

Spotify Search

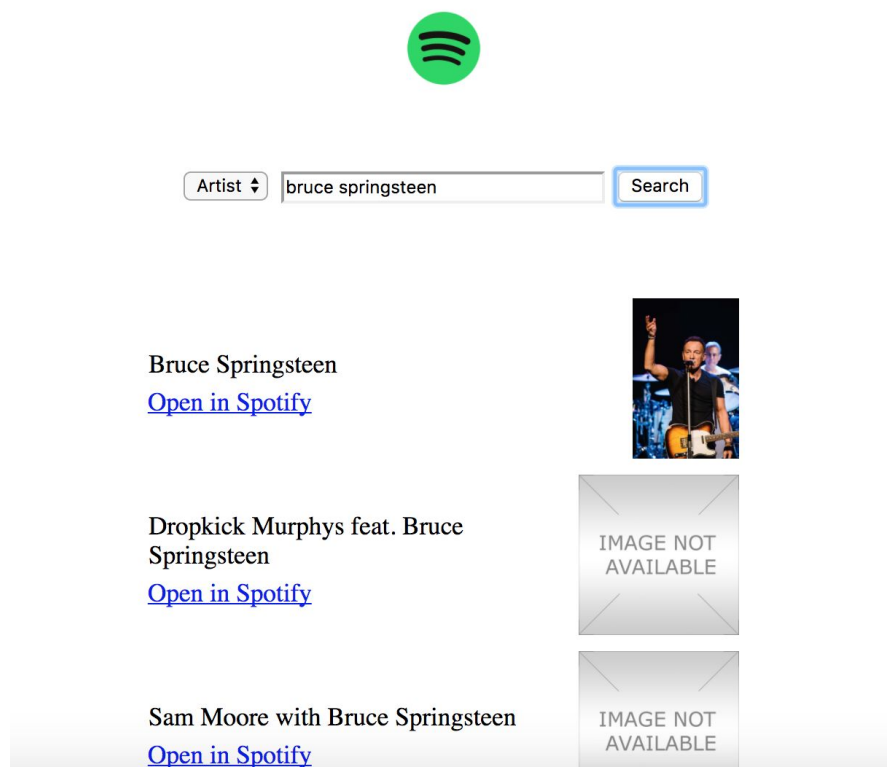
Denna övning syftar till att bygga en enkel webbsida med en frontend till Spotifys API för att söka efter låtar eller artister.

Funktionalitet

Webbsidan består av ett enkelt (Google-inspirerat) användargränssnitt som låter oss skriva in texten (namnet) på en artist eller låt. Sidan frågar därefter Spotify om att söka efter sådana objekt som matchar namnet och vi får som svar tillbaka 0 eller flera objekt som våra sökresultat.

Varje objekt visas sedan som en rad med ett namn, ett länk till att öppna objektet i Spotify samt en bild på objektet (eller en “not available”-bild om bild på objektet saknas).

Så här ser den färdiga webbsidan ut (efter en sökning):



Sökning

En Spotify sökning utförs att skicka en GET request till följande endpoint:

<https://api.spotify.com/v1/search>

På sidan Spotify Web API Console kan du testa denna request:

<https://beta.developer.spotify.com/console/get-search-item/>

För att utföra denna request krävs att du först loggar in på Spotify och därmed får en s.k. OAuth token. **Denna token går ut efter 1 timme**, varefter du måste gå in på Spotify Web API Console igen och åter logga in för att få en ny token.

(Givet att du har en loggat in och fått en token, ska denna senare användas i din kod när du utför en request)

Klicka sedan på knappen “FILL SAMPLE DATA”, som fyller i en exempelrequest. På raden “cURL Command” ser du hur denna request ser ut i “råformat”, däribland den kompletta URL:en samt HTTP headers som ingår.

Search for an Item

Description	Search for an Item docs
Endpoint	<code>https://api.spotify.com/v1/search</code>
HTTP Method	GET
OAuth	Required

Query *	<input type="text" value="Muse"/>
Item Type (album, artist, track, playlist) *	<input type="text" value="track,artist"/>
Market	<input type="text" value="US"/>
Limit	<input type="text" value="10"/>
Offset	<input type="text" value="5"/>
OAuth Token	<input type="text" value="BQB0jiCp0e59ddUsOnS2tyfhNlb95HaQqJdR5RJnG"/> CLEAR TOKEN

[TRY IT](#)[FILL SAMPLE DATA](#)

cURL Command

```
curl -X GET "https://api.spotify.com/v1/search?q=Muse&type=track,artist&market=US" -H "Accept: application/json" -H "Authorization: Bearer BQB0jiCp0e59ddUsOnS2tyfhNlb95HaQqJdR5RJnG0zug2MPgM_Nik26lMXGhERcUeKww2r7MIh5GMDv2Bm79SnKJ4TDXmX3YuC3jDX7f5k9IHu5cCCP9PwyWBL88vLlLLWfjlerVYA1LE30hHc"
```

[COPY](#)

I detta exempel söker vi alltså efter alla artister **och** låtar som har namnet “Muse” (observera: I UI:t för sökning ska användaren kunna välja artister **eller** låtar). Notera att din OAuth token används i en header som heter “Authorization”.

Klicka på knappen “TRY IT”; detta kör request:en och ger resultat i JSON-format.

```

{
  "artists" : {
    "href" : "https://api.spotify.com/v1/search?query=Muse&type=artist&market=US&offset=0&limit=20",
    "items" : [ {
      "external_urls" : {
        "spotify" : "https://open.spotify.com/artist/12Chz98pHFMPJEknJQMwvI"
      },
      "followers" : {
        "href" : null,
        "total" : 3463947
      },
      "genres" : [ "alternative rock", "modern rock", "permanent wave", "piano rock", "post-grunge", "rock" ],
      "href" : "https://api.spotify.com/v1/artists/12Chz98pHFMPJEknJQMwvI",
      "id" : "12Chz98pHFMPJEknJQMwvI",
      "images" : [ {
        "height" : 640,
        "url" : "https://i.scdn.co/image/19ac88c7aec1f68aa6e207aff29efa15d37336a7",
        "width" : 640
      }, {
        "height" : 320,
        "url" : "https://i.scdn.co/image/7ad2128db73a2814a1a96498404a5d9aabb4c15c",
        "width" : 320
      }, {
        "height" : 160,
        "url" : "https://i.scdn.co/image/f026a6204c28907e43e833eaa1820f9b674295ca",
        "width" : 160
      } ],
      "name" : "Muse",
      "popularity" : 79,
      "type" : "artist",
      "uri" : "spotify:artist:12Chz98pHFMPJEknJQMwvI"
    }, {
      "external_urls" : {
        "spotify" : "https://open.spotify.com/artist/0EcCExwuVReDp8Y86sLXUo"
      },
      "followers" : {
        "href" : null,
        "total" : 29
      },
      "genres" : [ ],
      "href" : "https://api.spotify.com/v1/artists/0EcCExwuVReDp8Y86sLXUo",
      "id" : "0EcCExwuVReDp8Y86sLXUo",
      "images" : [ {
        "height" : 640,
        "url" : "https://i.scdn.co/image/fef4944016086c9f0afb7ee943ab0a0d746b30e7",
        "width" : 640
      }, {
        "height" : 320,
        "url" : "https://i.scdn.co/image/39d74c5be6e3e929bdca315b2a4c60fc6c581487",
        "width" : 320
      }, {
        "height" : 160,
        "url" : "https://i.scdn.co/image/ae04510bb2b9493375de9e3748651e4574de4084",
        "width" : 160
      } ],
      "name" : "Charlee Muse",
      "popularity" : 34,
      "type" : "artist",
      "uri" : "spotify:artist:0EcCExwuVReDp8Y86sLXUo"
    } ],
  }, {

```

Här visas resultat för alla artister som matchar det sökta namnet. Bilden visar de två första sökresultaten/objekten i listan *artists.items*. För varje objekt ska följande properties läsas ut:

name

Det matchande namnet på artisten.

uri

Spotify-länken som användaren ska kunna klicka på för att öppna objektet i Spotify.

images

En lista av objekt, där varje objekt har en sträng vars värde är URL:en till en bild, relaterad till artisten.

Sökresultaten för låtar är nästan identiska, med ett litet undantag för var *images*-listan ligger i objektet (tips: gör en sökning för endast låtar - sätt "Item Type" till endast "track" istället för "artist,track", se bild ovan).

Listan *images* kan antingen finnas med i ett sökresultat eller ej; hantera detta enligt följande:

- Om listan ej finns ska en "not available" bild visas (det finns en konstant variabel med namnet `IMAGE_NA_URL` som ska användas).
- Om listan finns, använd den första URL:en i listan.

Om en sökning inte ger några resultat ska ett lämpligt meddelande (t.ex. "No search results found") visas. Om sökningen misslyckas (t.ex. om din OAuth token time:ar ut eller är fel) ska ett felmeddelande visas.

Kodtips

En given kodstruktur att utgå ifrån - läs kommentarerna i de respektive JS-filerna i mappen *src* för vägledning. Statisk HTML och CSS tillhandahålls redan via *index.html* och *style.css* i mappen *public*.

En byggmiljö saknas också; skapa en sådan genom att utgå från den byggmiljö du har skapat under en tidigare övning.

Några implementationstips:

- Studera hur exempelrequest:en ser ut i Spotify Web API Console för att bygga upp matchande request i din kod.
- Om ett fel inträffar då request:en utförs, läs ut HTTP statustext och använd som felmeddelande.
- Givet en lista av sökresultat ska en ny lista skapas, som endast innehåller de properties som ska användas (d.v.s. *name*, *uri* och *images*); denna nya lista ska användas för att bygga upp HTML:en för sökresultaten.

Obs: Om du inte har Spotify och därmed inte kan skapa en OAuth token, använd en kompis token eller sökresultaten i filen *searchresults.json*. Notera dock att för att klara övningen måste du först anropa Spotifys API och när du hanterar felet att du inte är auktoriserad att göra request:en, läs in och returnerna resultaten från *searchresults.json* istället.