



2017-01-19

Rapport

Webbapplikationsutveckling



Mats Wikmar

KNOW YOUR BEER

Innehållsförteckning

1. Produkten	1
2. Syfte.....	1
3. Funktionalitet	1
3.1 Sektionen bakgrundsbild	1
3.2 Sektionen navigation	1
3.3 Sektionen glas (API request 1).....	1
3.4 Sektionen citat.....	1
3.5 Sektionen kategorier (API request 2 samt 3)	2
3.6 Sektionen beeronality (API request 4)	2
3.7 Sektionen Kontakta oss	2
4. API.....	2
5. Arbetsprocessen.....	3
6. Referenser	3
7. Länkar	3

1. Produkten

Uppdraget var att ta fram en webbsida till mitt slutprojekt i kursen YHJUST16 – Webbapplikationsutveckling. Då temat var valfritt så bestämde jag mig för att koda en sida om öl. Detta sker genom två 2st API: er (BreweryDB samt Flickr) vilket kommer öka kunskapen för besökaren *"Know your beer"*.

2. Syfte

- Ta fram en websida med valfritt tema som även hämtar data från två stycken API: er
- Lära mig mera om tekniken AJAX

3. Funktionalitet

Nedanför presenteras samtliga funktioner från websidan. JavaScript koden hittas i mappen: js/custom.js och b.l.a. index.html samt css/custom.css kan behövas för att förstå helheten.

3.1 Sektionen bakgrundsbild

Det första som besökaren ser vid ett besök är den *"levande"* bakgrunden. Det är en gif som upprepar sig själv (giphy, 2017). Denna heltäckande sektion utförs med hjälp av CSS och b.l.a. attributet:

`background-size: cover;`. Den har även en centrerad knapp i botten av skärmen, som tar besökaren vidare till nästa sektion.

3.2 Sektionen navigation

Högst upp på sidan så följer det alltid med en fixerad Bootstrap meny för både vanliga och mindre skärmar. Vid mindre skärmar så la jag även till ordet "Menu" till den inbyggda *"hamburger menu"* då vissa användare inte förstår dess betydelse (@lmjabreu, 2016). Navigeringen till respektive sektion utförs slutligen med hjälp av JQuery. Ett klickat ord hämtas ut och aktiveras i en JQuery `.animate(...)`

3.3 Sektionen glas (API request 1)

Den första sektionen efter 3.1 bakgrunden är en egengjord karusell. Här hämtas alla glas med anropet: `breweryAPIobj.getGlass("/glassware");`. Detta sker genom ett Ajax anrop (via en Proxy, då API: et inte stödjer direkta klient förfrågningar (Taran, 2015)) som sedan returnerar och lägger samtliga glas i en Array. Funktionen `createGlassSlider(brewery_glass){...}` sätter därefter ut den första bilden samt tillhörande text i sektionen. Vid ett klick på *"next"* eller *"prev"* btns så hämtar funktionen ut den nuvarande bilden index position i arrayen och ökar ev. minskar med ett beroende på ens riktning.

3.4 Sektionen citat

I denna sektion så genereras det ut slumpmässiga citat från filen: *api/dataQuotes.json*. Denna funktion aktiveras genom: `quoteObj.getQuotes();`. AJAX returnerar citat som aktiverar den interna funktionen `addQuoteTODOM();`. Vilket körs med en JQuery `fadeOut() fadeIn() setInterval(...)`

3.5 Sektionen kategorier (API request 2 samt 3)

Här genereras alla öl kategorier från BreweryDB. Som tidigare så körs det genom en AJAX fråga (via en PROXY). Denna sektion aktiveras med två stycken metoder. En egengjord laddare "roterande bild" som försvinner när nästa metod är färdig med hämtningen.

```
breweryAPIobj.loader();  
breweryAPIobj.getCategories("/categories");
```

Kategorierna läggs i en Array innan de slumpas ut på skärmen med en `shuffle()`. En Bootstrap image thumbnails skapas för respektive ord och som tidigare så används JQuery `fadeOut()` `fadeIn()` `setInterval(...)`. Via `activateMouseEvent(...)` så stoppas den nuvarande `fade` effekten så att besökaren kan kolla på respektive bild.

Ett klick på ens "kategori bild text" medför slutligen ett tredje API anrop mot Flickr. Detta fungerar genom att ett JQuery click event hämtar ut texten som sedan skickas iväg med Flickr metod. Detta genereras slutligen ut i en modal.

```
let query = $(this).text(); /* clicked text */  
flickrAPIobj.getImages(query); /* Images with a proxy */  
flickrAPIobj.getImagesNoProxy(query); /* Images without a proxy */
```

3.6 Sektionen beeronality (API request 4)

Här får besökaren svara på ett antal frågor. Detta generas med ett tillhörande JSON objekt. En summa räknas ut efter ett avslutat quiz och en unik ölsort och dess tillhörande glas generas ut.

```
endQuestion: function(){  
  console.log("end");  
  console.log("You got "+this.total+" points");  
  
  breweryAPIobj.getBeer("/beer/random",function(obj){...})
```

3.7 Sektionen Kontakta oss

Här presentas ett stylat formulär med CSS. Jag har valt färgerna svart, vitt, "öl gult" (som är temat för websidan). Utöver det så la jag även till en öl bild med CSS :valid samt :invalid selektors.

4. API

Webbsidan använder 2st API: er. Huvud API: et är ett som hanterar hämtningen av öl kategorier, glas, samt en ölsort baserat på ens frågor i sektionen "beeronality". Det andra API: et (Flickr) hämtar bilder baserat på klickad kategori-text. En notering är att BreweryDB endast tillåter förfrågningar via en Proxy (t.ex. PHP) (Taran, 2015).

Flickr API <https://www.flickr.com/services/api/>
BreweryDB API <http://www.brewerydb.com/developers>

Via en Proxy (PHP) eller utan
Endast via en Proxy(PHP)

5. Arbetsprocessen

Jag har i princip arbetat i samma ordning anges i 3 Funktionalitet efter att jag fick igång respektive API. Det som tog tid var att förstå hur API :erna kunde användas. Tex så stödde inte BreweryDB direkta AJAX anrop mot backend. Vidare så har de ett begränsat antal metoder när man kör med en gratis API nyckel. Vid slutet av utvecklingen så kollade jag även att olika webbläsare stödde min sida. Konstigt nog så bestämde sig **host** Internet Explorer för att inte visa min kod, vilket visade sig höra samman med den nya arrow function, for of samt this syntaxen. Det bidrog till att jag översatte en kopia till ES5 syntax.

I övrigt så har det gått bra, där jag har försökt att isolera kod till sitt objekt med dess metoder. Jag la även till doc. blocks kod ovanför respektive objekt, vilket förklarar dess syfte och ev. framtida förbättringar.

Förslag på förbättringar:

- *"Beeronality"* skulle behöva en
 - *"kör quiz igen metod samt knapp"*
 - Efter ett avslutad quiz så hämtas en unik random öl, men det skulle vara bättre om det var kopplat till den totala summan (*läs på ifall API stödjer en sådan metod*).
- Refaktorera alla AJAX anrop till ett.
- Lägg till en intervall() metod på glas karusellen.

6. Referenser

@lmjabreu, 2016. <https://lmjabreu.com>. [Online]

Available at: <https://lmjabreu.com/post/why-and-how-to-avoid-hamburger-menus/>
[Accessed 19 Januari 2017].

giphy, 2017. <http://giphy.com/>. [Online]

Available at: <http://giphy.com/gifs/beer-cinemagraph-92wsX8GEoNTYA>
[Accessed 19 Januari 2017].

Taran, 2015. *stackoverflow*. [Online]

Available at: <http://stackoverflow.com/questions/10365469/jquery-ajax-api-request-in-json-format-causes-uncaught-syntaxerror-unexpected>
[Accessed 19 Januari 2017].

7. Länkar

Websidan live: <https://www.bviking.se/lernia/webbappdev/>

Websidan repo: https://github.com/mow-github/course_webbappdev_finalproject