

Installatiehandleiding



Naam: Nangialai Waziri

Datum: 14-11-2021

Opdracht: full-stack

Inhoudsopgave

Inhoud	Pagina
Inleiding	3
Hoofdstuk 1: Benodigdheden	4-5
Hoofdstuk 2: Installatie backend	6
Hoofdstuk 3: Installatie front-end	6
API documentatie	7-13

Inleiding

Deze installatiehandleiding is geschreven voor het project Don Diablo Demo Application. Via deze applicatie kunnen gebruikers muziek demo's uploaden om hierop feedback te krijgen.

Het doel van deze handleiding is om ervoor te zorgen dat de applicatie in verschillende IT-omgevingen te installeren en te gebruiken. De database wordt tijdens de installatie procedure gevuld met test data in een database.

Om de installatie en gebruiksprocedure zo simpel mogelijk te laten verlopen zijn de volgende stappen opgesteld:

1. Hardware en software benodigdheden
2. Stappen plan installatie backend.
3. Stappen plan installatie front-end.
4. API Documentatie.

Hoofdstuk 1: Hardware en software benodigdheden

In dit hoofdstuk worden de benodigdheden die vooraf geïnstalleerd dienen te worden om te applicatie te kunnen draaien. De applicatie draait op de localhost en deze wordt gebruikt in de hele handleiding. Verder vereist de applicatie toegang tot het internet omdat het automatisch een email stuurt nadat de gebruiker een file uploadt.

Hardware

Deze applicatie is ontwikkeld op een system met 10^e generatie Intel I7 CPU en 16GB RAM. Verwacht wordt dat de applicatie kan draaien op een minder zwaar systeem. Echter, houdt er rekening mee dat de applicatie niet zou kunnen draaien op oude system zoals bijvoorbeeld een Intel Pentium-4 processor en 1GB RAM.

Software

IntelliJ IDEA v2021.2.3

De backend API is geschreven in IntelliJ IDEA. Deze kan gedownload worden via <https://www.jetbrains.com/idea/download/#section=windows>.

Installatie stappen:

1. Open de locatie waar de executabel (.exe) is op geslagen.
2. Run de executabel.
3. Windows vraagt of het toegestaan is om de executabel te runnen → Klik op “Ja”
4. In de volgende 3 vensters klik op “Next”
5. Klik in de volgende venster op “Install”
6. Klik op finish na de installatie.
7. Open de applicatie en ga akkoord met de gebruiksvoorwaarden.

AdoptOpenJDK v11.0.11

De programmeer taal die gebruikt om te backend te schrijven is Java v11. Om deze te installeren is gebruikt gemaakt van de AdoptOpenJDK die gedownload kan worden via <https://adoptopenjdk.net/?variant=openjdk11&jvmVariant=openj9>. Kies voor de versie OpenJDK 11 (LTS) en selecteer OpenJ9 voor de JVM. Deze JDK (java development kit) is nodig om Java applicaties in IntelliJ te schrijven.

Installatie stappen:

1. Open de locatie waar de executabel (.exe) is op geslagen.
2. Run de executabel.
3. Windows vraagt of het toegestaan is om de executabel te runnen → Klik op “Ja”
4. In de volgende 2 vensters klik op “Next”
5. Klik in de volgende venster op “Install”
6. Klik op finish na de installatie.
7. Open de applicatie en ga akkoord met de gebruiksvoorwaarden.
8. Open de commandline prompt en voer de command: java -version in → Nu zul je de hierboven genoemde versie van de JDK zien staan.

Java Run Time Environment (JRE) v18.9

De JRE wordt mee geïnstalleert bij de installatie van IntelliJ en de AdoptOpenJDK. De build die gebruikt is voor dit project is 11.0.10+8-LTS-162.

Apache Maven v3.6.3

De backend maakt gebruik van Apache Maven als build tool. Deze wordt op de motorkap geïnstalleerd tijdens de installatie van IntelliJ IDEA. De installatie van Maven zou correct moeten verlopen bij de installatie van een nieuwere versie van IntelliJ. In de pom.xml zijn alle dependencies weer gegeven.

Springboot v2.5.4

Tijdens de initialisatie proces van de backend applicatie is gekozen voor een springboot project die draait op de hierboven genoemde versie. De packaging wordt gedaan in Jar. IntelliJ IDEA kan hiermee overweg, u hoeft geen extra stappen uit te voeren om springboot te installeren.

NodeJS v14.16

De front-end is geschreven met ReactJS, een JavaScript library dat gebruikt wordt voor front-end applicaties. De gegeven versie van NodeJS is te downloaden via <https://nodejs.org/de/blog/release/v14.16.0/>. Download de correcte installer of binary voor uw OS.

Installatie stappen:

1. Kies de correcte Windows 64-bit installer.
2. Download de file.
3. Navigeer naar de download locatie en run de gedownloade file.
4. Windows geeft een beveiligingswaarschuwing → klik op "OK".
5. Klik op "next" in de volgende venster.
6. Accepteer de gebruikersvoorwaarden en klik op "next".
7. Klik in de volgende 2 vensters op "next".
8. Klik op "install"
9. Klik op "Ja" in de windows waarschuwing
10. Nadat de NodeJS geïnstalleerd is op een command prompt en voer de command: node – version uit.

NPM v6.14.11

Javascript packages worden in dit project geïnstalleerd via NPM. NPM wordt geïnstalleerd met NodeJS op windows in de stap hierboven.

Webstorm v2021.3

Webstorm is een IDEA voor front-end development die gebruikt is om de front-end te programmeren. De IDEA kan gedownload worden via:

<https://www.jetbrains.com/webstorm/download/#section=windows>

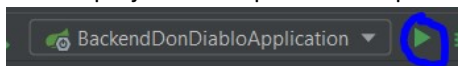
Installatie stappen:

1. Navigeer naar de download locatie en run de executable.
2. Windows geeft een beveiligingswaarschuwing → klik op "OK".
3. Klik op "next" in de volgende 3 vensters.
4. Klik op "install" in de volgende venster.
5. Klik op "finish" na de installie.

Hoofdstuk 2: Installatie backend applicatie

In deze hoofdstuk wordt de installatie en opstart procedure van de backend besproken. De gebruiker krijgt een aantal stappen die in de gegeven volgorde uitgevoerd moeten worden. De applicatie zal draaien op localhost:8080. Zorg er dus voor dat de poort vrij is. De applicatie maakt automatisch de volgende users aan: Username: nang (Password: test) en Username: admin (Password: test). U kunt deze gebruiken om direct aan de slag te gaan met de applicatie.

1. Clone de repository: <https://github.com/NWaziri/fullstack-DonDiablo>.
2. Open PgAdmin en maak de database: donDiablo aan in Postgres on de gebruiker postgres en password: admin. Mocht u andere wachtwoord hebben pas dan deze aan in ~/ backend-DonDiablo\src\main\resources\application.properties file aan onder:
spring.datasource.password=<wachtwoord>
3. De backend applicatie probeert postgres op poort 5432 te bereiken, zorg dat deze poort vrij is of verander ~/ backend-DonDiablo\src\main\resources\application.properties onder
spring.datasource.url=jdbc:postgresql://localhost:<zet hier de geschikte poort> /donDiablo
4. Open IntelliJ IDEA.
5. Open de backend-DonDiablo map uit het geclone project in IntelliJ IDEA.
6. Rechts onder in de hoek zal een pop-up verschijnen "Maven Build script found" → klik op "import".
7. Run het project door op de run knop te drukken.



Hoofdstuk 3: Installatie front-end applicatie

In deze hoofdstuk wordt de front-end applicatie geïnstalleerd. De applicatie draait op localhost:3000. Zorg ervoor dat eerst de backend applicatie zoals besproken in hoofdstuk 2 is opgestart voordat de front-end wordt gestart. Er volgt een reeks aantal stappen om de applicatie te installeren en op te starten. Let op! De applicatie is getest op Google Chrome en Mozilla Firefox. Er is geen garantie dat de applicatie goed werkt in andere browsers. De applicatie maakt automatisch de volgende users aan: Username: nang (Password: test) en Username: admin (Password: test). U kunt deze gebruiken om direct aan de slag te gaan met de applicatie.

1. Open Webstorm IDEA.
2. Open de folder front-end-dondiablo in Webstorm als project
3. In webstorm open een terminal en run de command: npm install → zorg dat je daadwerkelijk in de front-end-dondiablo zit.
4. Installeer jest: npm install jest --save
5. Nadat de packages zijn geïnstalleerd run je de command: npm start
6. De applicatie wordt automatisch geopend in de browser, zo niet voer dan:
<http://localhost:3000> in de browser adresbalk.

API Documentatie

Authentication API

De authentication API zorgt ervoor dat gebruikers een json web token kunnen ophalen

- Haal json web token op via basic Auth

URL: `http://localhost:8080/authenticate`

Method: POST

Auth required: Basic

Success Response

Code: 200 OK

Content voorbeeld

For een gebruiker in de database met naam: nang en wachtwoord: test

Body

```
{
  "username": "nang",
  "password": "test"
}
```

User API

The users API can be used to perform CRUD operations on users. It contains the following requests:

- Haal details op van alle gebruiker.
- Haal details op van een specifieke gebruiker.
- Verwijder gebruiker.
- Haal gebruiker rol(len) op.

Haal details op van alle gebruiker:

URL: `http://localhost:8080/users/`

Method: GET

Auth required: Bearer Token

Success Response

Code: 200 OK

Content voorbeeld

Response

```
[
  {
    "username": "nang",
    "password":
"$2a$10$.BzdLWwqLaVf225/E9JPCe/uIkDL4ZFJmBOR56Km6J7FYqIeWMZf6",
    "enabled": true,
    "email": "test1@gmail.com",
    "authorities": [
      {
        "username": "nang",
        "authority": "ROLE_USER"
      }
    ]
  },
  {
    "username": "admin",
    "password":
"$2a$10$.BzdLWwqLaVf225/E9JPCe/uIkDL4ZFJmBOR56Km6J7FYqIeWMZf6",
    "enabled": true,
    "email": "test1@hotmail.com",
    "authorities": [
      {
        "username": "admin",
        "authority": "ROLE_ADMIN"
      }
    ]
  }
]
```

Haal details op van een specifieke gebruiker.

URL: `http://localhost:8080/users/{username}`

Method: GET

Auth required: Bearer Token

Success Response

Code: 200 OK

Content voorbeeld

For een gebruiker in de database met naam: nang

Response

```
{
  "username": "nang",
  "password":
"$2a$10$.BzdLWwqLaVf225/E9JPce/uIkDL4ZFJmBOR56Km6J7FYqIeWMZf6",
  "enabled": true,
  "email": "test1@gmail.com",
  "authorities": [
    {
      "username": "nang",
      "authority": "ROLE_USER"
    }
  ]
}
```

Maak een gebruiker aan:

URL: `http://localhost:8080/users/`

Method: POST

Auth required: None

Success Response

Code: 201 Created

Content voorbeeld

Maak een user aan met de volgende gegevens in de body:

Body

```
{
  "email": "jan@gmail.com",
  "username": "jan",
  "password": "password"
}
```

Verwijder gebruiker:

URL: `http://localhost:8080/users/{username}`

Method: DELETE

Auth required: Bearer Token

Success Response

Code: 204 NO_CONTENT

Content voorbeeld

Voer een delete uit naar de volgende url: <http://localhost:8080/users/jan>

Haal de rollen op van een specifieke gebruiker.

URL: <http://localhost:8080/users/{username}/authorities>

Method: GET

Auth required: Bearer Token

Success Response

Code: 200 OK

Content voorbeeld

For een gebruiker in de database met naam: nang.

url: <http://localhost:8080/users/nang/authorities>

Response

```
[
  {
    "username": "nang",
    "authority": "ROLE_USER"
  }
]
```

Files API

De files API zorgt ervoor dat files geupload en gedownload kunnen worden. De API bevat de volgende requests

- Haal informatie over een file op.
- Haal informatie over alle files voor een specifieke gebruiker op.
- Upload file.
- Download file.

Haal informatie van een specifieke files op van een specifieke gebruiker:

URL: <http://localhost:8080/files/filesinfo/{name}>

Method: GET

Auth required: Bearer Token

Success Response

Code: 200 OK

Content voorbeeld

Haal de informatie op met gebruikersnaam: nang:

URL: <http://localhost:8080/files/filesinfo/nang>

Response

```
[
  {
    "id": 1,
    "fileName": "file_example_WAV_2MG.mp3",
    "uploader": "nang",
    "uploadDate": "2021/11/14",
    "comment": {
      "id": 1,
      "content": "We zijn bezig met het beoordelen van de demo,
binnen 24 uur ontvangt u commentaar"
    }
  }
]
```

Upload een file:

URL: `http://localhost:8080/files/upload`

Method: POST

Auth required: Bearer Token

Success Response

Code: 200 OK

Text: file uploaded

Content voorbeeld Upload file met

gebruikersnaam: nang:

email: test1@gmail.com

URL: `http://localhost:8080/files/filesinfo/nang`

Body form-data

file <path to file>

uploader nang

email test1@gmail.com

Response

file uploaded

Download een file:

URL: `http://localhost:8080/files/{filename}`

Method: GET

Auth required: Bearer Token

Success Response

Code: 200 OK

Content: file

Content voorbeeld

Download de eerder geuploade file

Let op! De filename moet een .mp3 extension hebben omdat deze geconverteerd is. Dus als test.wav geupload was dan is de filenaam voor het downloaden test.mp3

URL: `http://localhost:8080/files/filesinfo/nang`

Response

De file in bytes

Comments API

De comments API haalt informatie op over de commentaar dat op files is gegeven en commentaar kan geupdate worden. Deze Api bevat de volgende requests

- Update de comment per file
- Haal de comment up per file

Update de comment:

URL: `http://localhost:8080/comment/{id}`

Method: PUT

Auth required: Bearer Token

Success Response

Code: 204 NO_CONTENT

Content voorbeeld

Geef commentaar op de audio file door de volgende body te sturen. Let op pas de filenaam met een .mp3 extension aan en de uploadDate in de correcte formaat (jaar/maand/dag), zie voorbeeld body

Body

```
{
  "content": "nieuwe content",
  "musicFile": {
    "fileName": "test.mp3",
    "uploader": "nang",
    "uploadDate": "2021/11/14"
  }
}
```

Update de comment:**URL** : `http://localhost:8080/comment/{id}`**Method** : GET**Auth required** : Bearer Token

Success Response

Code : 200 OK**Content voorbeeld**

Haal de commentaar op met id = 1

Body

```
{
  "id": 1,
  "content": "We zijn bezig met het beoordelen van de demo, binnen 24
uur ontvangt u commentaar"
}
```