# Hemuppgift-backend-mashup

Datum: 2016-11-28

Av: Martin Skiöld

## Beskrivning av lösning

Lösningen är skriven i Java med IntelliJ IDEA som IDE.

Lösningen använder Spring MVC som ramverk och byggs med Maven.

Källkoden är strukturerad enligt följande:  
  
Controllers/[API:ets controllers som hanterar http requests]  
Exceptions/[Skräddarsydda exceptions för felhantering]

Models/[Data]

Services/[API:ets logik, sammanfogar information från olika API:er]\*

\* API:ets logik körs på flera trådar för att minska dötid på tråden i controllern.

OBS!

Jag har nästintill obefintlig tidigare erfarenhet av Maven och Spring MVC. Jag är därför inte helt övertygad huruvida lösningen kommer installeras korrekt på ett annat operativsystem (Jag kör Mac OS X El Capitan). Därför har jag också skickat med källkoden.

## Installation

Cd:a till projektets root-folder:

$ mvn clean install

## Exekvering

$ cd target

$ java –jar cygni\_backend\_martinskiold-0.0.1-SNAPSHOT

## Avsluta exekvering

Tryck ctrl-c i terminalen/kommandotolken \*

\* Kommandot skickar SIGINT-signal och avslutar exekveringen

## Användning av API:t

Request görs till följande url:  
<http://localhost:8080/artist>

Query-parametrar:  
\* mbid (Artistens MusicBrainz identifierare)

Exempel på request:  
<http://localhost:8080/artist?mbid=cc197bad-dc9c-440d-a5b5-d52ba2e14234>

## Felhantering

API:t hanterar följande fel:

* Om query-parameter inte är angiven svarar tjänsten med 400 Bad Request
* Om mbid:t inte motsvaras av en artist på MusicBrainz svarar tjänsten med 404 Not Found
* Avsaknad av albumbilder eller wikipediabeskrivning anges som ”N/A” (Not available) i JSON-svaret.

## Förbättringar

* Felhantering hade kunnat göras bättre. Speciellt tror jag att man kan skicka JSON-felmeddelanden och samtidigt http error message vid påträffande av fel.
* Ett mindre problem är att 404 Not Found-meddelandet i nuläget visar den utlösande exception-klassen för användaren - vilket är onödigt.
* Det hade varit bra om API:t kunde hantera fler typer av requests. I nuläget stöds endast GET-request. Exempelvis hade det varit bra om API:t stödjer OPTIONS-request så att andra programmerare enklare kan ta del av API:t.
* För att snabba på återkommande sökningar vid hög belastning hade man kunnat cache:a sökningar i en databas. Speciellt är det information från MusicBrainz API som skapar störst väntetider och det är därför dennes information som har störst behov av cachning.