

# Övning 10: JavaScript och Webb API:er

Öva på att kalla på API:er och presentera data i DOM:en.

## Teori

**API** står för Application Programming Interface. Ett webb-api består av en eller flera publika endpoints. Oftast används http-protokollet och endpointen är helt enkelt en uri (webbadress) till var på servern data vi vill hämta ligger.

Hela systemet bygger på request och response. Om vi skickar ett request till endpointen `https://api.github.com/users/NET22B/repos` får vi tillbaka en respons med alla repositories som är upplagda för klassen (testa gärna att klistra in detta i er browser och se vad ni får tillbaka). Om vi vill bygga en hemsida för väderprognoser kan vi göra förfrågan (requests) till en endpoint i MetaWeathers API istället för att samla data själva. Olika api:er stödjer olika dataformat men json och xml är de vanligaste.

Det här möjliggör att vi kan använda oss av de resurser (data) som finns på api:et oberoende av vilket programmeringsspråk vi utvecklar med och oberoende av vilket språk api:et är byggt med.

**URI** står för Uniform Resource Identifier och används för att göra förfrågan till API-bibliotek.

`https://example.com/api/people?name=Kalle#nose`

Diagram illustrating the components of the URI:

- `https://`: schema
- `example.com`: authority
- `/api/people`: sökväg (path)
- `?name=Kalle`: fråga (query)
- `#nose`: fragment

**HTTP request methods**, även kallat **HTTP-verb**, anger vilken action man vill utföra med sin förfrågan. De fem vanligaste visas i tabellen nedan. Om vi till exempel vill hämta data så gör man ett GET request. Det enda ni behöver använda er av i den här övningen är GET!

HTTP-VERB	CRUD
POST	Create
GET	Read
PUT	Update/Replace (full update)
PATCH	Update/Modify (partiell update)
DELETE	Delete

**Request/response body** är datat som skickas, oftast i JSON format.

```
{
  "object": {
    "a": "b",
    "c": "d",
    "e": "f"
  },
  "array": [
    1,
    2
  ],
  "string": "Hello World"
}
```

**Postman** är ett program som kan användas till att anropa och testa api:er när man utvecklar.

<https://www.postman.com/downloads/>

## Uppgift 1: Star Wars Biometrics

The Star Wars API innehåller mängder av data gällande karaktärer, arter, rymdskepp, planeter osv från Star Wars-filmerna. Dokumentation: <https://www.swapi.tech/documentation>

Man kan till exempel lista alla karaktärer genom att göra en GET-request:

<https://www.swapi.tech/api/people>

Man kan också göra en GET-request med en query string för att få en specifik karaktär:

<https://www.swapi.tech/api/people/?name=chewbacca>

Er uppgift är att bygga en sida där användaren skriver in namnet på en Star Wars-karaktär och får tillbaka biometrisk data. Några exempelnamn är Luke Skywalker, R2-D2, Darth Vader, Leia Organa, Chewbacca.

1. Skapa en textruta för att ta mot namnet. Använd `<input></input>` och ge den attributen `type="text"`.
2. Skapa en knapp med `<button></button>`.
3. Skapa en textruta för output med `<textarea></textarea>` och ge den minst 5 fem rader med `rows`-attributet.
4. Skriv en funktion i `js`-filen som:
  - a. läser av värdet av input-textrutan och lägger till den till strängen `"https://www.swapi.tech/api/people/?name="` så vi får en fullständig URI,
  - b. använder URI:n som parameter till en `fetch`-request,
  - c. och skriver ut informationen till output-textrutan.

```
function getApi() {  
  
  /*Skriv din kod här*/  
  
  fetch(fullUri)  
    .then(res => res.json())  
    .then(data => {  
      /*Och här*/  
    })  
    .catch(err => console.log(err))  
}
```

`data` är ett JSON-objekt som innehåller all hämtad information om karaktären. Kolla i dokumentationen eller skriv ut med `console.log` för att se hur objektet är uppbyggt, Plocka ut `height`, `mass`, `gender` och `hair_color` och använd dem för att bygga en sträng med hjälp av template literals (bakåtvända fnuttar).

Exempel: `let s = `Value 1: ${value1}, Value 2: ${value2}`;`

5. Lägg till en `addEventListener` som lyssnar på knappen och kör ovanstående funktion vid klick.

## Uppgift 2: Dra ett kort

Deck of Cards API låter oss dra kort från en eller flera kortlekar. I response body så får man både vilket värde kortet har (suit och value) samt en url till en bild på det dragna kortet. Dokumentation: <http://deckofcardsapi.com/>

Ni ska skriva en sida som visar ett kort i taget när användaren klickar på en knapp.

1. Skapa en `<button></button>`.
2. Skapa en `<div></div>` där bilden på kortet ska skivas ut.
3. Skriv en liknande js-fil som i föregående uppgift med vissa skillnader:
  - a. Använd denna URI för fetchen:  
"`https://deckofcardsapi.com/api/deck/new/draw/?count=1`"
  - b. Kolla i dokumentationen för att se hur du ska plocka ut url:en till bilden på kortet från `data`.
  - c. Använd `createElement`, `setAttribute`, `innerHTML` och `appendChild` för att skapa ett image-element, sätta dess `src`-attribut, nollställa `div`-en och lägga till image-elementet till den.
4. Lägg till en `addEventListener` som lyssnar på knappen och kör ovanstående funktion vid klick.

## Extra

Bygg ett kortspel :) Tag hjälp av dokumentationen för apiet. Ni väljer själv vad ni bygger.

Ett enkelt Blackjack spel kanske?