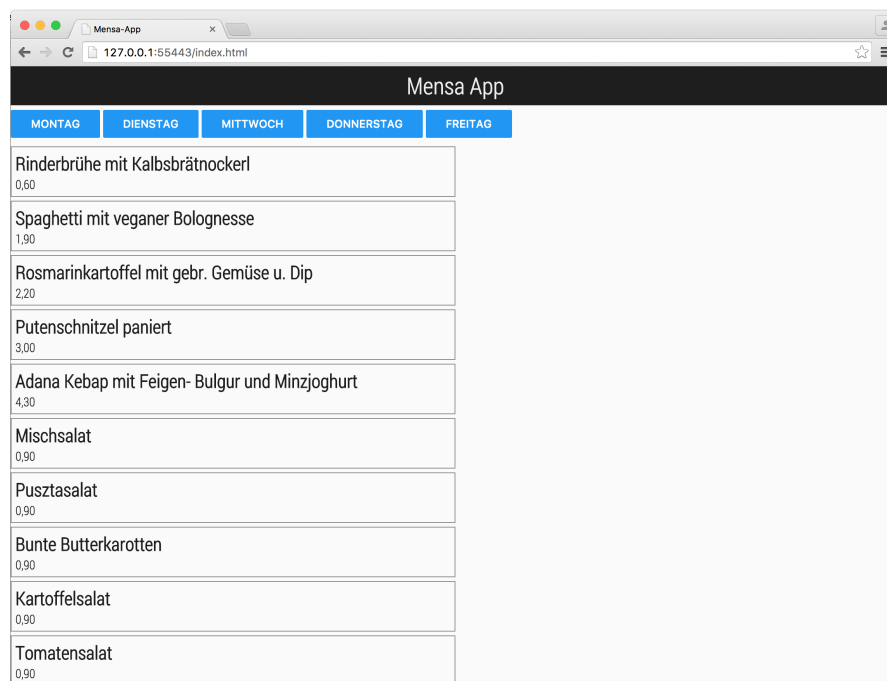
Medieninformatik Webtechnologien - Prof. Dr. Markus Heckner

### Aufgabe 1 : App zur Abfrage des Mensa Speiseplans

Ziel dieser Aufgabe ist es eine JavaScript Applikation zu entwickeln, mit deren Hilfe sich der Mensa-Speiseplan der aktuellen Woche abrufen lässt. Sie sollen dazu einen Webservice verwenden, der Ihnen die Gerichte eines Wochentags als JSON-formatierte Antwort zurückgibt.

Verwenden Sie für Ihre Lösung jQuery zur Abfrage des Speiseplans.

Ihre fertige Lösung sollte wie auf dem folgenden Screenshot aussehen:



Ihre App soll die folgenden Anforderungen erfüllen:

- Laden des aktuellen Speiseplans eines Wochentags, bei Klick auf den entsprechenden Button (Montag, Dienstag, Mittwoch, Donnerstag, Freitag).
- Anzeige des Gerichts und des Preises für Studenten.

Für diese Aufgabe erhalten Sie über Grips wieder ein Starterpaket, das den HTML-Code, die jQuery Library, das CSS sowie eine leere JavaScript Datei enthält.

## Das MensaApp Backend REST API

Die Daten für die MensaApp erhalten Sie von einem bestehenden Webservice. Über diese CSV2JSON-Bridge können Sie die Speisepläne der Uni Mensa im JSON-Format über eine REST-Schnittstelle beziehen. Das zurückgelieferte JSON besteht aus einem Array, in dem alle verfügbaren Hauptgerichte, Suppen, Beilagen und Desserts des entsprechenden Tags als einzelne Objekte enthalten sind. Über die Eigenschaften der Objekte kann auf deren Beschreibung, Kategorie, Preis und Labels zugegriffen werden.

Beispielsweise sieht die Antwort für einen Wochentag wie folgt aus (es werden nicht alle Gerichte angezeigt):

```

1  [
2    {
3      "name": "Feine Kraeutersuppe",
4      "day": "Mo",
5      "category": "Suppe",
6      "labels": "V",
7      "cost":
8        {
9          "students": "0,60",
10         "employees": "0,80",
11         "guests": "1,30"
12       },
13      "id": 50,
14      "upvotes": 7,
15      "downvotes": 2
16    },
17
18    {
19      "name": "Gruene Nudeln mit Gorgonzola",
20      "day": "Mo",
21      "category": "HG1",
22      "labels": "V",
23      "cost":
24        {
25          "students": "1,90",
26          "employees": "2,70",
27          "guests": "3,50"
28        },
29
30      "id": 12,
31      "upvotes": 12,
32      "downvotes": 1
33    },
34    ... ]
35  
```

Eine ausführliche Anleitung zur Nutzung dieses REST APIs, sowie Beispiele für die Antworten des Webservices erhalten Sie unter der folgenden URL: <http://api.regensburger-forscher>.

[de/mensa/](#)

### Achtung:

Verwenden Sie unbedingt die folgende URL zur Abfrage der Daten <http://132.199.139.24:9001/mensa/uni/> und nicht die in auf der API-Website angegebene Variante (<http://api.regensburger-forscher.de/mensa/uni/>).

Den Speiseplan für Montag der aktuellen Woche erhalten Sie dementsprechend unter der folgenden URL: <http://132.199.139.24:9001/mensa/uni/mo>.

### Tipps zur Bearbeitung:

- *Think big start small:* Versuchen Sie erst die Daten für einen Wochentag zu laden. Erst wenn das funktioniert, können Sie sich um die anderen Wochentage kümmern.
- Die Abfrage der Mensagerichte erfolgt über die jQuery-Methode **ajax**. Das folgende Codebeispiel zeigt einen funktionierenden Aufruf dieser Methode:

```
1      $.ajax({  
2          url: dataForDayURL,  
3          dataType: "json",  
4          success: mensaResultsReceived,  
5          error: errorRequestingMensaData  
6      });
```

- **url**: Hier steht die vollständige URL zum Aufruf des Webservices.
- **mensaResultsReceived** ist eine Funktion, die aufgerufen wird, wenn der Aufruf erfolgreich war. Diese Funktion erhält ein Argument **data**, das die Antwort des Webserverns enthält.
- **errorRequestingMensaData**: Diese Funktion wird aufgerufen, falls der Aufruf nicht erfolgreich war. Sie finden Sie Parameter dieser Funktion in der jQuery API Dokumentation: <http://api.jquery.com/jquery.ajax/>.
- Auslesen der Antwort in der Funktion **mensaResultsReceived**. Wie beschrieben erhält diese Funktion ein Argument **data**, aus dem Sie die Daten auslesen können. **data** ist ein Array, jedes Element des Arrays repräsentiert ein Gericht eines Tages. Beispielsweise können Sie mit dem folgenden Code den Namen des zweiten Gerichts des angefragten Tages auslesen:

```
1      var dishName = data[1].name;
```

**name** steht hier immer für den Namen des Gerichts. Vergleichen Sie die Dokumentation des Webservices hier, um herauszufinden, wie Sie weitere Daten auslesen können <http://132.199.139.24/~baa56852/www/mensa/>.

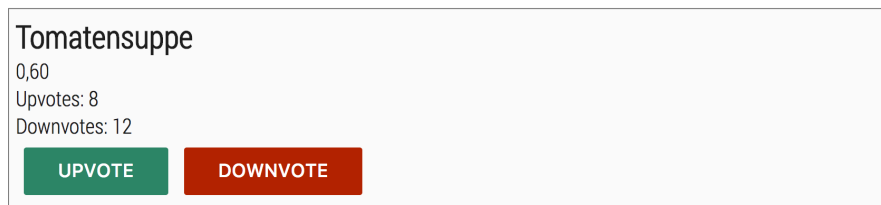
- Die einzelnen Gerichte eines Tages sollen als Listenelemente einer **<ul>** in das HTML Dokument eingefügt werden. Die **<ul>** ist bereits im HTML-Element vorhanden.

- Bei der Abfrage neuer Wochentage müssen Sie die Liste des vorigen Wochentags löschen. jQuery bietet Ihnen dazu eine schnelle und einfache Möglichkeit an. Recherchieren Sie die Lösung in Google. Eine sinnvolle Suchabfrage könnte wie folgt lauten: *jQuery remove child elements*.

### Aufgabe 2 : Voting-Funktionalität ergänzen

Das Mensa-API bietet Ihnen auch die Möglichkeit für eine bestimmte Speise eine Bewertung (d.h. *Upvote* oder *Downvote*) anzugeben (vgl. API Dokumentation).

Passen Sie Ihre Anwendung so an, dass die Anwendung die Bewertungen anzeigt und der Benutzer für jedes Gericht eine Bewertung abgeben kann. Ihre Lösung sollte so wie auf dem folgenden Screenshot aussehen und die Bewertungen sollen bei Klick auf den entsprechenden Button abgegeben werden. Der Server gibt Ihnen auch eine Antwort auf eine Bewertung. Lesen Sie diese Antwort aus und aktualisieren Sie anschließend ohne Neuladen der Seite die Votes des bewerteten Gerichts.



**Tipp:** Sie müssen Ihre HTML-Elemente eines bestimmten Gerichts kennzeichnen, um später feststellen zu können, welches Gericht *gevotet* werden soll. Die folgenden Hinweise helfen Ihnen dabei: [https://developer.mozilla.org/de/docs/Web/Guide/HTML/Using\\_data\\_attributes](https://developer.mozilla.org/de/docs/Web/Guide/HTML/Using_data_attributes).

### Erweiterungen (optional):

Sobald Ihre Anwendung funktioniert, können Sie noch die folgenden Erweiterungen ergänzen:

- Bauen Sie Ihr Beispiel so um, dass Ihre MensaApp dem **Revealing Module Pattern** folgt. D.h. bieten Sie ein klares Interface nach außen an und verstecken Sie den internen Zustand Ihrer Anwendung. Ergänzen Sie einen Controller, der die Ergebnisse des Mensa-Moduls entgegennimmt und die Darstellung aktualisiert.
- Zeigen Sie alle vegetarischen Gerichte mit grünem Rahmen oder grünem Text an. Eine Möglichkeit dazu ist den CSS Code anzupassen und mittels jQuery die Klasse des Listeneintrags entsprechend zu verändern.
- Preise für alle Gäste anzeigen. Erweitern Sie Ihren Code so, dass man auch die Preise für Mitarbeiter und Gäste ansehen kann.

- Gestalten Sie Ihre Liste so, dass eine klare Trennung zwischen Hauptgerichten, Beilagen und Nachtischen ersichtlich wird.