

Opdracht 3 - Notflix



Casus:

Eenvoudige movie-rating service

Beschrijving:

Een stel vrienden besluit op een avond de movie-club "NotFlix" te beginnen. Als lid van de club is het zaak om een aantal vooraf gedefinieerde films te bekijken en te beoordelen(raten).

Met gebruik van:

- Jersey
- JAXB
- Jackson
- JAX-RS

Begeleidende docent:

Paul Goolkate

Door:

| <i>Naam</i> | <i>Email</i> | <i>Studentnummer</i> | <i>Klas</i> |
|--------------------|--------------------------|-----------------------------|--------------------|
| Marco Jansen | Jansen.marco95@gmail.com | 345864 | EIN2vB |
| Daan Veldhof | Daanveldhof@gmail.com | 357852 | EIN2vB |

Inhoudsopgave

| | |
|--|----------|
| Verantwoording/Ontwerp | 3 |
| <i>Requests</i> | 3 |
| 1. Movies | 3 |
| 2. Users | 3 |
| 3. Rating | 4 |
| <i>Gebruikte statuscodes</i> | 5 |
| Koppeling tussen domein en REST-service | 6 |
| Testverslag | 7 |
| <i>War file</i> | 10 |
| <i>Beveiliging</i> | 11 |

Verantwoording/Ontwerp

Requests

1. Movies

1. GET movies

URL /movies

Parameters: Geen

Token nodig: Nee

Response: Lijst met films (statuscode = 200)

2. GET rated movies

URL /movies/rated

Parameter: Geen

Token nodig: Ja

Response: Lijst met gerankte films en gemiddelde rating. (statuscode = 200)

Response vals token: Null (Statuscode = 403)

3. GET unrated movies

URL /movies/unrated

Parameter: Geen

Token nodig: Ja

Response: Lijst met films die niet gerankt zijn (statuscode = 200)

Response vals token: Null (Statuscode = 403)

2. Users

1. POST user (Register)

URL /users

Parameters: Achternaam, Tussenvoegsels, Voornaam, Nickname en wachtwoord

Token nodig: Nee

Response: user token (statuscode = 201)

Response account bestaat al: Null (statuscode = 406)

2. POST login (Login)

URL /users/login

Parameters: Nickname, wachtwoord

Token nodig: Nee

Response: User token (statuscode = 200)

Response foute account: Null (statuscode = 403)

3. GET users

URL /users

Parameters: Geen

Token nodig: Ja

Response: Lijst met users (statuscode = 200)

Response vals token: Null (statuscode = 403)

4. GET user

URL /users/{nickname}

Parameters: Geen

Token nodig: Ja

Response: Een user object. (statuscode = 200)

Response vals token: null + (statuscode = 403)

Response nickname bestaat niet: Null (statuscode = 406)

3. Rating

1. POST rating

URL /ratings

Parameters: IMDb ID(Van film), Rating (double)

Token nodig: Ja

Response: (statuscode = 201)

Response vals token: (Statuscode = 403)

Response rating bestaat al: (statuscode = 406)

2. PUT rating

URL /ratings

Parameters: IMDb ID(Van film), Rating (double)

Token nodig: Ja

Response: (statuscode = 202)

Response vals token: (statuscode = 403)

Response rating bestaat niet: (statuscode = 304)

3. DELETE rating

URL /ratings

Parameter: IMDb ID(van film)

Token nodig: Ja

Response: (statuscode = 202)

Response vals token: (statuscode = 403)

Response rating bestaat niet: (statuscode = 406)

4. GET rating

URL /ratings

Parameter: Geen

Token nodig: Ja

Response: Lijst met films die de opvrager (Token) gerated heeft en bij elke film je gegeven rating.(statuscode = 200)

Response vals token: Null (statuscode = 403)

5. GET unrated

URL /ratings/unrated

Parameter: Geen

Token nodig: Ja

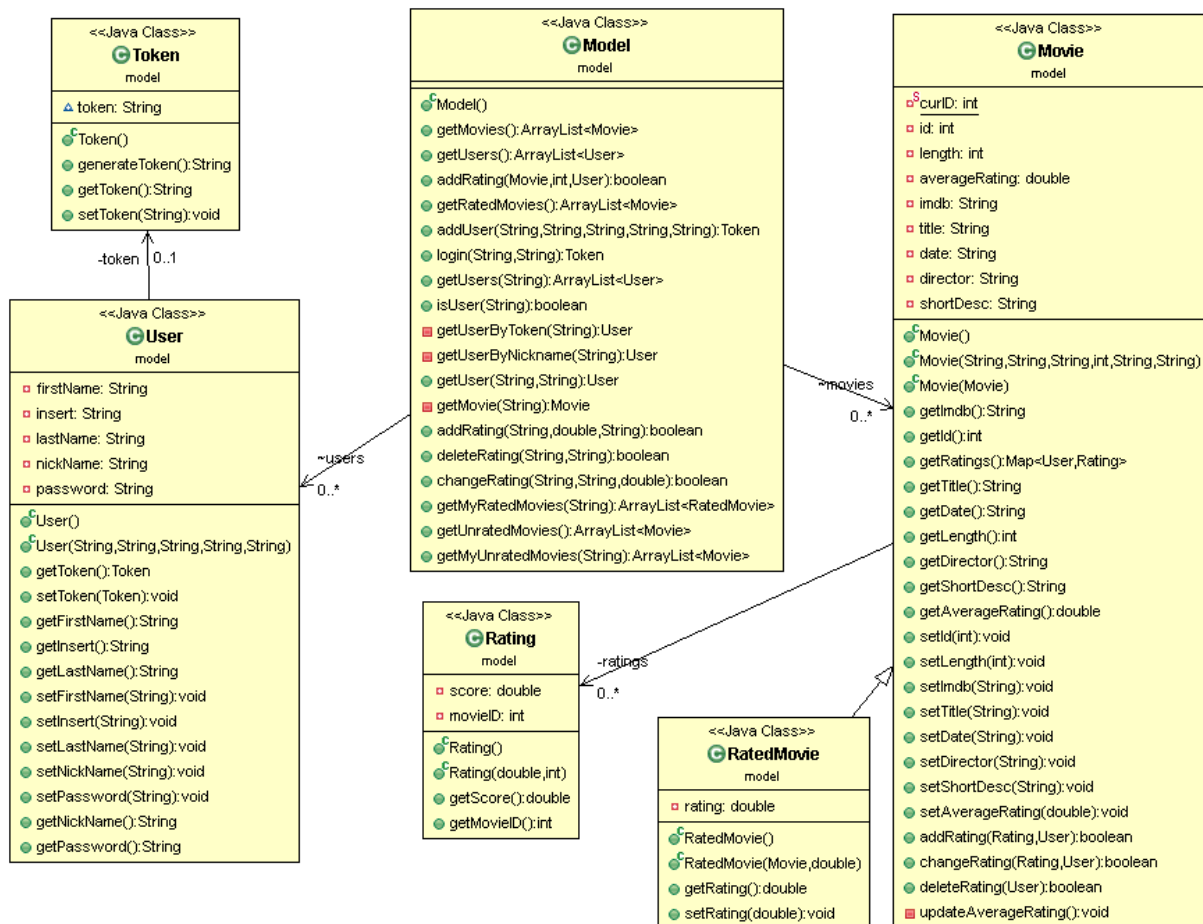
Response: Lijst met films die de opvrager(Token) nog niet gerated heeft. (statuscode = 200)

Response vals token: Null (statuscode = 403)

Gebruikte statuscodes

- 200 - OK
- 201 - Created
- 202 - Accepted
- 304 - Not modified
- 403 - Forbidden
- 406 - Not acceptable

Koppeling tussen domein en REST-service



We hebben er voor gekozen dat het Model een lijst met User objecten en Movie objecten heeft. Hierdoor kan eenvoudig alle Users of alle Movies op gevraagd worden.

Een Movie heeft een HashMap waarbij een User de sleutel is en Rating de waarde. Hierdoor kan een User maar 1 rating doen bij een film. We hebben gekozen voor een Map omdat dit de User direct linkt aan een Rating. Door een Map te gebruiken wouden we de redundantie ook zo laag mogelijk houden.

De RatedMovie wordt gebruikt als de User een overzicht wil waar alle films worden weergegeven met de rating die de User heeft gedaan.

Een User heeft één Token die gebruikt wordt om de User te identificeren. Als de User correct heeft ingelogd krijgt hij een Token van de web service die bij het opstarten van de server gegenereerd wordt. Deze moet hij elke keer dat hij een request doet meegeven in de header van de request. Door een token te gebruiken wordt er niet steeds een nickname en password combinatie over het internet verstuurd.

Testverslag

Al onze tests zijn uitgevoerd met de header : Accept application/xml. Ook zijn deze tests uitgevoerd met Accept application/json en dit leverde de bijbehorende json zonder problemen. We hebben in ons testverslag ervoor gekozen om alleen xml te laten zien. Er staan standaard 2 films in ons model en hebben we bij het testen van onderdelen die te maken hebben met rating een rating toegevoegd. (m.b.v. Post rating)

1.1 Get movies

Verwachte response: Lijst met films (statuscode=200)
Response: 200

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<movies>
  <movie imdb="1234">
    <averageRating>0.0</averageRating>
    <date>2012</date>
    <director>Steven Seagal</director>
    <length>138</length>
    <shortDesc>Deze film is een serie</shortDesc>
    <title>Breaking Bad</title>
  </movie>
  <movie imdb="1235">
    <averageRating>0.0</averageRating>
    <date>2013</date>
    <director>Steven Seagal</director>
    <length>140</length>
    <shortDesc>Dit is een test</shortDesc>
    <title>Breaking Good</title>
  </movie>
</movies>
```

1.2 Get rated movies

Verwachte response: Lijst met gerankte films (statuscode= 200)
Response(Na 1x post rating gedaan te hebben): 200

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<movies>
  <movie imdb="1234">
    <averageRating>8.0</averageRating>
    <date>2012</date>
    <director>Steven Seagal</director>
    <length>138</length>
    <shortDesc>Deze film is een serie</shortDesc>
    <title>Breaking Bad</title>
  </movie>
</movies>
```

1.3 Get unrated movies

Verwachte response: Lijst met ongerankte films(statuscode = 200)

Response(Na 1x post rating gedaan te hebben): 200

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<movies>
  <movie imdb="1235">
    <averageRating>0.0</averageRating>
    <date>2013</date>
    <director>Steven Seagal</director>
    <length>140</length>
    <shortDesc>Dit is een test</shortDesc>
    <title>Breaking Good</title>
  </movie>
</movies>
```

2.1 Register

Verwachte response: Token (statuscode = 201)

Response: (201)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<token>
  <token>5KRF7OI1768dPFkThY0RUoLo</token>
</token>
```

2.2 Login

Verwachte response: Token (statuscode = 200)

Response: (200)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<token>
  <token>1Za2AHYHoFe8U81b1uuXm8qn</token>
</token>
```

2.3 Get users

Verwachte response: Lijst met users. (statuscode = 200)

```
Response: (200) <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<users>
  <user>
    <firstName>marco</firstName>
    <lastName>jansen</lastName>
    <nickName>marco</nickName>
  </user>
  <user>
    <firstName>daan</firstName>
    <lastName>veldhof</lastName>
    <nickName>daan</nickName>
  </user>
  <user>
    <firstName>marco</firstName>
    <lastName>jansen</lastName>
    <nickName>marco0546</nickName>
  </user>
</users>
```


2.4 Get user

Verwachte response: Een user met de gevraagde nickname. (statuscode = 200)

Response: (200)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
▼<user>
  <firstName>marco</firstName>
  <lastName>jansen</lastName>
  <nickName>marco</nickName>
</user>
```

3.1 Post rating

Verwachte response: (statuscode = 201)

Response: 201

3.2 Put rating

Verwachte response: (statuscode = 202)

Response: 202

3.3 Delete rating

Verwachte response: (statuscode = 202)

Response: 202

3.4 Get rating

Verwachte response: lijst met gerate films + jouw rating. (statuscode = 200)

Response: 200

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ratedMovies>
  ▼<ratedMovie imdb="1234">
    <averageRating>8.0</averageRating>
    <date>2012</date>
    <director>Steven Seagal</director>
    <length>138</length>
    <shortDesc>Deze film is een serie</shortDesc>
    <title>Breaking Bad</title>
    <rating>8.0</rating>
  </ratedMovie>
</ratedMovies>
```

3.5 Get unrated

Verwachte response: lijst met niet gerate films (statuscode = 200)

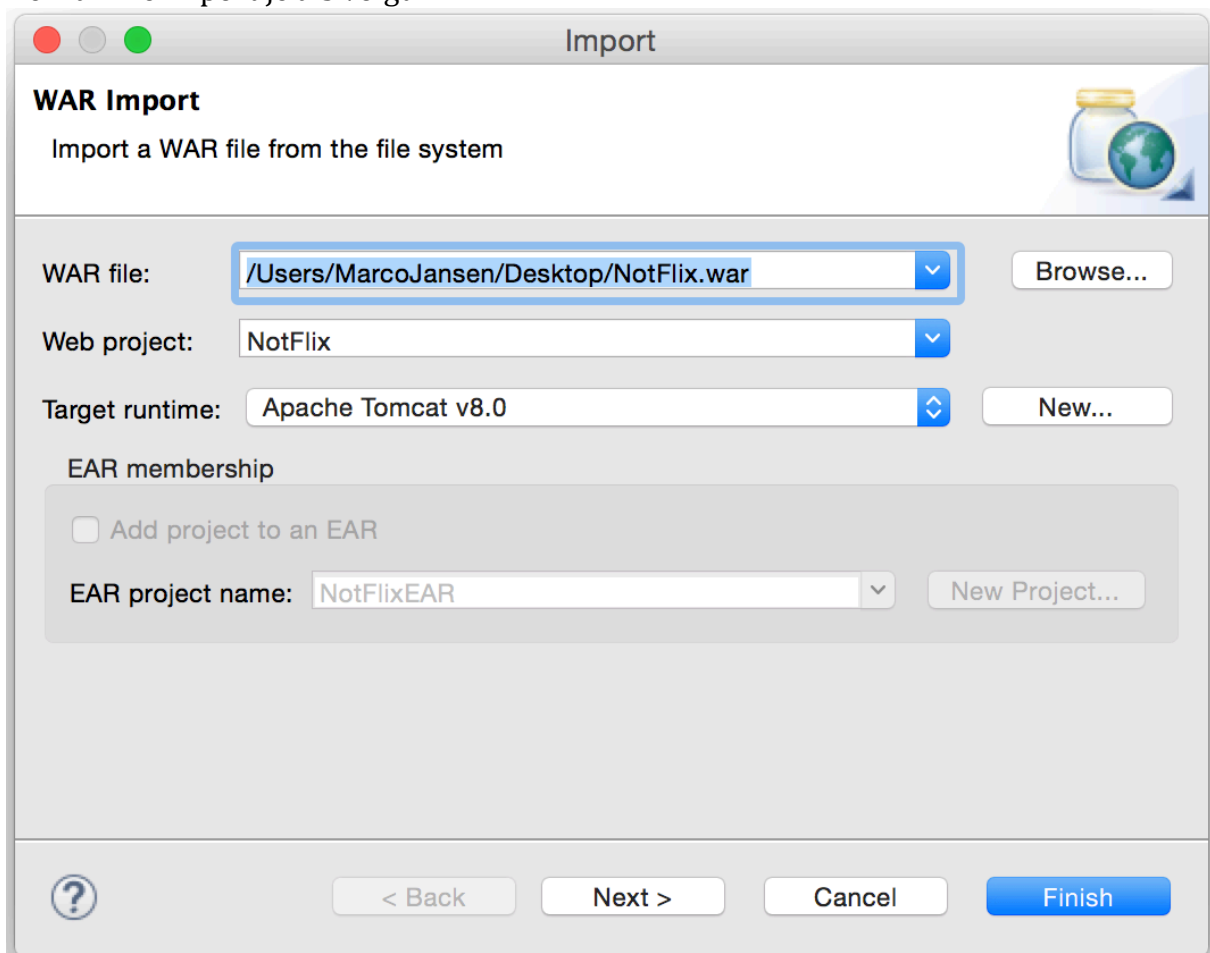
Response: 200

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<movies>
  <movie imdb="1235">
    <averageRating>0.0</averageRating>
    <date>2013</date>
    <director>Steven Seagul</director>
    <length>140</length>
    <shortDesc>Dit is een test</shortDesc>
    <title>Breaking Good</title>
  </movie>
</movies>
```

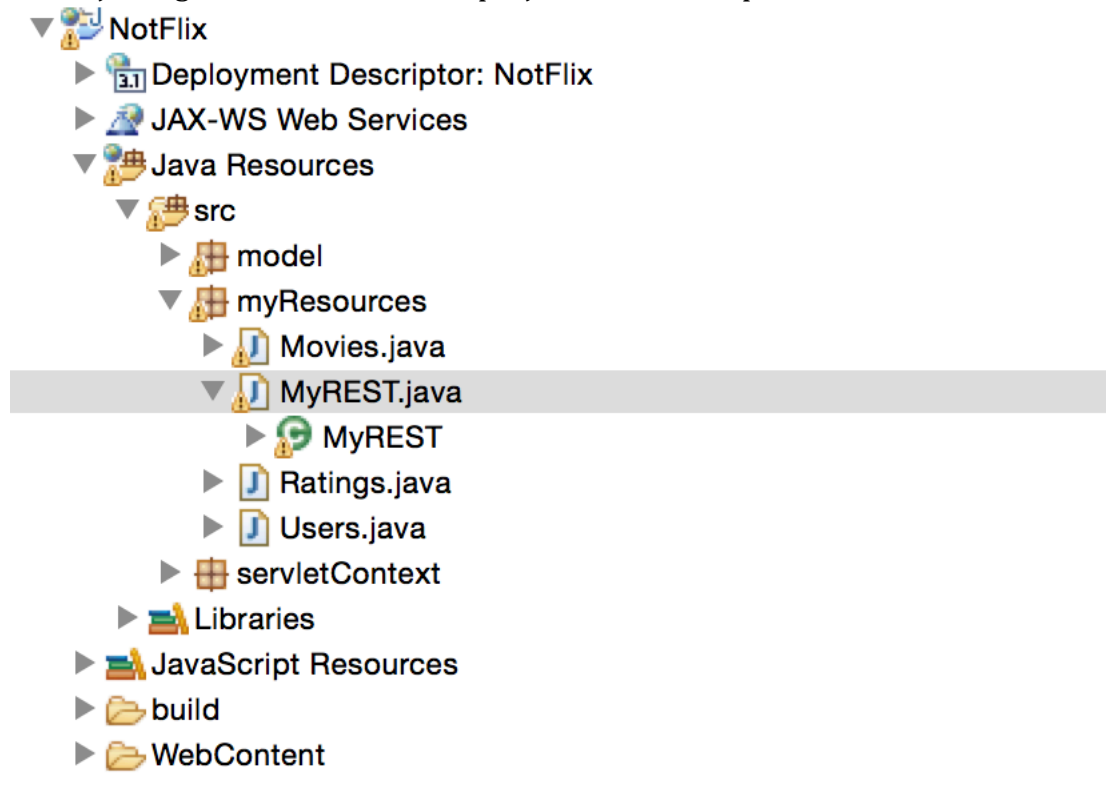
Alle testen zijn geslaagd. (verwacht komt overeen met daadwerkelijke response)

War file

De war-file import je als volgt:



Nadat je dit gedaan hebt komt het project in de workspace te staan.



De test is geslaagd. Het project heeft geen compile errors en het kan uitgevoerd worden.

Beveiliging

In onze REST service maken we gebruik van een token. Deze token is 24 karakters lang en bestaat uit kleine letters, hoofdletters en getallen. Wanneer je een user aanmaakt krijg je een token terug als response. Je kan ook inloggen om dit token op te halen. Voor elke request is een token nodig behalve voor de volgende drie:

1. Register (Post User)
2. Login (Post Login)
3. Get movies

Wanneer je token niet bekend is in onze service dan wordt er een 403 (Forbidden) terug gestuurd als response.