

Systemintegration och integration med tredjepartssystem

Föreläsning 01 - jQuery & API

Dagens ämnen

- ReST
- Bygga API-backend

API Consumer

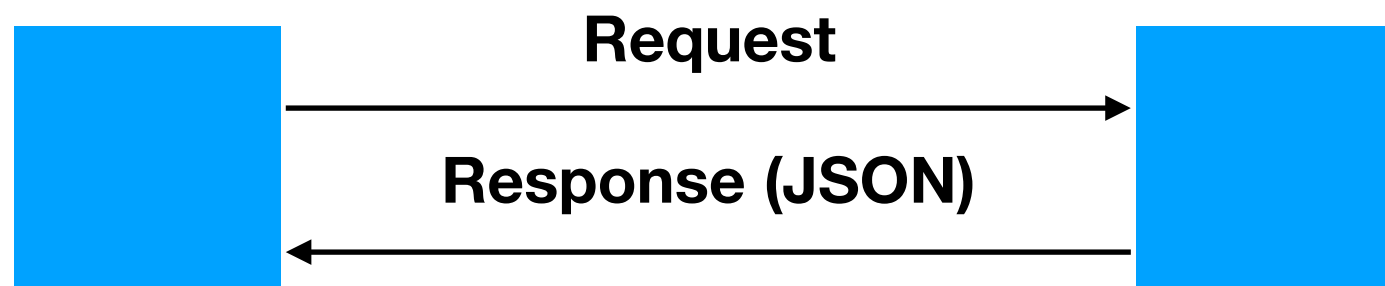
- En applikation som hämtar data från ett API brukar kallas *consumer*.
- Igår gjorde vi en consumer som hämtade data från <https://randomuser.me/api/>

```
1 ▾ $("button").click(function() {  
2 ▾   $.ajax({  
3     url: 'https://randomuser.me/api/?results=5',  
4 ▾   success: function(data) {  
5     console.log(data["results"][2]["name"]["first"]);  
6   }  
7 });  
8 });  
9
```

```
{  
  "results": [  
    {  
      "gender": "male",  
      "name": {  
        "title": "Mr",  
        "first": "John",  
        "last": "Doe",  
        "full": "Mr John Doe",  
        "picture": "https://randomuser.me/api/portraits/male/1.jpg",  
        "large": "https://randomuser.me/api/portraits/large/1.jpg",  
        "seed": "ae5ecc018f20af63",  
        "results": 1,  
        "page": 1  
      }  
    },  
    {  
      "gender": "female",  
      "name": {  
        "title": "Mrs",  
        "first": "Jane",  
        "last": "Doe",  
        "full": "Mrs Jane Doe",  
        "picture": "https://randomuser.me/api/portraits/female/1.jpg",  
        "large": "https://randomuser.me/api/portraits/large/1.jpg",  
        "seed": "ae5ecc018f20af63",  
        "results": 1,  
        "page": 1  
      }  
    },  
    {  
      "gender": "male",  
      "name": {  
        "title": "Mr",  
        "first": "John",  
        "last": "Doe",  
        "full": "Mr John Doe",  
        "picture": "https://randomuser.me/api/portraits/male/1.jpg",  
        "large": "https://randomuser.me/api/portraits/large/1.jpg",  
        "seed": "ae5ecc018f20af63",  
        "results": 1,  
        "page": 1  
      }  
    },  
    {  
      "gender": "female",  
      "name": {  
        "title": "Mrs",  
        "first": "Jane",  
        "last": "Doe",  
        "full": "Mrs Jane Doe",  
        "picture": "https://randomuser.me/api/portraits/female/1.jpg",  
        "large": "https://randomuser.me/api/portraits/large/1.jpg",  
        "seed": "ae5ecc018f20af63",  
        "results": 1,  
        "page": 1  
      }  
    },  
    {  
      "gender": "male",  
      "name": {  
        "title": "Mr",  
        "first": "John",  
        "last": "Doe",  
        "full": "Mr John Doe",  
        "picture": "https://randomuser.me/api/portraits/male/1.jpg",  
        "large": "https://randomuser.me/api/portraits/large/1.jpg",  
        "seed": "ae5ecc018f20af63",  
        "results": 1,  
        "page": 1  
      }  
    }  
  ],  
  "page": 1,  
  "results": 5  
}
```

Vår applikation

<https://randomuser.me/api/>



API Consumer

- Vi kan hämta data från API:er med PHP också.
- Olika metoder fungerar på olika servrar, beroende på vad som är tillåtet i PHP-inställningarna och vad som finns installerat på servern.
- Enklast är kanske `file_get_contents`.
- Anndra alternativ: `curl`, `fopen`
- <https://oscarliang.com/six-ways-retrieving-webpage-content-php/>

API Consumer

- exercises/02-api/01-consume.php

```
$url = 'https://randomuser.me/api/';  
$file_content = file_get_contents($url);  
var_dump($file_content);
```

API Consumer

- Vi skulle t ex kunna hämta data från ett API och spara i vår egen databas.
- Ibland är datat vi får strukturerat på ett sätt som gör det olämpligt för våra syften. Då kan vi strukturera om det och spara det.
- Vi kan *mellanlagra* data för att inte överbelasta API-servern.
- Vi får bättre performance om vi kan hämta data från vår egen databas.

API Provider

- En server som tillhandahåller ett API kallas API Provider.
- Vi kan börja med att göra en applikation som listar alla produkter i en html-tabell, precis som tidigare.
- Vi utgår från classic models. Få filen att funka:

```
api/products/list-html.php
```

API Provider

- Nu har vi gjort en fil som skapar html. Låt oss se om vi kan ta nästan samma fil och få den att producera ett lämpligt JSON-svar.

`api/products/read.php`

Testa

- Hur testar vi att anropet fungerar som det ska?
- Kan vi modifiera filen vi skapade förut?

`exercises/02-api/01-consume.php`

HTTP-verb

- Hittills har vi alltid använt GET i våra API-anrop. Vilka andra finns?
- Dessa är exempel, det finns ännu fler.

HTTP verb		CRUD
	POST	Create
	GET	Read
	PUT	Update/ Replace
	DELETE	Delete

POST

- Vi borde kunna fixa en sida som tar hand om POST-requests och skapar en ny produkt.

`api/products/write.php`

Testa

- Hur testar vi att posta data?
- Vår förra fil gör ett GET-request. Vi behöver göra ett POST-request.
- Det finns olika sätt, vi kan t ex använda Curl.

`exercises/02-api/02-post.php`

Endpoints

- De URL:er vi har skapat nu skulle kunna utgöra *endpoints*.
- En endpoint är i princip en punkt man kan ansluta till i ett API.

Hantera data med JS

- Nu när ni har börjat förstå hur API:er fungerar förstår ni kanske också lite mer om hur man kan använda JS för att hantera data?
- Istället för att spara data i en cookie, textarea eller liknande kan man ansluta till ett API och hämta/spara data den vägen.

Uppdatera

- Om det finns tid över, skapa en endpoint som låter användaren uppdatera data.
- Skapa en sida med ett formulär som innehåller produktdata.
- Låt användaren uppdatera data med JS/Ajax istället för att ladda om sidan med PHP.

Tills nästa tillfälle

- Skapa en endpoint som låter användaren ta bort en specifik produkt.
- Ladda ner Postman och lär dig förstå hur den fungerar.
 - https://www.youtube.com/watch?v=juldrxDrSH0&list=PLhW3qG5bs-L-oT0GenwPLcJAPD_SiFK3C
 - <https://learning.getpostman.com/getting-started/>
- Testa att interagera med ert api.

Sammanfattning

- ReST
- Bygga API-backend

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

