

## Assignment 2.

Anders Rochester, FWD 16

### Syfte och problem

Målet med denna applikation är att man skall kunna hämta nyheter från ett urvar news api generera innehållet i en sida med hjälp av flex-card. Utöver att visa nyheter skall man kunna spara anteckningar lokalt så de kan hämtas igen även om browsern uppdateras eller stängs av. Anteckningarna kan raderas, editeras och skrivas ut efter önskemål.

### Lösningar

En av utmaningarna för mig under arbetets gång var att hålla reda på rätt index när man klickar på de knappar man skapat och hämta rätt information från rätt värde i objekten. Den viktigaste funktionen gällande spara lokal data var även den en utmaning. Jag tittade runt och fastnade för en lösning att spara objektet direkt i en array. Som med hjälp av `JSON.parse` sparar och hämtar `localStorage`.

Jag kallade den `items` och tilldelade den en tom array:

```
24  var items = [];
```

Men om `local.Storage` är tom ifrån början får man ett felmeddelande och jag valde att fixa en enkel error-trap och läser endast in om `localStorage.items` inte är `"undefined"`;

En viktig sak att komma ihåg att applikationen består av två delar Notes och News. Eftersom News är live och ligger inne i AJAX-request så får man vara inne i scopet för att komma åt de olika korten. Det är så att säga två olika sätt att spara på. Sparar du på news sparar du hela artikeln medan om du sparar i Notes så uppdaterar du endast det specifika notes i objektet. Den skriver över med det nuvarande innehållet.

```
149  items[myIndex].notes = currentContent;
150  localStorage.items = JSON.stringify(items);
```

En viktig sak för mig var upplevelsen av presentationen. Alltså hur man som användare interagerar och vad som är väsentligt och inte. Frågan är inte enbart estetisk utan innehåller viktiga ställnings-taganden som t. ex. storlek på bilder, headlines, knappar. Hur knappar och länkar reagerar och till vilket syfte de gör det. Samtidigt skapas sidan beroende på vilka nyhetsrubriker och dess innehåll är inte statiskt utan väldigt dynamiskt. Där måste finnas utrymme för förändring och skalning i presentationen utan att den skall se konstig ut. För att åstadkomma en skalbarhet och dynamik valde jag att skapa en ram i html-koden och "skriva" koden med hjälp av `javaScrip`. Presentationen i css är flex-card och box-sizing: border-box.

Nedan följer en kort beskrivning av flödet i applikationen;

skapa XMLHttpRequest

```
21 var xhr = new XMLHttpRequest();
```

skapa objekt newsSource och tilldela API källor med nycklar

```
61 var newsSource = [  
62   {name: "Die Zeit", source: "https://newsapi.org/v1/  
articles?source=die-zeit&sortBy=latest&apiKey="},  
  ...  
  ...  
  ...  
];
```

välj från listan får index, tilldela val till newsSourceChoice

```
229 var newsSourceChoice = newsSource[sourceIndex].source;
```

kalla källan från API

```
230 xhr.open("GET", newsSourceChoice, true);  
231 xhr.send(null);
```

tilldela svaret till newsObject

```
194 var newsObject = JSON.parse(xhr.responseText);
```

Nu har vi nyheterna lokalt i objektet newsObject och nedan ser vi strukturen på objektet. Det innehåller en array articles med objekt som är de faktiska nyheterna

```
{ status: "ok", source: "bbc-news", sortBy: "top", articles [  
{ author: "", description: "", }
```

För varje item i articles presentera html

```
197 newsObject.articles.forEach(addNews);
```

Funktionen addNews() lägger till html-kod med hjälp av jQuery-metoden append()

```
103 $(".card-wrapper").append("<div class='card-main'> ...  
[mycket tags här]</br>");
```

På samma sätt skapas sidan med Notes

Malmö den 27 februari 2017