

Wymagania wstępne:

Node.js w wersji 20.9.0

Konto Azure z utworzonym zasobem Azure Cosmos DB

Skonfigurowane zmienne środowiskowe (plik .env) z danymi dostępowymi do Cosmos DB i portem

Rozpoczęcie pracy:

1. Sklonuj repozytorium:

```
git clone https://github.com/wkoltonowska/Scan-my-stuff
```

```
cd Scan-my-stuff
```

2. Zainstaluj zależności:

```
npm install
```

```
npm install uuid dotenv
```

3. Konfiguracja:

Utworzyć plik .env i uzupełnić:

```
COSMOS_DB_ENDPOINT=your_cosmos_endpoint  
COSMOS_DB_KEY=your_cosmos_key  
COSMOS_DB_DATABASE=your_database_name  
COSMOS_DB_CONTAINER=your_container_name  
PORT=8080
```

Wymagane paczki

Aplikacja korzysta z:

- dotenv - do ładowania zmiennych środowiskowych
- uuid - do generowania unikalnych identyfikatorów

Są one automatycznie instalowane przez npm install jeśli są w package.json. Jeśli potrzebujesz zainstalować je ręcznie:

```
npm install uuid dotenv --save
```

4. Uruchomienie lokalne:

```
npm start lub node server.js
```

Aplikacja będzie dostępna pod adresem: <http://localhost:8080>

5. Wdrożenie na Azure:

a) Utwórz aplikację Web App

Stwórz nową Web App (zalecane: Linux, Node.js)

b) Skonfiguruj CI/CD

Połącz repozytorium GitHub z aplikacją w Azure

Azure automatycznie wykryje package.json i zainstaluje zależności

c) Ustaw zmienne środowiskowe w Azure Portal:

- Przejdź do swojej aplikacji → Configuration → Application Settings
- Dodaj wszystkie zmienne z pliku .env

d) Weryfikacja wdrożenia

Po udanym wdrożeniu aplikacja będzie dostępna pod adresem:

<https://scan-my-stuff-app.azurewebsites.net>

4. Testowanie po wdrożeniu

a) Otwórz adres produkcyjny

b) Przetestuj funkcjonalność skanowania kodów kreskowych

c) Sprawdź czy dane są zapisywane w Cosmos DB

5. Rozwiązywanie problemów

Jeśli wystąpią problemy:

- Sprawdź logi w Azure Portal (Log Stream)
- Upewnij się, że wszystkie zmienne środowiskowe są poprawnie ustawione
- Sprawdź połączenie z Cosmos DB w Azure Cosmos DB Overview

Rozwiązywanie problemów