

Mashup projekt

MDK Mashup | Max Mildh | Daniel Hägerström | Kennet Szeszowicki

Syfte

Vi har valt att ta fram en tjänst som erbjuder information om bl.a. filmer, serier och dokumentärer. Den information som presenteras innehåller metadata om filmen och en trailer. Vårt mål med tjänsten är att den skall vara lätt att använda och lättöverskådlig. Därav satsade vi på en avskalad design vars mål i första hand är att presentera ett relevant urval av data för användaren. För att åstadkomma det har vi aktivt valt att inte presentera viss metadata om filmen. Data som vi ansett vara mindre relevant för användaren. Vi har också valt att sökresultatet ska vara det centrala innehållet på startsidan. Även det för att åstadkomma en lätt och överskådlig tjänst för användaren.

Målet med det API vi skapat följer samma mål som vår tjänst. Alltså att det ska vara lätt att använda och förstå. På så vis skapade vi ett API med ett tydligt upplägg där användaren får möjlighet att testa API:et i ett webbgränssnitt.

Designval och tekniska lösningar

Filmtjänst

Vi har valt att skapa vår back-end med hjälp av Python med Flask som ramverk. Valet av Python föll sig naturligt då alla i projektgruppen arbetat med det tidigare. Fördelarna med Python är att det är enkelt och snabbt att utveckla i. Bland annat är syntaxen enkel att arbeta med.

Anledningen till att vi använde Flask istället för liknande ramverk som bl.a. Bottle är för att det finns tydligare dokumentation att tillgå. Något som bl.a. förenklade kommunikationen med vår front-end. Dessutom framstod Flask som ett mer utvecklat ramverk med flera funktioner som vi kunde få användning av.

Som front-end använder vi förutom vanlig HTML och CSS även AngularJS och JavaScript. Valen av HTML och CSS blev självklara då vi utvecklade en webbtjänst och webben drivs i grunden av de två. AngularJS och JavaScript har vi använt för att skapa en interaktiv miljö. AngularJS valde vi för att få en mer utmanande process då språket var nytt för samtliga i projektgruppen. Tanken i valet av AngularJS var att minska andelen JavaScript-funktioner som vi annars hade varit tvungna att skriva själva, och genom det skapa en smidigare utveckling och tydligare kod.

Den tekniska processen

En förfrågan görs mot vår server och Python svarar med en renderad HTML-template. Inuti det template finns det AngularJS-kod som tillkallar JavaScript-kod och det kommunicerar med vår Python-back-end. Det hämtar i sin tur data från ett eller flera API:er. AngularJS presenterar sedan datan på webbsidan.

API

Grundläggande ville vi att de API:er vi arbetade emot skulle ge oss svar i JSON. Då vi uppfattade JSON som enklare att använda.

De API:er vi använder är OMDB API och My API Films. OMDB använder vi då det är ett öppet och enkelt API för att komma åt filminformation utan begränsningar för förfrågningar. Det ger dessutom ett svar i JSON.

My API Films hämtar sin data från Trailer Addict som tillhandahåller en samling med trailers. Valet av My API Films istället för att direkt använda Trailer Addict var för att My API Films lät oss få ett svar i

JSON. Något vi var mer bekväma med samt förenklade utvecklingen då det skapade samma tekniska specifikationer över båda API:erna. Nackdelen är dock att det är ganska långsamt.

URL designen följer de principer som är betraktade enligt praxis. Exempel inom vårt API använder vi movie/ID för att hämta hem information om en given film. Tanken här är hur vi logiskt letar information, därav är det verbet i singular. Då endast information om en film visas. I vår movie endpoint har vi valt att tvätta data ifrån OMDb och My API Films samt slå ihop dem för att förutom information om filmen också få ut en länk till filmens trailer.

För API-dokumentation valde vi Swagger eftersom de lät oss skapa en demo av vår dokumentation i deras gränssnitt. Genom det kunde vi senare exportera koden i JSON och på så sätt skapa vårt eget gränssnitt av dokumentationen där man bl.a. kan testa API: et. Swagger använder YAML, vilket är ett språk som använder en syntax som liknar bl.a. Python, vilket projektgruppen är bekväma med. YAML är likt XML ett språk som kommunicerar med en dokumentmall, i detta fall Swaggers. Vilket kan konverteras till JSON.

Databehandling

I vår kommunikation med OMDb och My API Films har vi valt att endast presentera viss data. Den data som presenteras är bearbetad för att visa det vi anser mest relevant för användaren.

Ett exempel JSON-svar från OMDb:

```
{
  "Actors": "Tom Hardy, Charlize Theron, Nicholas Hoult, Hugh Keays-Byrne",
  "Awards": "1 win & 3 nominations.",
  "Country": "Australia, USA",
  "Director": "George Miller",
  "Genre": "Action, Adventure, Sci-Fi",
  "Language": "English, Russian",
  "Metascore": "89",
  "Plot": "A woman rebels against a tyrannical ruler in post apocalyptic Australia in search for her homeland with the help of a group of female prisoners, a psychotic worshiper, and a drifter named Max.",
  "Poster": "http://ia.media-imdb.com/images/M/MV5BMTUyMTE0ODcxNF5BMl5BanBnXkFtZTgwODE4NDQzNTE@._V1_SX300.jpg",
  "Rated": "R",
  "Released": "15 May 2015",
  "Response": "True",
  "Runtime": "120 min",
  "Title": "Mad Max: Fury Road",
  "Type": "movie",
  "Writer": "George Miller, Brendan McCarthy, Nick Lathouris",
  "Year": "2015",
  "imdbID": "tt1392190",
  "imdbRating": "8.3",
  "imdbVotes": "345,918"
}
```

Ett exempel JSON-svar från My API Films:

```
{
  "trailer": [{"embed": "<iframe width=\\"480\\" height=\\"270\\" src=\\"//v.traileraddict.com/105892\\" allowfullscreen=\\"true\\" webkitallowfullscreen=\\"true\\" mozallowfullscreen=\\"true\\" scrolling=\\"no\\" frameborder=\\"0\\"></iframe>\n <p><a href=\\"http://www.traileraddict.com/mad-max-fury-road/featurette-tanker-stunt\\">Featurette - Tanker Stunt</a> for <a"}]
```

```
href="\http://www.traileraddict.com/mad-max-fury-road\">Mad Max: Fury Road</a> on <a
href="\http://www.traileraddict.com\">TrailerAddict</a>.</p> ",
"imdb":"1392190",
"link":"http://www.traileraddict.com/mad-max-fury-road/featurette-tanker-stunt",
"pubDate":"Wed, 30 Sep 2015 12:53:45 -0700",
"title":"Mad Max: Fury Road: Featurette - Tanker Stunt",
"trailer_id":"105892"}}
}
```

Vad vi väljer att visa i vårt API genom att slå ihop information ifrån OMDb och My API Films:

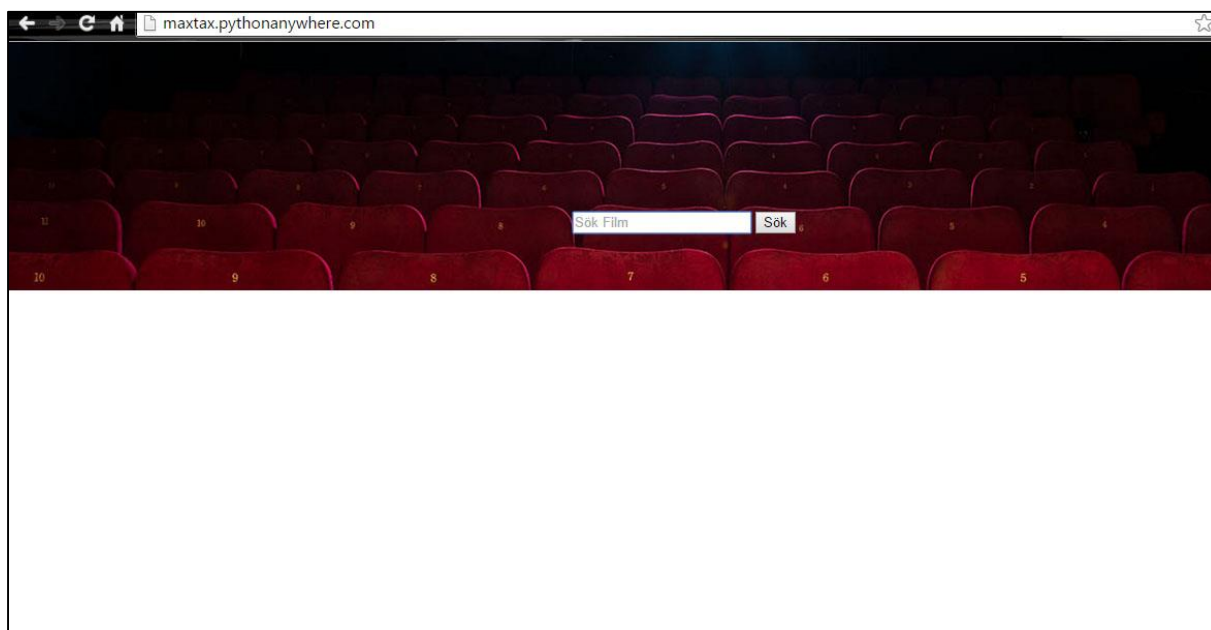
```
{
  "Actors": "Tom Hardy, Charlize Theron, Nicholas Hoult, Hugh Keays-Byrne",
  "Genre": "Action, Adventure, Sci-Fi",
  "Plot": "A woman rebels against a tyrannical ruler in post apocalyptic Australia in search for her
homeland with the help of a group of female prisoners, a psychotic worshiper, and a drifter named
Max.",
  "Runtime": "120 min",
  "Title": "Mad Max: Fury Road",
  "Writer": "George Miller, Brendan McCarthy, Nick Lathouris",
  "imdbID": "tt1392190",
  "imdbRating": "8.3",
  "trailer_embed": "<iframe width=\"480\" height=\"270\" src=\"v.traileraddict.com/105892\"
allowfullscreen=\"true\" webkitallowfullscreen=\"true\" mozallowfullscreen=\"true\"
scrolling=\"no\" frameborder=\"0\"></iframe>\n <p><a href=\"http://www.traileraddict.com/mad-
max-fury-road/featurette-tanker-stunt\">Featurette - Tanker Stunt</a> for <a
href=\"http://www.traileraddict.com/mad-max-fury-road\">Mad Max: Fury Road</a> on <a
href=\"http://www.traileraddict.com\">TrailerAddict</a>.</p> ",
  "trailer_link": "http://www.traileraddict.com/mad-max-fury-road/featurette-tanker-stunt"
}
```

Det finns även en endpoint i vårt API som heter search där användaren kan söka efter en film via dess titel. Vi har här valt att inte tvätta datan utan tunnlar endast ifrån det ursprungliga API:et, alltså OMDb's API.

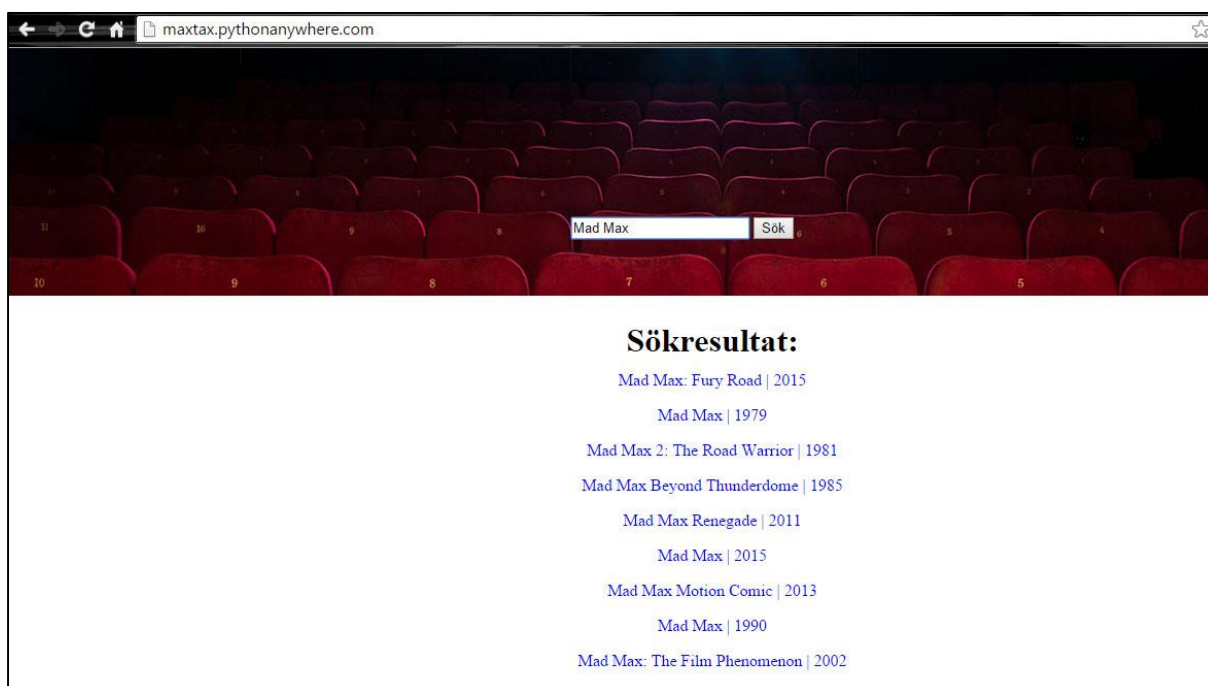
Användarmanual

I sökfältet skriver du in vilken film du vill söka efter.

<http://maxtax.pythonanywhere.com>



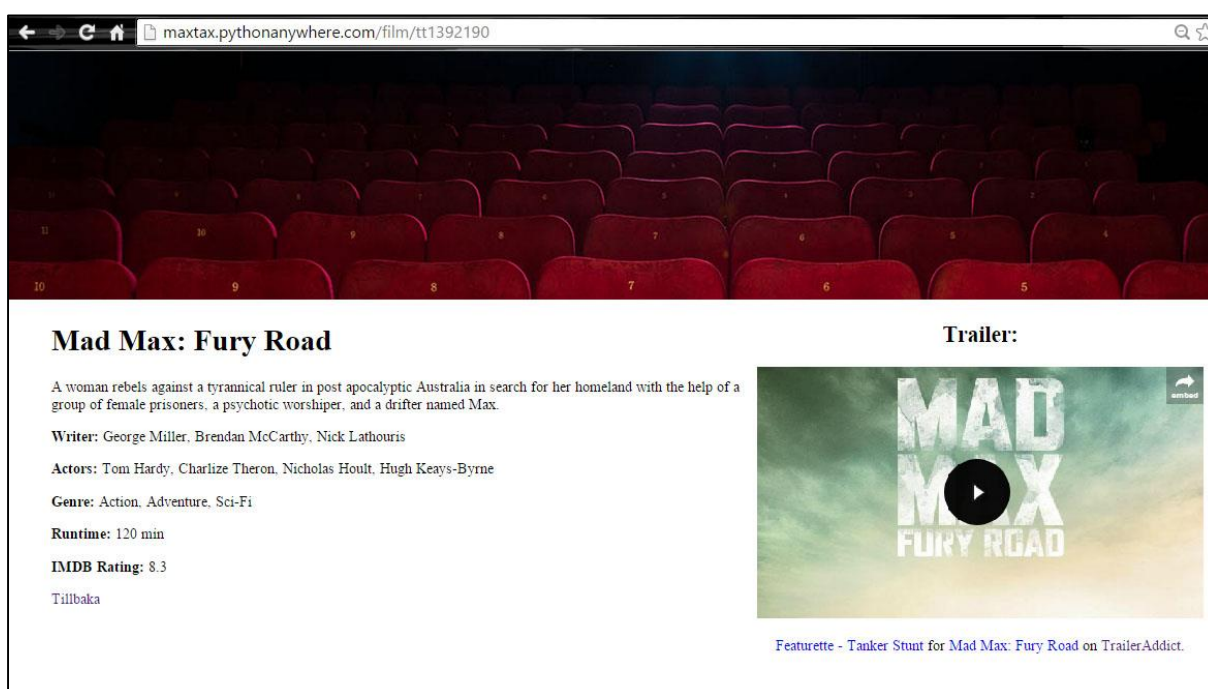
Sedan presenteras de sökresultat som hittas. Hittades inga filmer blir du meddelad om detta.



Därefter väljer du det resultat som passar dig bäst och blir omdirigerad till en sida med mer utförlig information om ditt val. Där du bl.a. kan se filmens handling samt en trailer (i vissa fall finns det ingen trailer kopplad till filmen). Observera också att laddningstiderna ibland kan vara långa.

Skulle du välja Mad Max: Fury Road | 2015 blir du omdirigerad till följande URL:

<http://maxtax.pythonanywhere.com/film/tt1392190>



Andra exempel på filmer där trailer finns:

<http://maxtax.pythonanywhere.com/film/tt0083866>

<http://maxtax.pythonanywhere.com/film/tt0468569>

<http://maxtax.pythonanywhere.com/film/tt2310332>

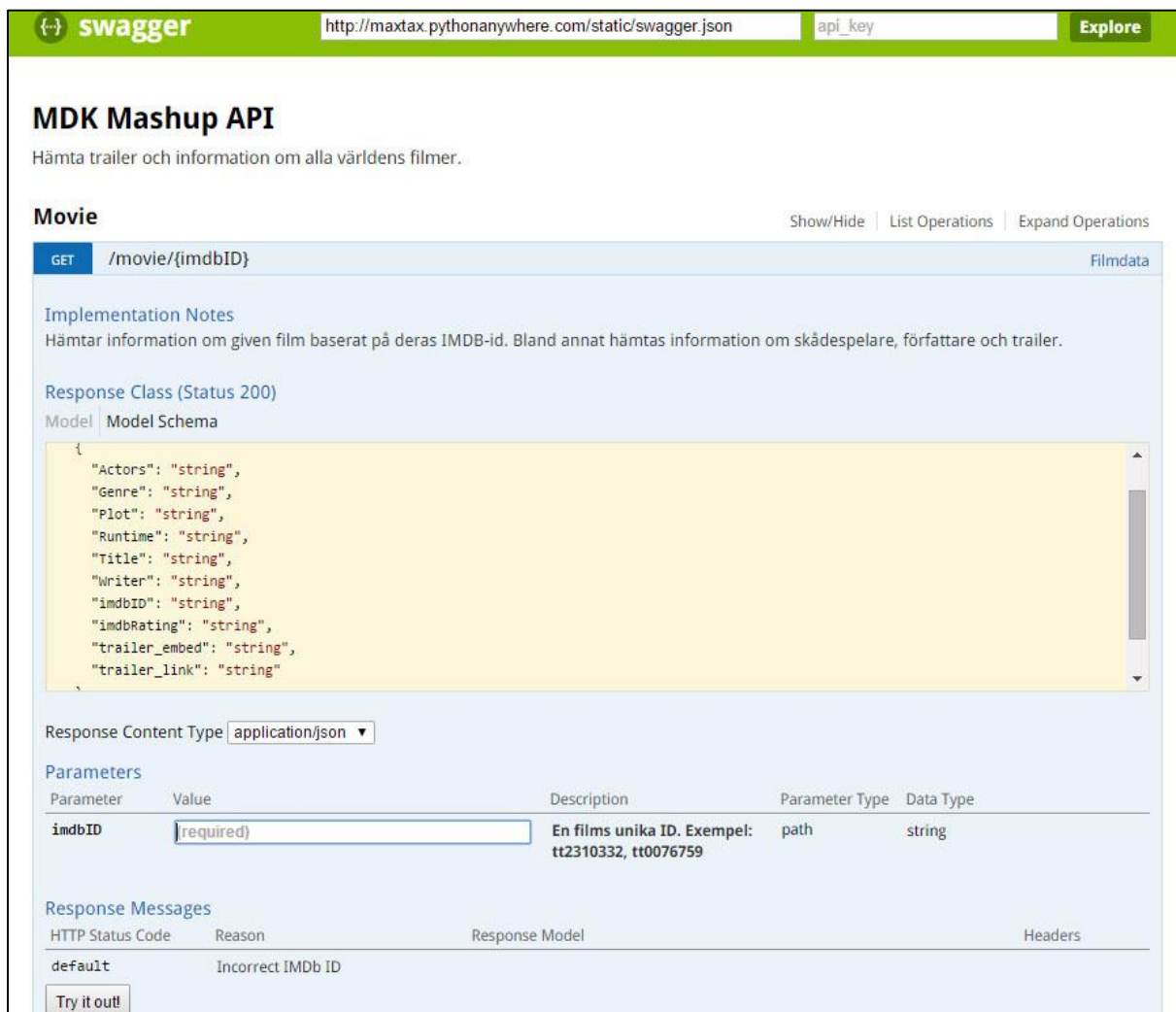
API

För API:ets dokumentation, manual och exempel se:

<http://maxtax.pythonanywhere.com/static/docs/index.html>

Här får du instruktioner hur du kan testköra API:et. Det finns även några test ID att använda sig av.

Se nedan:



The image shows the Swagger UI for the MDK Mashup API. The interface is in Swedish and displays the API's endpoints, parameters, and response schemas. The main header shows the Swagger logo, the API URL, and an 'Explore' button. The title 'MDK Mashup API' is followed by a description: 'Hämta trailer och information om alla världens filmer.' The 'Movie' section is active, showing the endpoint '/movie/{imdbID}' with a GET method. The 'Implementation Notes' section states: 'Hämtar information om given film baserat på deras IMDB-id. Bland annat hämtas information om skådespelare, författare och trailer.' The 'Response Class (Status 200)' section shows a JSON schema for the response, including fields like 'Actors', 'Genre', 'Plot', 'Runtime', 'Title', 'Writer', 'imdbID', 'imdbRating', 'trailer_embed', and 'trailer_link'. The 'Parameters' section shows a table with one parameter: 'imdbID' (required), which is a path parameter of type 'string'. The 'Response Messages' section shows a table with one message: 'default' (Incorrect IMDB ID). A 'Try it out!' button is visible at the bottom left.

swagger <http://maxtax.pythonanywhere.com/static/swagger.json> **Explore**

MDK Mashup API

Hämta trailer och information om alla världens filmer.

Movie

Show/Hide | List Operations | Expand Operations

GET /movie/{imdbID} Filmdata

Implementation Notes
Hämtar information om given film baserat på deras IMDB-id. Bland annat hämtas information om skådespelare, författare och trailer.

Response Class (Status 200)
Model | Model Schema

```
{
  "Actors": "string",
  "Genre": "string",
  "Plot": "string",
  "Runtime": "string",
  "Title": "string",
  "Writer": "string",
  "imdbID": "string",
  "imdbRating": "string",
  "trailer_embed": "string",
  "trailer_link": "string"
}
```

Response Content Type:

Parameters

| Parameter | Value | Description | Parameter Type | Data Type |
|-----------|---|--|----------------|-----------|
| imdbID | <input type="text" value="{required}"/> | En films unika ID. Exempel: tt2310332, tt0076759 | path | string |

Response Messages

| HTTP Status Code | Reason | Response Model | Headers |
|------------------|-------------------|----------------|---------|
| default | Incorrect IMDB ID | | |

För prova att söka genom vårt API skriver du in det du är ute efter i sökfältet. Se nedan:

The image shows the Swagger UI for the MDK Mashup API. The top bar is green with the Swagger logo, the URL `http://maxtax.pythonanywhere.com/static/swagger.json`, an `api_key` field, and an **Explore** button.

The main title is **MDK Mashup API** with the description "Hämta trailer och information om alla världens filmer."

There are two sections: **Movie** and **Search**. The **Search** section is expanded, showing the `GET /search?q={search}` endpoint. It includes a "Filmdata" link, implementation notes, a response class (Status 200), and a model schema.

Implementation Notes: Hämtar information om filmer baserat på söksträng.

Response Class (Status 200): Model | Model Schema

```
[
  {
    "Poster": "string",
    "Title": "string",
    "Type": "string",
    "Year": "string",
    "imdbID": "string"
  }
]
```

Response Content Type: `application/json`

Parameters:

| Parameter | Value | Description | Parameter Type | Data Type |
|-----------|---|--------------|----------------|-----------|
| search | <input type="text" value="([required)]"/> | En söksträng | path | string |

Response Messages:

| HTTP Status Code | Reason | Response Model | Headers |
|------------------|------------------|----------------|---------|
| default | Movie not found! | | |

At the bottom, there is a "Try it out!" button.

Instruktioner för körning av filmtjänst

Det finns två val att välja. Lokalt eller externt. Vi har lagt upp tjänsten så att den kan nås externt via <http://maxtax.pythonanywhere.com/>.

Om den skall köras lokalt behövs Python 2.7, Flask, Flask-CORS och Flask-triangle dessa installeras via `pip install flask flask-cors flask-triangle`. I root-mappen körs sedan `hello.py` med Python.