

Systemintegration och integration med tredjepartssystem

Föreläsning 01 - jQuery & API

Kursplan

- <https://server.pingpong.net/courseId/105426/content.do?id=47706183>

Dagens ämnen

- jQuery
- JSON
- API

Flipped classroom

- Vanligtvis får man efter en föreläsning i läxa att fördjupa sig om ämnet man har gått igenom.
- I det flippade klassrummet får man istället material att förbereda sig på inför nästa föreläsning.
- Detta ersätter inte föreläsningar, men det ger mer tid över till workshops, diskussioner, projekt och annat fördjupande arbete.
- Ni kommer att ha ett stort ansvar för att förstå materialet på egen hand. Studiegrupper är att rekommendera.
- Jag tänker mig att vi avsätter lite tid varje förmiddag till att gå igenom svårigheter med ämnet men jag kommer att förutsätta att ni förstår grunderna.
- https://sv.wikipedia.org/wiki/Omv%C3%A4nt_klassrum

Styra utseende med JS

- Man kan styra enskilda style-properties.
 - `document.getElementById("something").style.backgroundColor = "#ccdd33";`
- Man kan även lägga till eller ta bort klasser.
 - `document.getElementById("div1").classList.add("classToBeAdded");`
 - `document.getElementById("div1").classList.remove("classToBeRemoved");`

jQuery

- Inkluderas som ett vanligt JS.

- <https://code.jquery.com/>

```
<script src="https://code.jquery.com/  
jquery-3.3.1.min.js">
```

- Er egen JS-fil ska inkluderas *efter* jQuery för att ni ska kunna använda det. Ändra inget i jQuery-filen.

```
<script src="https://code.jquery.com/  
jquery-3.3.1.min.js">  
<script src="my_script.js">
```

jQuery

`$(selector).action()`

`$(this).hide()`

- gömmer aktuellt element.

`$("p").hide()`

- gömmer alla p-element.

`$(".test").hide()`

- gömmer alla element med klassen test.

`$("#test").hide()`

- gömmer elementet med id test.

jQuery

```
$ (document) .ready (function () {  
    $ ("button") .click (function () {  
        $ ("p") .toggle ();  
    } ) ;  
} ) ;
```

http://jsfiddle.net/emmio_micke/8bawsjv3/1/

http://jsfiddle.net/emmio_micke/8bv0p2fL/1/

http://jsfiddle.net/emmio_micke/3tc7kd6v/2/

http://jsfiddle.net/emmio_micke/7yo4g5Ls/6/

http://jsfiddle.net/emmio_micke/u7rgxL2o/1/

jQuery

```
// A $( document ).ready() block.  
$( document ).ready(function() {  
    console.log( "ready 1!" );  
});
```

```
// Shorthand for $( document ).ready()  
$(function() {  
    console.log( "ready 2!" );  
});
```

https://jsfiddle.net/emmio_micke/z264hp3a/2/

Uppgift

- Skapa en tabell över användare.
 - Förnamn, efternamn, epost
- Lägg till en knapp på varje rad som låter besökaren ta bort en användare.
 - Fråga om användaren är säker först med `confirm`.
 - Lägg därefter på en klass på raden som gör bakgrunden röd och gör en fadeout på elementet.
- Ni kan forka denna om ni vill:
https://jsfiddle.net/emmio_micke/qkrpngm4/1/

API

- När vi gör ett anrop till en webbsida, vad innehåller svaret?
- Kan man få andra resultat?
 - <https://randomuser.me/api/?results=5>
- Application Programming Interface
- Låter oss hämta det data vi behöver i ett format som vi kan bearbeta, vanligen JSON eller XML.

JSON

- JSON (JavaScript Object Notation), är ett kompakt, textbaserat format som används för att utbyta data.

- Ni minns detta sedan tidigare.

```
{
  "firstName": "Jason",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ],
  "newSubscription": false,
  "companyName": null
}
```

- `var obj = JSON.parse(text);`

Ajax med jQuery

- Asynchronous JavaScript and XML
- Asynkron datahämtning innebär att programmet kan köra flera "trådar" och väntar inte in resultatet från en instruktion innan den påbörjar nästa.
- Ajax innebär i realiteten att man kan utföra instruktioner i bakgrunden, t ex spara eller hämta information.
- Vi kan t ex hämta data i bakgrunden och uppdatera bara delar av webbsidan.

Asynkront utförande

```
const request = require("request");
request(
  "http://swapi.co/api/starships/10/",
  function(err, response, body) {
    console.log("Sen det här");
    console.log(JSON.parse(body));
  }
);
console.log("Det här kommer skrivas ut först");
```

- Exemplet ovan är för Node.JS.
- http://jsfiddle.net/emmio_micke/u8esvgdm/38/

Uppgift

- <https://randomuser.me/api/>
- Utgå från er användarlista från tidigare.
- Vid knapp-klick ska ni hämta data från api:et med ett Ajax-anrop och använda datat du får tillbaka när du lägger till den nya raden.

Tills i morgon

- Förstå grunderna i vad ett API är. Kolla in vilka källor som passar dig. Du behöver inte kolla alla, bara du förstår konceptet.
- <https://www.youtube.com/watch?v=s7wmiS2mSXY>
- <https://www.youtube.com/watch?v=cpRcK4GS068>
- <https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82>

Tills i morgon

- Förstå grunderna i vad ett REST-ful API är. Kolla in vilka källor som passar dig. Du behöver inte kolla alla, bara du förstår konceptet.
- <https://www.youtube.com/watch?v=e6h87rzeGJE>
- https://www.youtube.com/watch?v=QFeuZd_2fLQ
- <https://www.mulesoft.com/resources/api/what-is-rest-api-design>

Sammanfattning

- jQuery
- JSON
- API

Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?

Tack för idag!

Mikael Olsson
mikael.olsson@emmio.se
076-174 90 43

