



Martin Skiöld <martin.skiold11@gmail.com>

Jobb inom backend/full-stack

Per Rasmussen <per.rasmussen@cygni.se>
To: Martin Skiöld <martin.skiold11@gmail.com>

Mon, Nov 21, 2016 at 5:05 PM

Hej Martin,

Testet är teknikagnostiskt. Välj ditt bästa språk.

Här kommer hemtestet. Läs igenom hela beskrivningen. Det är helt ok att lämna in ett resonemang som lösning på några delar (exempelvis vad som skulle krävas för att få API:et produktionsfärdigt). Om något är oklart, så hör du av dig.

Lycka till,
//Per

Hemuppgift - backend - mashup

Denna uppgift går ut på att skapa ett REST API som helt enkelt erbjuder en mashup av några bakomliggande API:er. Vi vill att du skickar in din lösning senast en vecka efter att vi skickat uppgiften till dig.

De API:er som ska kombineras är [MusicBrainz](#), [Wikipedia](#) och [Cover Art Archive](#). MusicBrainz erbjuder ett API med bland annat detaljerad information om musikartister (information såsom artistens namn, födelseår, födelseland osv). Wikipedia är en community-wiki som innehåller beskrivande information om bland annat just musikartister. Cover Art Archive är ett systemprojekt till MusicBrainz som innehåller omslagsbilder till olika album, singlar eps osv som släppts av en artist.

Ditt API - själva hemuppgiften - ska ta emot ett [MBID \(MusicBrainz Identifier\)](#) och leverera tillbaka ett resultat bestående av

- En beskrivning av artisten som hämtas från Wikipedia. Wikipedia innehåller inte några MBID utan mappningen mellan MBID och Wikipedia-identifikatorer finns via MusicBrainz API.
- En lista över alla album som artisten släppt och länkar till bilder för varje album. Listan över album finns i MusicBrainz men bilderna finns på Cover Art Archive.

Externa API:er

MusicBrainz

- Dokumentation: http://musicbrainz.org/doc/Development/XML_Web_Service/Version_2
- URL till API: <http://musicbrainz.org/ws/2>
- Exempelfråga: <http://musicbrainz.org/ws/2/artist/5b11f4ce-a62d-471e-81fc-a69a8278c7da?&fmt=json&inc=url-rels+release-groups>

Wikipedia

- Dokumentation: https://www.mediawiki.org/wiki/API:Main_page
- URL till API: <https://en.wikipedia.org/w/api.php>
- Exempelfråga: [https://en.wikipedia.org/w/api.php?action=query&format=json&prop=extracts&exintro=true&redirects=true&titles=Nirvana_\(band\)](https://en.wikipedia.org/w/api.php?action=query&format=json&prop=extracts&exintro=true&redirects=true&titles=Nirvana_(band))

Hint: I JSON-svaret på exempelfrågan mot MusicBrainz ovan hittar ni en lista vid namn `relations`, kolla in relationen vars `type` är `wikipedia`. Där hittar ni det namn som kan användas för uppslag mot Wikipedias API nämligen `Nirvana_(band)`

Cover Art Archive

- Dokumentation: https://wiki.musicbrainz.org/Cover_Art_Archive/API
- URL till API: <http://coverartarchive.org/>
- Exempelfråga: <http://coverartarchive.org/release-group/1b022e01-4da6-387b-8658-8678046e4cef>

Hint: I JSON-svaret på exempelfrågan mot MusicBrainz ovan hittar ni en lista vid namn `release-groups`. Där finns bland annat ett albums titel (`title`) och dess MBID (`id`). Detta MBID används sedan för uppslaget mot Cover Art Archive.

Krav

Ditt API ska vara förberett för att kunna köras i produktion (om du inte hinner fixa allt - exempelvis felhantering osv så är det bra om du kan beskriva detta i lösningen). API:t måste kunna hantera hög last vilket kan vara utmanande eftersom de bakomliggande API:erna kan vara ganska långsamma och har konsumtionsbegränsningar. Vi vill att du levererar din lösning med tydliga instruktioner som beskriver hur vi ska installera och köra lösningen.

Tänk på din lösning som en startpunkt för ett större projekt som andra ska kunna bygga vidare på. Dokumentera gärna om något saknas eller om du tar genvägar. Använd hellre tekniker som du känner till än något nytt. Vi bedömer kodkvalité, struktur och förväntar oss att du kan motivera de val du gjort.

- API:et ska vara **REST-baserat**
- Som transportformat ska **JSON** användas.
- Ditt resultat - som innehåller beskrivning och album - ska vara ett JSON-svar.

Ett svar från er tjänst kan se ut något i stil med följande:

```
{
  "mbid" : "5b11f4ce-a62d-471e-81fc-a69a8278c7da",
  "description" : "<p><b>Nirvana</b> was an American rock band that was formed ... osv
osv ... ",
  "albums" : [
    {
      "title" : "Nevermind",
      "id": "1b022e01-4da6-387b-8658-8678046e4cef",
      "image": "http://coverartarchive.org/release/a146429a-cedc-3ab0-9e41-1aaf5f6cdc2d/3012495605.jpg"
    },
  ],
}
```

```
{
  ... more albums...
}
]
```

Java

- Om du bygger en javalösning ska [Apache Maven](#) eller [Gradle](#) användas för att bygga och paketera API:et.
- API:et ska kunna startas direkt från Maven/Gradle (exempelvis genom att [Jetty](#) och `mvn jetty:run`) eller som en executable jar [via exempelvis Spring Boot](#).
- Exempel på möjliga ramverk: SpringMVC ([Spring Boot](#)), [Dropwizard](#), [Jersey](#)

.NET

- Om du bygger en .Net-lösning skall en Visual Studio 2015-solution användas.
- API:et skall kunna startas direkt från Visual Studio.

JavaScript - Node.js

- Kika gärna på ramverk såsom [Hapi](#), [Koa](#) eller [Express](#)
- ECMAScript 6 är något som vi gillar, kolla exempelvis in [Babel](#)
-

Per Rasmussen

Konsultchef, Cygni

Tel: 070-947 21 50

Epost: per.rasmussen@cygni.se



[Quoted text hidden]