

Wydział Informatyki i Telekomunikacji
Politechniki Poznańskiej

Instalacja systemu aplikacji mobilnej „PetFinder” w wersji dla programisty

Michał Kapała
Bartosz Czarnecki

Utworzono: 30.03.2021 r.

Ostatnie zmiany: 21.01.2022r.

Spis treści

Wstęp	3
Serwer bazy danych	3
Aplikacja serwerowa	3
Emulator	3
Expo CLI	5
Instalacja	5
Uruchomienie serwera Expo Dev Tools	7

Wstęp

Niniejszy dokument przedstawia instrukcje instalacji elementów niezbędnych do uruchomienia aplikacji mobilnej „PetFinder”. Aplikacja kliencka jest implementowana w języku programowania JavaScript przy użyciu frameworku **React Native**. Narzędzia niezbędne do jej uruchomienia to:

- Node.js (wersja 17.0.1)
- Expo CLI (narzędzia uruchomieniowe aplikacji mobilnej w React Native; wersja 44.0.0)
- Android Studio (środowisko wirtualizacji mobilnego systemu operacyjnego; niezbędne w przypadku uruchomienia na emulatorze; wersja 2020.3.1.26)

System zarządzania bazą danych działa w oparciu o kontener programu Docker oraz serwer baz danych SQL Server. Aplikacja serwerowa została napisana w języku Go. Narzędzia niezbędne do uruchomienia tych części systemu to:

- Go (język programowania, wykorzystywany w aplikacji serwerowej, wersja 1.17.6)
- Docker (środowisko konteneryzacji serwera bazy danych, Docker Engine wersja 20.10.8)

Serwer bazy danych

Aby uruchomić serwer bazy danych należy, po zainstalowaniu programu Docker, wykonać następujące komendy za pomocą wiersza poleceń w folderze projektu:

1. `cd db`
2. `docker-compose up`

Aplikacja serwerowa

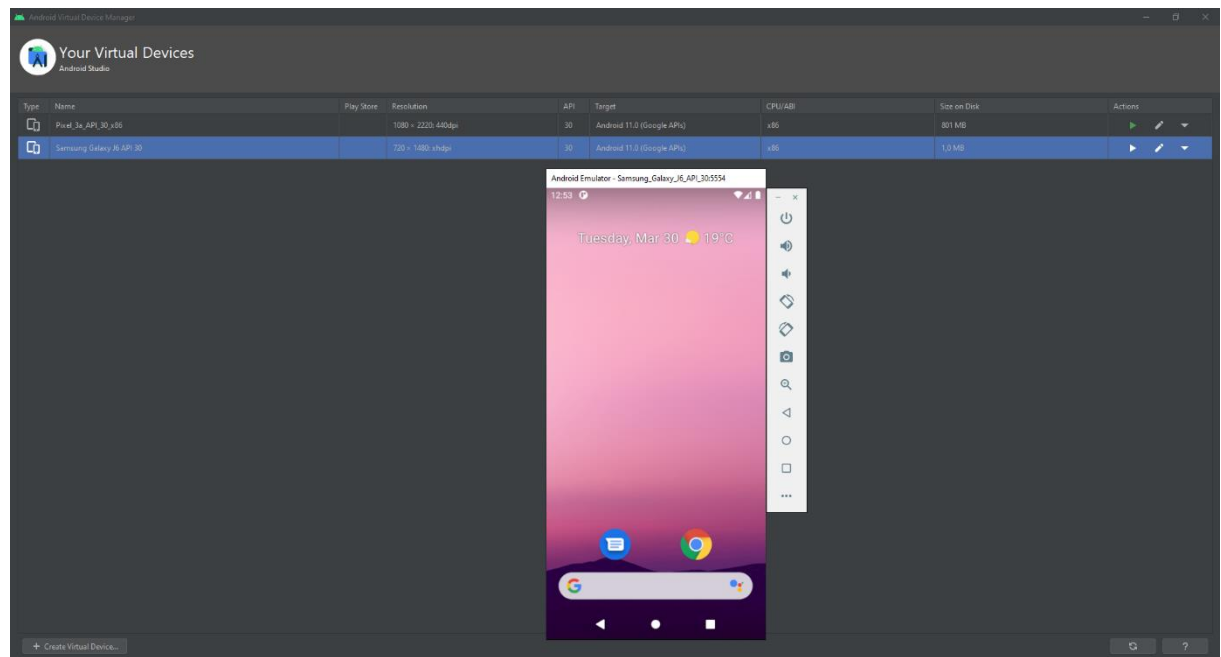
Aby uruchomić aplikację serwerową należy wykonać następujące komendy za pomocą wiersza poleceń w folderze projektu:

1. `cd server`
2. `go run main.go`

Emulator

Aby skonfigurować emulator systemu Android w systemie Windows, należy:

1. Zainstalować środowisko Android Studio
2. Utworzyć urządzenie wirtualne z systemem Android
3. Uruchomić urządzenie wirtualne



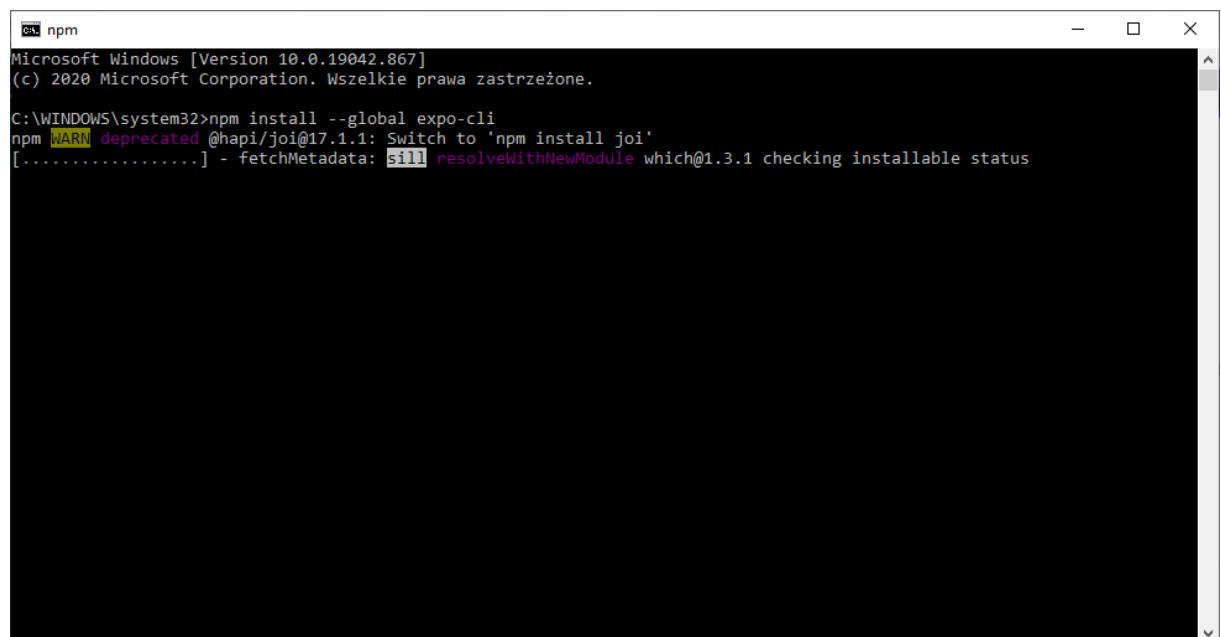
Rysunek 1 Uruchomione urządzenie wirtualne

Expo CLI

Expo CLI to zestaw narzędzi dla platformy React Native, umożliwiających komunikację między środowiskiem deweloperskim a docelowym środowiskiem uruchomieniowym aplikacji mobilnej oraz innych narzędzi dla deweloperów. Do podstawowych funkcjonalności należy m.in. serwer udostępniający aplikację wraz z zasobami, tworzenie pakietów instalacyjnych czy dostęp do platform dla aplikacji mobilnych takich jak Google Play czy Apple Store. Do zalet tego rozwiązania należy też brak potrzeby osadzania plików aplikacji w środowisku uruchomieniowym – interfejs udostępnia aplikację środowiskom uruchomieniowym w czasie rzeczywistym, a do wprowadzenia zmian wystarczy zapisanie stanu projektu.

Instalacja

Po pomyślnej instalacji Node.js należy wywołać komendę `npm install -global expo-cli` z wiersza poleceń:



```
npm
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\WINDOWS\system32>npm install --global expo-cli
npm WARN deprecated @hapi/joi@17.1.1: Switch to 'npm install joi'
[.....] - fetchMetadata: sill resolveWithNewModule which@1.3.1 checking installable status
```

Rysunek 2 Rozpoczęcie instalacji z wiersza poleceń

```
Administrator: Wiersz polecenia
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> core-js@3.9.1 postinstall C:\Users\Michal\AppData\Roaming\npm\node_modules\expo-cli\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.1 (node_modules\expo-cli\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~1.2.7 (node_modules\expo-cli\node_modules\watchpack-chokidar2\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~1.2.7 (node_modules\expo-cli\node_modules\webpack-dev-server\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN @pmmmwh/react-refresh-webpack-plugin@0.3.3 requires a peer of react-refresh@^0.8.2 but none is installed. You must install peer dependencies yourself.

+ expo-cli@4.3.2
added 1811 packages from 783 contributors in 195.571s

C:\WINDOWS\system32>
```

Rysunek 3 Ukończenie instalacji Expo CLI

Uruchomienie serwera Expo Dev Tools

Aby udostępnić aplikację środowiskom uruchomieniowym (są dwie opcje) należy uruchomić serwer Expo Developer Tools:

1. Wejść do folderu projektu (bezpośrednio zawierającego pliki projektu) w wierszu poleceń.
2. Wywołać komendę `cd client`
3. Z lokalizacji projektu wywołać komendę `npm start`.

[illegible]

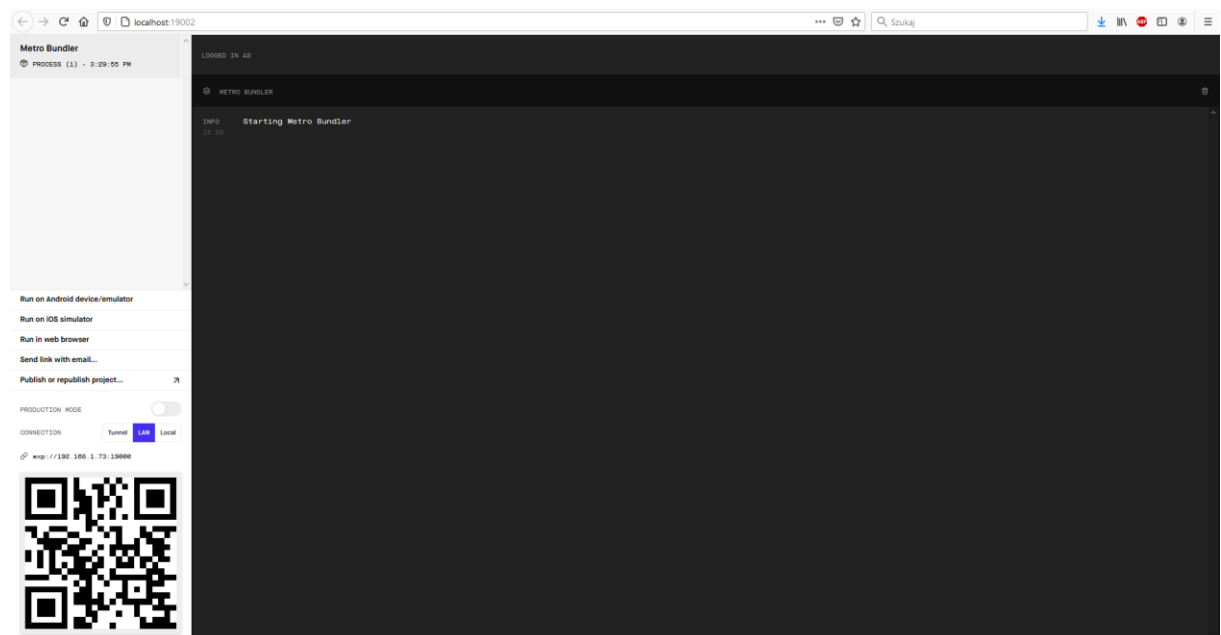
Rysunek 4 Komenda `npm start`

4. Jeżeli nie otworzyła się nowa karta z panelem serwera w przeglądarce, należy wywołać komendę `expo start`.



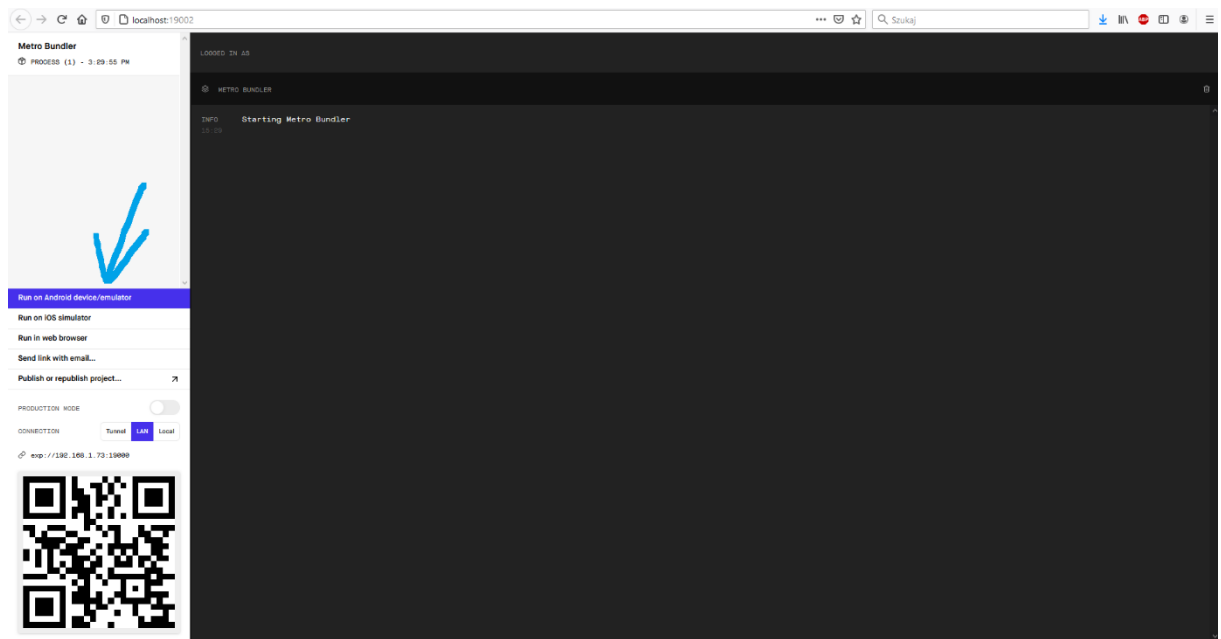
Rysunek 5 Uruchomienie serwera Expo

5. W domyślnej przeglądarce powinna otworzyć się nowa karta z interfejsem Expo Dev Tools (adres **localhost:19002**).



Rysunek 6 Expo Dev Tools

6. Uruchomić aplikację na emulatorze



Rysunek 7 Uruchomienie urządzenia na emulatorze