

## Laboratorium 5

Node.js, Express, Mongoose, REST API, Nodemailer

### Zadanie 1 – pusty projekt na początek

Tworzymy sobie pustą aplikację Express'a o nazwie lab05 (silnik szablonów generalnie dowolny ale polecam Pug).

### Zadanie 2 – utworzenie prostego API na bazie biblioteki Mongoose

Tworzymy API działające na adresach zaczynających się od '/api/users', które będzie pozwalać:

- na dostęp do listy użytkowników – metoda GET,
- dodawanie nowych użytkowników – metoda POST,
- aktualizację danych znajdujących się już w kolekcji – metoda PUT,
- usuwanie użytkowników – metoda DELETE.

Powinno ono obsługiwać następujące adresy URL

Adresu URL	Metoda	Opis
/api/users	GET	Zwraca tablicę z danymi wszystkich użytkowników, którzy są w bazie.
/api/users/:id	GET	Zwraca dane użytkownika, którego ID podano w adresie URL.
/api/users	POST	Dodaje obiekt z danymi użytkownika przekazanymi w formacie JSON.
/api/users/:id	PUT	Aktualizuje cały dokument z danymi użytkownika, którego adres ID podano w adresie URL. Dane wysyłamy w formacie JSON.
/api/users/:id	DELETE	Usuwa dokument, którego ID podano w adresie URL.

Wykorzystujemy do tego celu bazę MongoDB, bibliotekę Mongoose. Przykład na wykładzie. Każdy z dokumentów powinien przechowywać następujące informacje o użytkowniku, lista pól:

- **username** – login użytkownika, bardzo ważne,
- **password** – hasło użytkownika zapisane jako funkcja skrótu sha1, bardzo ważne
- **email** – adres e-mail użytkownika,
- **admin** – informacja true/false czy użytkownik jest adminem, domyślnie false,
- **active** – informacja true/false czy użytkownik ma aktywne konto, tzn. może się zalogować, domyślnie false, aktywacji konta dokonuje albo administrator albo dzieje się to po kliknięciu w link aktywacyjny wysłany mailem.

Pola **username** oraz **email** mają być unikalne i wymagane, pola **password** oraz **active** mają być wymagane. Zobacz poniższy przykład:

```
var ExampleSchema = new Schema({
  login    : { type : String , unique : true, required : true },
  pass     : { type : String , required : true },
  status   : { type : Boolean, default: true }
});
```

Utworzone API należy przetestować przy pomocy narzędzia REST np. Postman.

### Zadanie 3 – przetestowanie aplikacji

Pod adresem [szablon.zip](#) znajduje się szablon aplikacji napisanej w MEAN Stack, zawierającej najważniejsze elementy takie jak:

- działający system szablonów wykorzystujący silnik Pug,
- działający mechanizm logowania i rejestracji,
- aktywacja konta za pomocą linku przesyłanego w wiadomości e-mail,
- prosty szablon frameworka Bootstrap,
- zwracanie danych użytkowników w formacie JSON, resetowanie użytkowników,
- ograniczenie dostępu tylko do wybranych stron .

Proszę:

- pobrać szablon, rozpakować go,
- doinstalować potrzebne do uruchomienia pakiety,
- uruchomić bazę danych MongoDB i uruchomić aplikację.

Proszę przejrzeć dostępne strony i dostępną funkcjonalność:

- logowanie użytkownika (np. admin, hasło takie samo jak do maszyn wirtualnych),
- resetowanie użytkowników,
- rejestracja nowego użytkownika z aktywacją konta poprzez e-mail – proszę przetestować czy działa,
- sprawdzić dostępność stron tylko dla zalogowanych,
- wyświetlić listę użytkowników w formacie JSON.

Następnie proszę samodzielnie:

- dodać jedną stronę dostępną tylko dla zalogowanych użytkowników (treść dowolna),
- dodać jedną stronę dostępną tylko dla każdego użytkownika (treść dowolna),
- dodać funkcjonalność zadania 2 do tego szablonu i sprawdzić działanie,
- dodać stronę wykorzystującą silnik szablonów PUG.