

Dokumentacja wdrożeniowa

Projekt zespołowy PAI

„Flexible Scheduler”

Emilia Pustelnik oraz Artur Matuszczyk

1. Pobranie kodu

Ze strony: <https://github.com/Kiriar21/scheduler> klonujemy repozytorium i odpalamy w VSC.

- Klonowanie w konsoli: `gh repo clone Kiriar21/scheduler`
- Pobieranie: Code -> download ZIP -> Wypakowujemy w wybranym przez nas miejscu na dysku

Jak już pobierzemy kod, potrzebujemy zainstalować wszystkie wymagane biblioteki wykorzystywane w projekcie. Otwieramy konsolę i szukamy naszego wypakowanego projektu „scheduler”. Osobno uruchamiamy komendę: „npm install” w folderze backend oraz frontend.

```
C:\Projekty\scheduler> cd .\backend\  
C:\Projekty\scheduler\backend> npm install
```

```
C:\Projekty\scheduler> cd .\frontend\  
C:\Projekty\scheduler\frontend> npm install
```

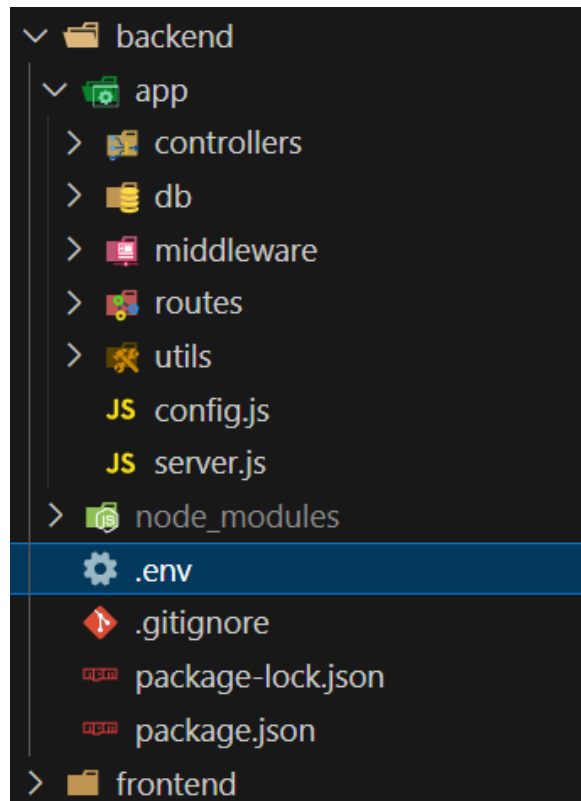
WAŻNE: Żeby móc zainstalować potrzebujemy mieć zainstalowany „Node” na komputerze.

*Kod aplikacji był testowany i uruchamiany na systemie Windows. Aplikacja powinna w ten sam sposób działać na systemach z rodziny Linux – jednak nie dajemy gwarancji poprawności działania.

2. Dodanie bazy danych

Aplikacja ma już ustawioną konfigurację, możliwą do pobrania – nic nie trzeba ustawiać. Jeżeli jednak chcemy działać na swojej bazie MongoDB to instrukcja wygląda następująco:

Potrzebujemy w plikach backendu dodać informację o połączeniu z bazą danych. Wchodzimy w kodzie w folder backend i szukamy pliku „.env”.



W nim musimy uzupełnić „**MONGO_URI**” – znajdziemy go, jeżeli założymy konto na MongoDB online i znajdziemy tam pomoc w podłączeniu do bazy na zasadzie:

```
mongodb+srv://"NAZWAUSERA": "HASLOUSERA"@ "LINKDOBAZY"
```

Jak to podmienimy, wystarczy, że zresetujemy nasz serwer backendu i działamy na swojej bazie!

3. Uruchomienie aplikacji

Można już uruchomić aplikację, za każdym razem musimy odpalić dwa serwery, jeden dla backendu oraz drugi dla front-endu. Najpierw backend uruchamiamy przez: „npm start”

```
PS C:\Projekty\scheduler> cd .\backend\;
PS C:\Projekty\scheduler\backend> npm start

> scheduler@1.0.0 start
> node app/server.js

(node:12228) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:12228) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Serwer działa na porcie 5000
Połączono z MongoDB
```

A następnie uruchamiamy frontend, również używając „npm start”

```
PS C:\Projekty\scheduler> cd .\frontend\  
PS C:\Projekty\scheduler\frontend> npm start  
  
> frontend@0.1.0 start  
> react-scripts start
```

...

```
Compiled successfully!  
  
You can now view frontend in the browser.  
  
Local:          http://localhost:3000  
On Your Network: http://192.168.11.230:3000  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully
```

I teraz wchodzimy pod adres w przeglądarce: <http://localhost:3000> i możemy korzystać z aplikacji!

Wszystko co dostępne w aplikacji można znaleźć w dokumentacji użytkowej.

Dokumentacja techniczna – Node/back-end:

[Link](#)

Dokumentacja techniczna – React/front-end:

[Link](#)

Dokumentacja użytkowa

[Link](#)