

Zespół Szkół NR. 10  
Im. Stanisława Staszica

## **Sprawozdanie z aplikacji ProxyRotator**

Autor: Adam Płochocki  
Warszawa 11.12.2024r.

## Spis treści

<b>1. Opis projektu</b>	
● Cel .....	3
● Użyte technologie .....	3
● Funkcjonalności .....	3
<b>2. Struktura bazy danych .....</b>	<b>4</b>
<b>3. Użycie programowania obiektowego</b>	
● Dziedziczenie .....	5
<b>4. Możliwości rozszerzenia .....</b>	<b>5</b>
<b>5. Podsumowanie .....</b>	<b>6</b>

## Opis projektu

### Cel

Celem projektu ProxyRotator jest stworzenie aplikacji do zarządzania i rotacji serwerów proxy. Aplikacja umożliwia użytkownikom dodawanie, usuwanie oraz filtrowanie proxy, a także ich wizualizację na mapie.

### Użyte technologie

- **Java** jako główny język programowania.
- **JavaFX** do tworzenia interfejsu użytkownika.
- **Maven** jako system zarządzania projektami i zależnościami.
- **MySQL** jako baza danych do przechowywania informacji o proxy.
- **JSON** do przetwarzania danych.

### Funkcjonalności

- Logowanie i rejestracja użytkowników to kluczowe elementy platformy, które zapewniają bezpieczny dostęp do różnorodnych usług. Przykładowo, użytkownicy mogą tworzyć unikalne konta, zabezpieczone hasłami, które chronią ich prywatność i dane. Rejestracja może również wymagać potwierdzenia poprzez e-mail, aby zweryfikować tożsamość użytkownika.
- Dodawanie proxy z plików lub ręcznie daje użytkownikom elastyczność w zarządzaniu swoimi preferencjami. Na przykład, użytkownicy mogą importować listę proxy z pliku tekstowego lub dodawać je indywidualnie poprzez interfejs aplikacji. To ułatwia dostęp do różnych serwerów proxy w zależności od potrzeb użytkownika.
- Filtrowanie proxy według adresu i kraju umożliwia precyzyjne dostosowanie ustawień proxy do konkretnych wymagań. Na przykład, użytkownicy mogą wybrać serwery proxy z

określonego kraju lub z konkretnego adresu IP, co pozwala na bardziej skuteczne omijanie geograficznych blokad treści online.

- Wizualizacja proxy na mapie świata zapewnia intuicyjny sposób prezentacji informacji o lokalizacji serwerów proxy. Przykładowo, użytkownicy mogą szybko zidentyfikować, gdzie znajdują się ich preferowane serwery proxy i łatwo porównać ich położenie geograficzne.
- Możliwość usuwania proxy z bazy danych pozwala na sprawną aktualizację i zarządzanie listą serwerów proxy. Na przykład, użytkownicy mogą usuwać niepotrzebne lub nieaktywne proxy, aby utrzymać bazę danych w optymalnym stanie i zapewnić szybki dostęp do najbardziej aktualnych serwerów.

## Struktura bazy danych

proxy-rotator-db users	
id	int
name	varchar(30)
last_name	varchar(40)
email	varchar(50)
password	varchar(70)

proxy-rotator-db proxies	
id	int
user_id	int
proxy_address	varchar(255)
countryName	varchar(50)
countryCode	varchar(3)

### Tabela **users**

Tabela przechowuje dane użytkowników korzystających z aplikacji. Zawiera informacje, takie jak imię, nazwisko, adres e-mail i hasło. Kolumna id pełni rolę klucza głównego i jest unikalnym identyfikatorem każdego użytkownika.

### Tabela **proxies**

Tabela przechowuje dane dotyczące serwerów proxy, takie jak adres IP i port (proxy\_address), kraj (countryName i countryCode) oraz powiązanie z konkretnym użytkownikiem poprzez kolumnę user\_id. Relacja pomiędzy tabelami proxies i users jest realizowana za pomocą klucza obcego user\_id, który odnosi się do kolumny id w tabeli users.

# Użycie programowania obiektowego

## Dziedziczenie

Przykład dziedziczenia klasy bazowej **AccountBaseController.java** przez moją klasę dziedziczącą **LoginController.java** i wywołania metody `switchToMainLayout`

```
1 usage  ▸ Adam *
public class LoginController extends AccountBaseController {

    ▸ Adam *
    @FXML
    private void login() {
        if (email.getText().isEmpty() || password.getText().isEmpty()) return;

        try {
            boolean success = UserService.login(email.getText(), password.getText());

            if (!success) {
                errorLabel.setText("Account with this email or password doesn't exist");
                clearFields();
                return;
            }

            switchToMainLayout();
        } catch (SQLException e) {
            errorLabel.setText("Database error while logging in. Restart the app and try again");
        }
    }

    ▸ Adam *
    @FXML
    private void goRegister() {
        MainApplication.switchScene( sceneName: "RegisterLayout.fxml");
    }
}
```

## Możliwości rozszerzenia

1. Usuwanie proxy
2. Pokazywanie proxy na mapie
3. Sprawdzenie statystyk serwera proxy np. ping

## **Podsumowanie**

Projekt ProxyRotator to aplikacja stworzona w celu zarządzania i rotacji serwerów proxy, która umożliwia użytkownikom łatwe dodawanie, usuwanie oraz filtrowanie proxy według różnych kryteriów. Wykorzystując technologie takie jak Java, JavaFX, MySQL oraz JSON, aplikacja oferuje intuicyjny interfejs użytkownika oraz funkcjonalności, które ułatwiają pracę z proxy.