



# Inlämningsuppgift 4

Betyg: U/G

## Syfte

Syftet med den fjärde inlämningen är att göra sig bekant med hur vi kan hämta information på ett dynamiskt vis (i detta fallet - utan att sidan laddas om) från ett öppet API och sedan bearbeta och presentera denna datan. Ett API, i detta fallet, innebär en tjänst (webbsida) som tillhandahåller ett gränssnitt för hur vi kan hämta en viss typ av data (i denna uppgiften är det data om filmer från den populära sidan [imdb.com](#)). Detta innebär att vi i denna inlämning kommer fortsätta arbeta med Document Object Model, dvs. HTML och JavaScript kombinerat, samt att innehållet i vår HTML kommer hämtas och presenteras från en extern datakälla (ett API). En sista punkt värt att nämna är att när vi hämtar data från detta API kommer vi göra det i formatet **JSON** - vilket i grunden kommer från JavaScript och kommer därför inte vara något svårt för vår del att arbeta med. Anledningen till att det är värt att nämna är att det är ett format som kan användas av flera programmeringsspråk, inte bara JavaScript, och är därför bra att känna till.

Inför denna uppgiften rekommenderas det starkt att läsa kapitel 17 i boken Eloquent JavaScript.

Det rekommenderas även starkt att ta en titt på följande källor:

- [JSON Tutorial](#)
- [AJAX: Getting started](#)
- [Using XMLHttpRequest](#)
- [.addEventListener\(\)](#)

## Uppgiften

Denna uppgift är indelad i tre steg, det rekommenderas därför att förhålla sig till ordningen av dessa. Tanken med uppgiften är att skapa ett sökningsformulär där en användare ska kunna söka på filmer och få resultatet av sökningen presenterad för sig som en lista. Det API vi kommer att använda för att hämta denna data är **OMDb** - detta



är en inofficiell API för att hämta data från IMDb.

Tips 1: testa gärna att klistra in er kod på **JSHint** för att se om den kan ge förslag på förbättringar. Det är en sida som följer vissa standarder för hur en bör skriva kod på ett korrekt vis.

Tips 2: testa gärna att använda er av tjänsten **Dirty Markup** som hjälper er formatera er kod.

**Glöm inte att läsa igenom kravspecifikation innan ni börjar, så ni inte missar någonting eller får komplettera på grund av slarvfel.**

## Arbetsgång

Nedan presenteras ett förslag till arbetsgång för uppgiften:

1. Börja med att göra er bekanta med **OMDb's API**
2. Hur använder vi ett API? Jo - genom en vanlig URL, pröva att besöka `http://www.omdbapi.com/?s=the%20revenant&type=movie` och se vad ni får tillbaka (observera att detta är i formatet JSON)
3. Kontrollera hur URL:en är uppbyggd, notera parametrarna `s` och `type`
4. Lägg till lyssnare för formuläret (som ni blir tilldelade i steg 1)
5. Gör så att när formuläret skickas så hämtar ni värdet i sökrutan (glöm inte att avbryta formuläret så det inte skickar iväg användaren, tips: `preventDefault()` )
6. Pröva och experimentera med `XMLHttpRequest` mot API:n (ni finner ett kodexempel i steg 2 kring detta)
7. Koppla samman så att när en användare skriver in ett sökord i formuläret så skickas en förfrågan mot API:et. Detta innebär att den URL ni använder er av för att kommunicera måste ändras beroende på vad en användare söker på, t.ex.
  - `http://www.omdbapi.com/?s=the%20revenant&type=movie` för "the reventant"
  - `http://www.omdbapi.com/?s=batman&type=movie` för "batman"
8. Gällande punkt 7 så är ett tips att placera all kod för själva kommunikationen mot API:et i en egen funktion, t.ex. `fetchMovies(movieTitle)` eller dylikt
9. När ni hämtar innehåll från API:et - börja med en `console.log()` för att se vad ni får tillbaka
10. Baserat på datan - presentera denna i div:en `<div id="result">` på valfritt vis

## Steg 1: Formulär



Ni kan utgå (valfritt) från följande HTML:

```
<!-- Sökformulär -->
<form action="" method="get" id="search-form">
  Movie: <input type="text" name="query">
  <button type="submit">Search</button>
</form>

<!-- Här placerar vi vårt resultat -->
<div id="result"></div>
```

Tänk på att hämta formuläret, lägga till ett event för `submit` och avbryta formuläret från att skicka användaren till en ny sida `preventDefault()` .

## Steg 2: AJAX

Nedan finner ni ett kodexempel för hur ni kan använda `XMLHttpRequest` för att göra en förfrågan mot API:et och sedan skriva ut resultatet i webbkonsollen. Denna kod kommer framöver behövas placeras i en funktion så att sökningar kan göras på olika filmtitlar.

```
// Objekt för att hantera AJAX
var omdbAPI = new XMLHttpRequest();
// Den URL vi ska använda oss av (obs. att denna har förvalts "th
var omdbURL = "http://www.omdbapi.com/?t=the%20revenant&type=mov

// Vad vill vi göra när vi fått ett svar?
omdbAPI.addEventListener("load", function() {
  // Konvertera resultatet från JSON
  var result = JSON.parse(this.responseText);
  // Skriv ut resultatet
  console.log(result);
});

// Ange vilken metod (get) samt URL vi ska skicka med
omdbAPI.open("get", omdbURL, true);
// Skicka förfrågan
omdbAPI.send();
```



## Steg 3: Presentation

I det sista steget ska ni koppla samman ert formulär med sökningen mot API:et. Observera att det är i `"load"` eventet som ni får tillbaka resultatet från API:et och således här ni behöver placera den kod som lägger till resultatet i er HTML.

Movie:

The Revenant (2015)

The Revenant (2009)

The Apostate: Call of the Revenant (2015)

A World Unseen: The Revenant (2016)

The Revenant (2012)

The Revenant (2013)

The Revenant of East Lansing (2016)

The Revenant (2009)

The Richy Show - Episode 13 - Revenant Upheaval (2014)

Visigoth: The Revenant King (2015)

1. Exempel på en sökning av en film.

## Kravspecifikation

- På första raden i er fil ska ni använda raden `"use strict";` för att göra webbläsaren mer strikt i sin tolkning av er kod
- Det får inte visas några felmeddelande i webbkonsollen
- All JavaScript-kod ska placeras i en egen fil och länkas in till ert HTML-dokument
- All JavaScript-kod måste ha godtycklig indentering, dvs. allting får inte vara vänsterställt eller allt för svårt att läsa

### Specifikt för uppgiften:

- Använd er av CSS för att göra sidan mer användbar (här är det ganska fritt men använd sunt förnuft så blir det bra)
- Om sökfältet är tomt ska det inte göras en förfrågan mot API:et, utan då kan ni, t.ex. använda er av `alert()` för att notifiera användaren om detta



- Om en användare gör ytterligare en sökning på en ny film så måste listan tömmas innan de nya resultaten visas
- Om sökningen inte gav något resultat ska ni notifiera användaren om detta på ett valfritt vis, `alert()` eller kanske ett meddelande i er HTML

## Inlämning

**Glöm inte kontrollera att ni skickat med svar på alla uppgifter och att ni följt kravspecifikationen.**

När du är färdig med din uppgift ska du ladda upp denna som en `.zip`-fil innehållande alla dina filer på Canvas (på samma sätt som inlämningsuppgift 1). Döp denna enligt formatet `inl4_Förnamn_Efternamn.zip`. DU ska även ladda upp dessa filer på dvwebb och därefter ska du även inkludera länken dit.

Lycka till!

---

Detta projekt är publicerat som **öppen källkod** på **GitHub**.

Användning av allt material är fritt att tillgå genom **The MIT License (MIT)**.