

DOSSIER WEBAPPLICATIES IV

Sébastien De Pauw
Klas: 2B1
Academiejaar: 2019-2020

Inhoud

Inleiding	1
Screenshots	2
Login	2
Registreren	2
Watchlist.....	3
Details.....	3
Toevoegen film	4
404 error.....	4
Extra technologie.....	5
Swagger print-outs	6
AccountController	6
FilmController.....	8
Structuur applicatie	13
Overzicht testen	15

Inleiding

Deze applicatie werd geschreven in opdracht van Stefaan De Cock voor het opleidingsonderdeel Webapplicaties IV. Ik heb gekozen om een applicatie te schrijven waarop je een lijst van films kan bijhouden. Het film object bestaat uit een titel, een duur, een categorie, een uitgave jaar, een poster en een detail object. Dit detail object bestaat dan op zijn beurt uit een beschrijving, een storyline, een rating, een acteur object en een regisseur object. Deze laatste houden alle twee een naam bij en acteur houdt ook een rol naam bij. Aan de lijst kunnen er films worden toegevoegd en verwijderd. Indien je wenst is het ook mogelijk om de details van één van de films te bekijken. Aangezien een lijst van films verbonden is aan één gebruiker is het verplicht om eerst aan te melden alvorens men in de applicatie kan. Anders zouden er films ongewenst dubbel worden toegevoegd aan de lijst.

Screenshots

Login

The screenshot shows a web browser window with the address bar displaying 'localhost:4200/login'. The page has a purple header bar. On the left, there is a sidebar menu with the following items: 'Menu', 'Watchlist', and 'Voeg film toe'. The main content area is white and contains a 'LOGIN' section. It includes an email input field with 'admin@hotmail.com' entered, a password input field with masked characters '*****', a 'registreer' link, and a purple 'Login' button.

Menu

Watchlist

Voeg film toe

LOGIN

email
admin@hotmail.com

password

[registreer](#)

Login

Registreren

The screenshot shows a web browser window with the address bar displaying 'localhost:4200/register'. The page has a purple header bar. On the left, there is a sidebar menu with the following items: 'Menu', 'Watchlist', and 'Voeg film toe'. The main content area is white and contains a 'Register' section. It includes input fields for 'voornaam', 'achternaam', 'email', 'Wachtwoord', and 'Bevestig wachtwoord'. There is a 'login' link and a purple 'Registreer' button.

Menu

Watchlist

Voeg film toe

Register

voornaam _____ achternaam _____

email _____

Wachtwoord _____

Bevestig wachtwoord _____

[login](#)

Registreer

Watchlist

Menu

Watchlist


Voeg film toe

admin@hotmail.com [logout](#)

filter

End Game


Sciencefiction - 2019 - 181 minuten



[Details](#)

Titanic


Romantiek - 1997 - 210 minuten



[Details](#)

Joker


Misdaad - 2019 - 122 minuten



[Details](#)

Die Hard


Actie - 1988 - 132 minuten



[Details](#)

Pulp Fiction


Misdaad - 1994 - 154 minuten



[Details](#)

Forrest Gump

Drama - 1994 - 144 minuten



[Details](#)

Details


Menu

Watchlist

Voeg film toe

admin@hotmail.com [logout](#)

[←](#)



Pulp Fiction

Uitgebracht in: 1994

Categorie: Misdaad

Speelduur: 154 minuten

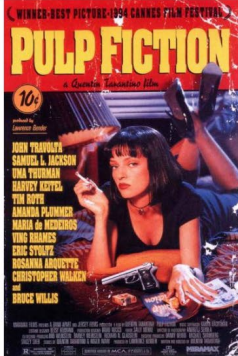
Rating: 6.5

Beschrijving:
Beschrijving

Storyline:
Korte inhoud

Acteurs:
Vincent Vega as John Travolta
Budge Coolidge as Bruce Willis
Jules Winnfield as Samuel L. Jackson

Regisseurs:
Quentin Tarantino



Toevoegen film

← → localhost:4200/film/add

Menu

admin@hotmail.com [logout](#)

Watchlist

Voeg film toe

Voeg Film toe:

Film:

Titel Jaar uitgave Speelduur (min.) Categorie

kies poster: [Bestand kiezen](#) Geen bestand gekozen

Details:

Korte beschrijving

Korte beschrijving

Storyline:

Storyline

Rating:

Rating

Acteurs:

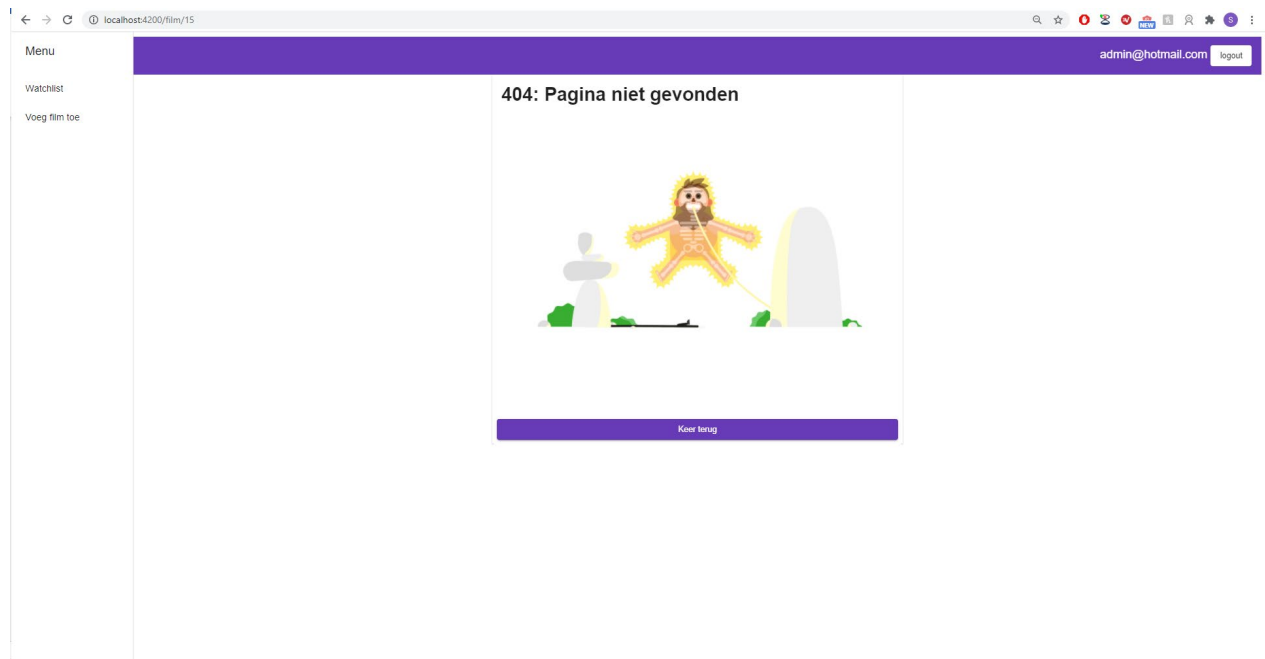
Voor- en achternaam Naam rol

Regisseurs:

Voor- en achternaam

[Toevoegen](#) [Annuleren](#)

404 error



Extra technologie

Als extra technologie koos ik voor het persisteren van een PNG. Als je een film wil toevoegen moet je een poster ook toevoegen. Dit is een foto die op de voorpagina onder de titel zal getoond worden. Wanneer je verkenner opent zullen alleen files kunnen gekozen worden met formaat. Deze file wordt dan omgezet naar een Base64 string die wordt doorgegeven naar de backend server via de POST methode waar hij ook wordt opgeslagen als Base64 string. Wanneer een GET wordt gedaan wordt deze Base64 string terug omgezet naar een afbeelding.

Swagger print-outs

Servers
https://localhost:5001

Authorize

Account

- POST** /api/Account Login
- POST** /api/Account/register Registreer
- GET** /api/Account/checkusername Kijkt of email vrij is als inlog

Film

- GET** /api/Film/{id} Geeft de details van een film in uw watchlist terug
- PUT** /api/Film/{id} Een film aanpassen
- DELETE** /api/Film/{id} Een film deleten
- GET** /api/Film Geef alle films in watchlist terug geordend op titel
- POST** /api/Film Voegt een film toe aan uw watchlist

Schemas

AccountController

POST /api/Account Login

Parameters

No parameters

Try it out

Request body **required**

application/json

de details van de login

Example Value | Schema

```
{  "email": "user@example.com",  "password": "string"}
```

Responses

Code	Description	Links
400	<p>Media type</p> <p>application/json</p> <p>Example Value Schema</p> <pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre>	No links

POST

/api/Account/register

Registreer

Parameters

Try it out

No parameters

Request body required

application/json

de details van de registratie

Example Value | Schema

```
{  "email": "user@example.com",  "password": "string",  "voornaam": "string",  "achternaam": "string",  "passwordConfirmation": "string"}
```

Responses

Code	Description	Links
400	<div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre></div>	

No links

 |

GET

/api/Account/checkusername

Kijkt of email vrij is als inlog

Parameters

Try it out

Name	Description
email	Email.
string	
(query)	

email - Email.

Responses

Code	Description	Links
200	<div>boolean die true returned als de email vrij is</div> <div>Media type<div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>true</pre></div> <div>No links</div>	

FilmController

GET `/api/Film/{id}` Geeft de details van een film in uw watchlist terug

Parameters

Try it out

Name	Description
id * required integer (path)	de id van de film waarvan de details getoond moeten worden

id - de id van de film waarvan de details geto

Responses

Code	Description	Links
200	een film	No links
Media type application/json		
Controls Accept header.		
Example Value Schema		
<pre>{ "id": 0, "titel": "string", "jaar": 0, "minuten": 0, "categori": "string", "poster": "string"}</pre>		
204	een film	No links
400		No links
Media type application/json		
Example Value Schema		
<pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre>		
404		No links
Media type application/json		
Example Value Schema		
<pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre>		
500		No links

PUT

/api/Film/{id} Een film aanpassen

Parameters

Try it out

Name	Description
<div><div>id * required</div><div>integer</div><div>(path)</div></div>	<div>id van de film die moet aangepast worden</div> <div>id - id van de film die moet aangepast worde</div>

Request body required

application/json

de herziene film

Example Value | Schema

```
{  "id": 0,  "titel": "string",  "jaar": 0,  "minuten": 0,  "categorie": "string",  "poster": "string"}
```

Responses

Code	Description	Links
200		No links
204		No links
400	<div><div>Media type</div><div>application/json</div><div>Example Value Schema</div><div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre></div></div>	No links
404	<div><div>Media type</div><div>application/json</div><div>Example Value Schema</div><div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre></div></div>	No links
500		No links

DELETE

/api/Film/{id}

Een film deleten

Parameters

Try it out

Name	Description
id <small>* required</small> integer (path)	het id van de film die je wil deleten
	<div>id - het id van de film die je wil deleten</div>

Responses

Code	Description	Links
200	<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 0, "titel": "string", "jaar": 0, "minuten": 0, "categorie": "string", "poster": "string"}</pre></div>	No links
204		No links
400	<div>Media type application/json</div> <div>Example Value Schema</div> <div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre></div>	No links
404	<div>Media type application/json</div> <div>Example Value Schema</div> <div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {}}</pre></div>	No links
500		No links

GET

/api/Film

Geef alle films in watchlist terug geordend op titel

Parameters

Try it out

Name	Description
titel string (query)	<input type="text" value="titel"/>
acteurNaam string (query)	<input type="text" value="acteurNaam"/>
regisseurNaam string (query)	<input type="text" value="regisseurNaam"/>

Responses

Code	Description	Links
200	<div>een film</div> <div>Media type <div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "id": 0, "titel": "string", "jaar": 0, "minuten": 0, "categorie": "string", "poster": "string" }]</pre></div> <div>No links</div>	No links
400	<div>Media type <div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {} }</pre></div> <div>No links</div>	No links
404	<div>Media type <div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {} }</pre></div> <div>No links</div>	No links
500		No links

POST

/api/Film Voegt een film toe aan uw watchlist

Parameters

Try it out

No parameters

Request body required

application/json

de nieuwe film

Example Value | Schema

```
{
  "titel": "string",
  "jaar": 0,
  "minuten": 0,
  "categorie": "string",
  "poster": "string"
}
```

Responses

Code	Description	Links
201	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div> <pre>{ "id": 0, "titel": "string", "jaar": 0, "minuten": 0, "categorie": "string", "poster": "string" }</pre> </div>	No links
400	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div> <pre>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "extensions": {} }</pre> </div>	No links
500		No links

Structuur applicatie

De componenten detail, film, acteur en regisseur beschikken over een model.ts om de JSON string te kunnen omzetten. De andere componenten bevatten de standaard files: CSS, html, spec.ts en ts

Modules: RoutingModule, FilmModule, UserModule, MaterialModule, AppModule

Componenten: FilmComponent, ActeurComponent, RegisseurComponent, FilmListComponent, FilmFilterPipe, AddFilmComponent, FilmDetailComponent, DetailComponent, MainNavComponent, PageNotFoundComponent, LoginComponent, RegisterComponent

Dataservices: FilmDataService

Hieronder vindt u een gedetailleerd overzicht van de applicatie weergegeven als boomstructuur

```

D:
| app-routing.module.ts
| app.component.css
| app.component.html
| app.component.spec.ts
| app.component.ts
| app.module.ts
+---film
| | film.module.ts
| +---acteur
| | | acteur.component.css
| | | acteur.component.html
| | | acteur.component.spec.ts
| | | acteur.component.ts
| | | acteur.model.ts
| +---add-film
| | | add-film.component.css
| | | add-film.component.html
| | | add-film.component.spec.ts
| | | add-film.component.ts
| +---detail
| | | detail.component.css
| | | detail.component.html
| | | detail.component.spec.ts
| | | detail.component.ts
| | | detail.model.ts
| +---film
| | | film-data.service.spec.ts
| | | film-data.service.ts
| | | film.component.css
| | | film.component.html
| | | film.component.spec.ts
| | | film.component.ts
| | | film.model.ts
| | | filmFilter.pipe.spec.ts
| | | filmFilter.pipe.ts
| | | FilmResolver.ts
| | | mock-films.ts
| +---film-detail
| | | film-detail.component.css
| | | film-detail.component.html
| | | film-detail.component.spec.ts
| | | film-detail.component.ts
| +---filmlist
| | | filmlist.component.css
| | | filmlist.component.html
| | | filmlist.component.spec.ts
| | | filmlist.component.ts
| \---regisseur
| | | regisseur.component.css
| | | regisseur.component.html
| | | regisseur.component.spec.ts
| | | regisseur.component.ts
| | | regisseur.model.ts
+---film
| | film-data.service.spec.ts
| | film-data.service.ts
| | film.component.css
| | film.component.html
| | film.component.spec.ts
| | film.component.ts
| | film.model.ts
| | filmFilter.pipe.spec.ts
| | filmFilter.pipe.ts
| | FilmResolver.ts
| | mock-films.ts
+---film-detail
| | film-detail.component.css
| | film-detail.component.html
| | film-detail.component.spec.ts
| | film-detail.component.ts
+---filmlist
| | filmlist.component.css
| | filmlist.component.html
| | filmlist.component.spec.ts
| | filmlist.component.ts
| \---regisseur
| | regisseur.component.css
| | regisseur.component.html
| | regisseur.component.spec.ts
| | regisseur.component.ts
| | regisseur.model.ts
+---http-interceptors
| | AuthenticationInterceptor.ts
| | index.ts
+---main-nav
| | main-nav.component.css
| | main-nav.component.html
| | main-nav.component.spec.ts
| | main-nav.component.ts
+---material
| | material.module.ts
+---page-not-found
| | page-not-found.component.css
| | page-not-found.component.html
| | page-not-found.component.spec.ts
| | page-not-found.component.ts
| \---user
| | | auth.guard.spec.ts
| | | auth.guard.ts
| | | authentication.service.spec.ts
| | | authentication.service.ts
| | | user.module.ts
+---login
| | login.component.css
| | login.component.html
| | login.component.spec.ts
| | login.component.ts
| \---register
| | register.component.css
| | register.component.html
| | register.component.spec.ts
| | register.component.ts

```


Overzicht testen

FilmApp

Support Docs Log In

Tests Runs Settings

Chrome 84

Search...

Run all specs

INTEGRATION TESTS

FormValidationTests.spec.js

HttpTests.spec.js

myfirsttest.spec.js

< Tests

✓ 2 ✗ -- ○ --

05.02

○ ↑ ○ ↺

cypress\integration\FormValidationTests.spec.js

▼ Form validation with wrong input shows error message

✓ Touch required fields show error

▼ Form validation tests with correct input but no photo cant submit

✓ Form filled in correctly enables submit button

< Tests

✓ 9 ✗ -- ○ --

14.74

○ ↑ ○ ↺

cypress\integration\HttpTests.spec.js

▼ Routertest

✓ Correct startup

✓ HttpGet films

✓ Mock film get

✓ Filter titel

✓ on error should show error message

✓ Routes naar voegtoe

✓ Routes naar watchlist

✓ Routes naar login

✓ Routes naar registreer