
Dokumentacja Techniczna Backend

Tomasz Chady

January 15, 2024

Contents

1	Dokumentacja API	1
1.1	Endpointy	1
1.1.1	Logowanie	1
1.1.2	Użytkownicy	1
1.1.3	Administracja	2
1.2	Autoryzacja	2
2	Architektura Backendu	2
3	Baza danych	2
3.1	Schemat	2
3.2	Modele	2

1 Dokumentacja API

1.1 Endpointy

Wszystkie endpointy są dostępne pod adresem `/api/`.

1.1.1 Logowanie

Poniżej znajdują się endpointy odpowiedzialne za logowanie. W odróżnieniu od innych endpointów, nie wymagają one autoryzacji.

- POST `/api/login`
- POST `/api/login-step2`

1.1.2 Użytkownicy

Poniżej znajdują się endpointy odpowiedzialne za zarządzanie użytkownikami.

- GET `/api/profile`
- POST `/api/disable-2fa`
- POST `/api/generate-2fa-secret`
- POST `/api/verify-otp`

1.1.3 Administracja

Poniżej wymienione są endpointy, powiązane z administracją systemem. Z reguły posiadają swoje własne metody autoryzacji.

- POST /api/signup

1.2 Autoryzacja

2 Architektura Backendu

Backend został napisany w formie monolitycznego REST API. Głównym zadaniem backendu jest udostępnienie danych z bazy danych. Dodatkowo backend jest odpowiedzialny za interakcję z powiązanymi systemami oraz autoryzację użytkowników. Relatywna pozycja backendu w systemie jest przedstawiona na schemacie 1.

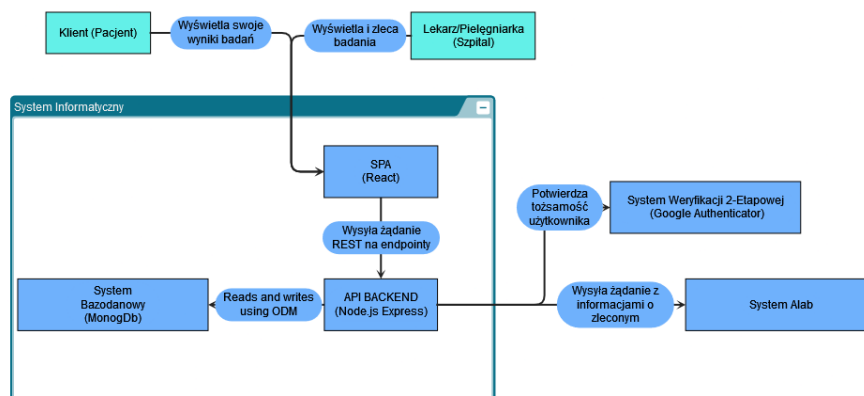


Figure 1: Schemat architektury

Backend jest napisany w języku JavaScript z użyciem środowiska Node.js. Aby zapewnić czytelność i łatwiejszy rozwój kod został podzielony na moduły w poszczególnych plikach. Poniżej znajduje się lista modułów wraz z krótkim opisem.

- **index.js** - główny plik aplikacji, zawiera konfigurację serwera, endpointów oraz uruchamia go.
- **auth.js** - moduł definiujący metody weryfikacyjne dla Passport.js.
- **controllers.js** - moduł zawierający kontrolery, czyli funkcje obsługujące zapytania HTTP.
- **db.js** - moduł odpowiedzialny za połączenie z bazą danych.
- **env.js** - moduł odpowiedzialny za wczytanie zmiennych środowiskowych.
- **models.js** - moduł zawierający modele danych.

Schemat zależności między modułami znajduje się na schemacie 2.

Trzeba zwrócić uwagę na nieoczywistą pośrednią zależność między modułami `auth.js` i `controllers.js`. Również trzeba zauważyć różnice między przedstawionymi zależnościami. Grubszymi strzałkami oznaczono pełny import modułu, a cieńszymi tylko import niektórych funkcji.

Dokładniejszy schemat zależności między funkcjami, z perspektywy API znajduje się na schemacie 3. Przedstawiono na nim jakie funkcje są wywoływane podczas korzystania z API, oraz do którego pliku należą.

3 Baza danych

Do projektu został wybrany silnik bazodanowy `Mongoose.db`.

3.1 Schemat

3.2 Modele

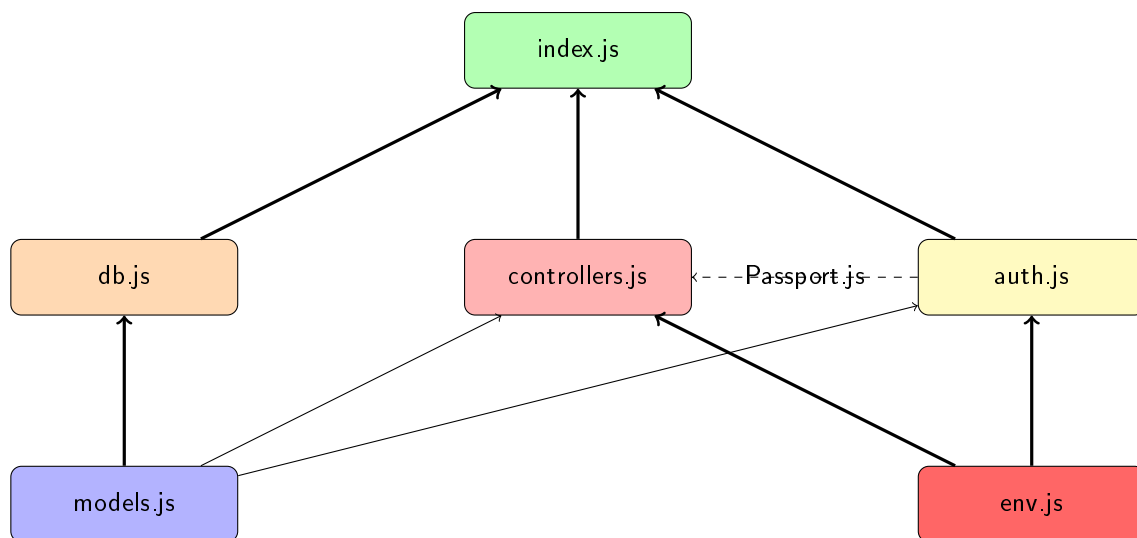


Figure 2: Schemat wymagań

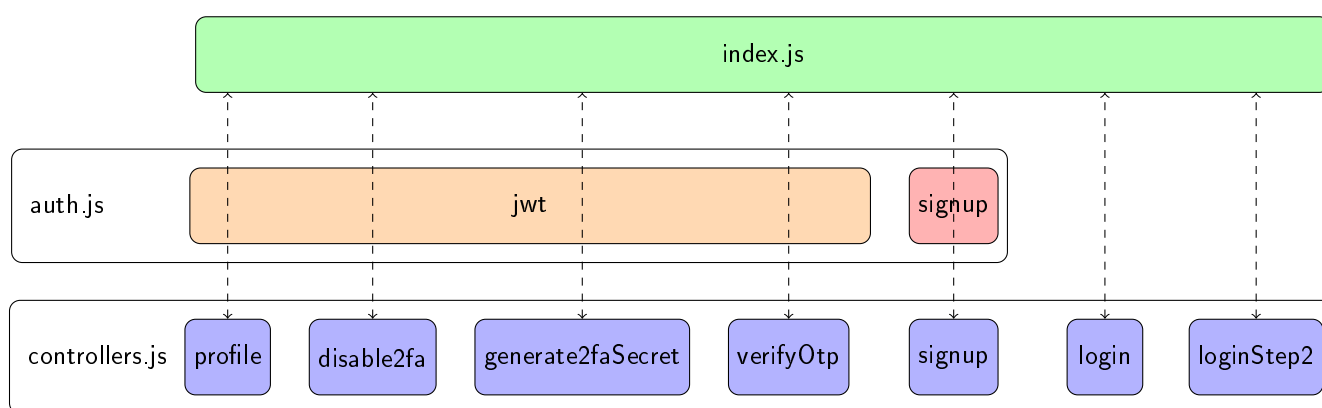


Figure 3: Schemat architektury wewnętrznej API