Rapport JSON & XML

Inledning

Vi har skapat en webbplats där användaren kan välja mellan att söka efter album, artist och filmer, där ett resultat presenteras på skärmen i form av en tabell. Vi valde att strukturera sidan med två startsidor, en där användaren kan söka efter musik och en sida där användaren söker efter filmer. Informationen som användaren söker efter hämtas från öppna API: er som hämtas hem med tekniken window fetch, i retur så får användaren tillbaka ett svar efter relevans till sökningen i form av en tabell. Vi har valt att visa en bild från album/film, titeln på albumet och filmen, och även vem som gjort albumet och vilket år filmen släpptes. Vi har använt oss av ramverket Bootstrap

för att skapa upp en responsiv hemsida, vi använde även Google fonts för att ta hem olika typsnitt och font awesome ikoner för extra styling.

Vi utgick ifrån de dataset som var angivna i uppgiften och valde två stycken dataset som skulle passa bra ihop med varandra, vi valde därför last FM:s API och OMDb:s API eftersom de båda förhåller sig till samma ämne. På Last FM:s API så kan man söka efter en filmtitel och få upp musiken som var med i filmen samt vem som skapat musiken. Sedan kan man söka efter band eller liknande på OMDb:s API och få fram dokumentärfilmer om bandet. Vi tyckte därför att båda API:erna knöts ihop på ett bra sätt och utgick därför ifrån de båda.

Analys

Vi har valt att arbeta med både JSON och XML. JSON – JavaScript Object Notation är ett dataformat baserat på syntaxen för objekt i Javascript. JSON används precis som XML för utbyte av data, genom att skicka data som ren text, text som kan förstås av människor.

Både JSON och XML kan läsas av människor, och är uppbyggda med en viss hierarki.

Skillnaden mellan JSON och XML

är den att i XML så måste man konvertera den hämtade strängen manuellt till XML-DOM. JSON är färdig konverterad och är redo att användas i ett färdigt JavaScript objekt. I JSON använder man inte sluttaggar som man gör i XML, JSON kan använda vektorer, som innehåller objekt med attribut, t ex text, bilder eller länkar. I XML får man tillbaka ett dokument med olika attribut.

XML ersätts mer och mer utav JSON som standard för datahantering på webben, men XML finns fortfarande kvar och används fortfarande till REST-API:er. XML används för listor, tabeller, trädstrukturer m.m.

Fördelen med JSON är att man inte behöver konvertera den hämtade strängen manuellt, I JSON har man ett globalt objekt som har metoder tillgängliga för att konvertera mellan JavaScript och JSON. JSON kan även ses som mer kompakt och lättare att läsa för de som är vana med Javascript, dels kan JSON vara snabbare än XML i överföringen av data eftersom de är färre data som överförs. JSON är idag det populäraste formatet för REST-API:er, därför är det en fördel att bemästra JSON.

Nackdelen med JSON är att inte alla datatyper stöds.

Fördelen med XML är att man får med ett schema med datatyper med strukturerad validering, och gör de möjligt att skapa nya datatyper. XML innehåller också XLST som överför data till olika output format. Originaldokumenten ändras inte, istället skapas ett nytt dokument som är baserat på de föregående. XML har inbyggt stöd för namnrymder som används för att lösa namnkonflikter i koden. XML är också mer likt HTML, så om man har erfarenhet och känner till HTML så kan de vara en fördel att använda XML.

Nackdelen med XML är att koden innehåller fler ord/kod i jämförelse med JSON, och att man får ut resultatet i mer data för samma mängd information.

Strukturerade dataset

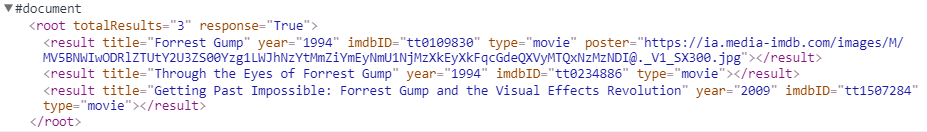
JSON

Strukturen på de två API:er vi använt oss ser olika ut då de presenteras på två olika sätt JSON och XML. När man använder fetch på last FM:s API så får man tillbaka ett objekt som innehåller ett objekt som kallas albummatches i det objektet ligger sedan ett till objekt som innehåller en vektor med 10 stycken objekt som innehåller information om sökningen. Säg att vi söker på nyckelordet ”Metallica”, det vi då får tillbaka är objektet albummatches och i det ligger album objektet som i sin tur innehåller en vektor med 10 stycken objekt, dessa objekt innehåller relevant information baserad på vår sökning. För att sedan presentera denna data så stegar man ner i strukturen för att hämta ut dessa 10 objekt och stoppar in deras information i html DOM:en.



XML

Den struktur man får tillbaka efter ett API anrop i XML form är ett dokument som innehåller elementen root och result. Båda innehåller attribut som innehåller information som vi sedan plockar ut, root elementet har attribute totalResults som anger antal träffar och response som då är en boolean som returnerar true och det gick bra eller false och det blev ett problem. Result elementet innehåller den information som vi presenterar i tabellen, title på filmen, år och en poster är dom tre attributen vi valt att hämta ut. Det man gör för att kunna presentera informationen i tabellen är att man tilldelar variabler informationen och sedan stoppar in variablerna i DOM:en, ungefär på samma sätt som JSON.



Användning av valda dataset

VI använder valda dataset för att hämta ut data för filmdata i XML, och musikdata i JSON. Vi använder dataseten för att presentera och möjliggöra data och information. Dataseten är öppna för alla, men på vissa öppna API:er krävs de att man är medlem och får fråga efter en API: nyckel, vilket vi var tvungna att göra på lastfm. Inom JSON använder vi dataseten för att komma åt artister och album. Dels så kan man söka fram vilket album eller vilken artist man vill komma åt genom en sökfunktion med ett formulär. Dels valde vi att presentera data med top 10 artister som skrobblas på last.fm just nu, som presenteras direkt på sidan. På XML sidan använder vi filmdata som dataset. Vi tänkte att de blir en bra sammanhållning där vi har en sida för filmdata och musikdata så kan man t.ex. söka på vilken film man vill veta mer om på Movie OMDb API:t och sedan söka på soundtracket för filmen på Last.fm. I XML använder vi datasetet på så sätt att man kan söka efter valfri film i en sökfunktion med ett sökformulär, där vi presenterar filmtiteln, de året filmen släpptes och filmomslaget.

Vår lösning är uppbyggd på de sättet att vi har använt oss utav ramverket Bootstrap för att skapa upp mallen för sidan, och för att snabbt kunna skapa en sida där vi presenterar vår data. Lösningen är uppbyggd med JSON och XML för att kunna komma åt datan från valda API:er. Där vi använder oss utav inbyggda metoder, och anrop. Vi kommer åt datan genom att göra ett fetch anrop mot <URL:erna> och med hjälp av en API: nyckel som behövs för att använda datan. Vi har även använt oss av CSS för att lösa vår uppgift.CSS använder vi oss främst för att styla upp sidan, men även för att få navigations-menyn att ha en fast position och att bakgrundstorleken ska bli oförändrad när man söker på en film eller artist.

Vi kom fram till vår lösning med hjälp av deltagande på föreläsningar, tidigare erfarenheter av HTML/CSS och JavaScript. Vi har granskat diverse kodexempel från föreläsningar som har varit relevanta till uppgiften, och deltagit på handledningar där vi fick en del hjälp och tips av Fredrik Johansson.

Källor

* <https://getbootstrap.com/> 2018-05-22

<https://fonts.google.com/> 2018-05-22

* <https://fontawesome.com/icons?d=gallery> 2018-05-22
* <https://www.last.fm/api/intro> 2018-05-22
* <http://www.omdbapi.com/> 2018-05-22
* <https://www.json.org/> 2018-05-22
* <https://sv.wikipedia.org/wiki/XML> 2018-05-22
* <https://kau.itslearning.com/> Fredrik Johanssons föreläsning om XML, JSON, och Dataformat 2018-05-22
* <https://javascript.info/async> 2018-05-22

Fredrik Björling & Rasmus Björklund ISGB-13 Portabla Format 2018-05-27.