

Dokumentacja (Backend)

Repozytorium: <https://github.com/Programowanie-Zespolowe-PAJDA/Backend>

Projekt stworzony w ramach przedmiotu "Programowanie Zespołowe"

[Wydział Matematyki i Informatyki UMK](#)

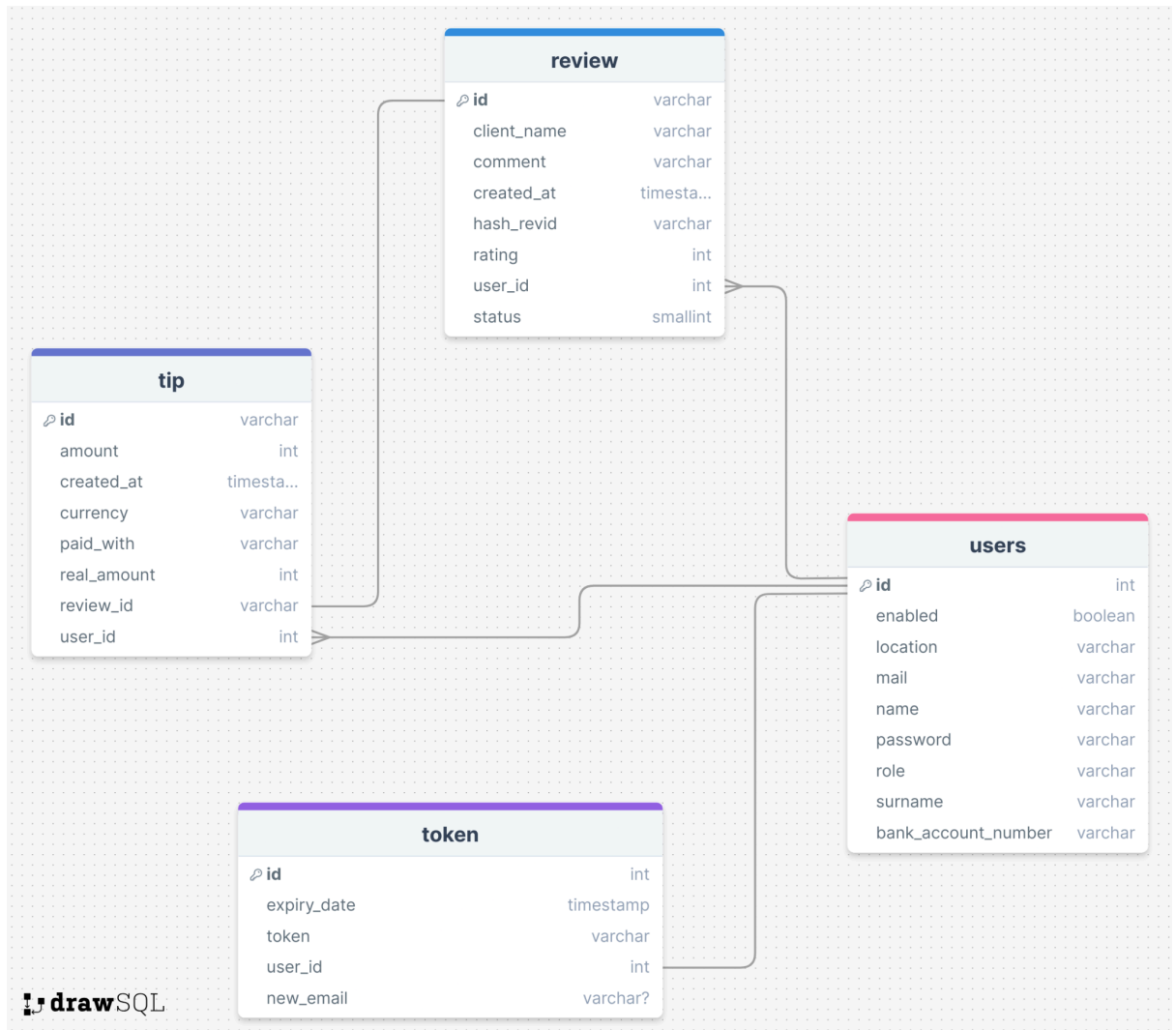
Zespół 2

- Kacper Dąbrowski (Opiekun zespołu)
- Jakub Urbańczyk (Kierownik zespołu)
- Dawid Radzimski (Programista Backend)
- Adrian Polkowski (Programista Backend)
- Adam Zieliński (Programista Frontend)
- Patryk Urbanowicz (Programista Frontend)

1. Opis

Backend został napisany w języku Java z użyciem frameworka Spring Boot. Komunikacja odbywa się przy pomocy endpointów, które są bardziej opisane w punkcie 3. Korzysta również z zewnętrznego API PayU oraz nasz serwer łączy się bazą danych PostgreSQL.

2. Encje



- User

Encja, która głównie służy do przechowywania informacji o danym użytkowniku.

| Nazwa | Typ | Opis |
|-------------------|--------------|--|
| id | Integer | Id użytkownika |
| name | String | Imię użytkownika |
| surname | String | Nazwisko użytkownika |
| password | String | Zahaszowane hasło użytkownika |
| role | String | Rola użytkownika [Admin, User] |
| mail | String | Email użytkownika |
| location | String | Nazwa restauracji użytkownika |
| bankAccountNumber | String | Numer rachunku bankowego użytkownika |
| tipList | List<Tip> | Lista napiwków, które posiada dany użytkownik |
| reviewList | List<Review> | Lista recenzji, które posiada dany użytkownik |
| enabled | boolean | Informacje o tym, czy dany użytkownik jest aktywny, w przeciwnym wypadku musi zweryfikować konto |
| token | Token | Token, który odpowiada za weryfikację emailu oraz jest związany z danym użytkownikiem |

- Token

Encja, która przechowuje informacje dotyczących weryfikacji lub zmiany emailu.

| Nazwa | Typ | Opis |
|------------|---------------|---|
| id | Integer | Id tokenu |
| token | String | Wartość tokenu UUID |
| user | User | Użytkownik, który jest związany z danym tokenem |
| expiryDate | LocalDateTlme | Data wygaśnięcia tokenu, czyli 24 godziny od jego utworzenia |
| newEmail | String | Informacja o nazwie nowego emailu podczas gdy chcemy zmienić istniejący |

- Tip

Encja, która przechowuje informację o napiwkach dla kelnerów.

| Nazwa | Typ | Opis |
|------------|---------------|---|
| id | String | Id napiwku |
| currency | String | Waluta w jakiej zapłacona napiwek [PLN, EUR, USD, GBP, CHF, DKK, SEK] |
| amount | Integer | Wartość napiwku wyrażona w PLN |
| realAmount | Integer | Oryginalna wartość napiwku |
| createdAt | LocalDateTime | Data utworzenia napiwku |
| paidWith | String | Metoda płatności |
| user | User | Użytkownik, który jest powiązany z danym napiwkiem |
| review | Review | Recenzja, która jest powiązana z danym napiwkiem |

- Review

Encja, która przechowuje informacje o recenzjach dla kelnerów.

| Nazwa | Typ | Opis |
|------------|---------------|--|
| id | String | Id recenzji |
| rating | Integer | ocena od 0 do 10 |
| comment | String | komentarz, który maksymalnie może posiadać 1500 znaków |
| createdAt | LocalDateTime | Data utworzenia recenzji |
| clientName | String | Nick osoby wystawiającej napiwek, który maksymalnie może posiadać 30 znaków |
| hashRevId | String | Zahaszowany IP osoby wystawiającej napiwek, użyte w celu wyłączenia możliwości daniu wielu napiwków w krótkim czasie |
| status | Status | Status płatności [PENDING, COMPLETED, WAITING_FOR_CONFIRMATION] |
| user | User | Użytkownik, który jest związany z daną recenzją |
| tip | Tip | Napiwek, który jest związany z daną recenzją |

3. Dokumentacja API

<https://enapiwek-api.onrender.com/swagger-ui/index.html>