



DOKUMENTACJA UŻYTKOWNIKA

PROJEKT Z SIECI KOMPUTEROWYCH I
BEZPIECZEŃSTWA

Wyszukiwarka połączeń

Autor:
Krzysztof BARAŃSKI

Nadzorujący:
dr Edward SZCZYPKA

Styczeń 2015

Spis treści

1	Wymagania	2
1.1	Aplikacja kliencka	2
1.2	Aplikacja serwerowa	2
1.3	Aplikacja przewoźnika	2
2	Konfiguracja	2
2.1	Aplikacja kliencka	2
2.2	Aplikacja serwerowa	3
2.3	Aplikacja przewoźnika	3
3	Uruchamianie	3
3.1	Aplikacja kliencka	3
3.2	Aplikacja serwerowa	3
3.3	Aplikacja przewoźnika	3
4	Obsługa	4
4.1	Aplikacja kliencka	4
4.2	Aplikacja serwerowa	4
4.3	Aplikacja przewoźnika	4
5	Odinstalowywanie	4
5.1	Aplikacja kliencka	4
5.2	Aplikacja serwerowa	4
5.3	Aplikacja przewoźnika	4
6	Lista obsługiwanych miast	5

1 Wymagania

1.1 Aplikacja kliencka

System operacyjny

Linux

Testowany na Linux debian 3.2.0-4-amd64.

Środowisko

Java SE Development Kit w wersji 1.8

Bash

Sprzęt

Połączenie z internetem.

1.2 Aplikacja serwerowa

System operacyjny

Linux

Testowany na Linux debian 3.2.0-4-amd64.

Środowisko

Java SE Development Kit w wersji 1.8

Bash

Sprzęt

Połączenie z internetem.

1.3 Aplikacja przewoźnika

System operacyjny

Linux

Testowany na Linux debian 3.2.0-4-amd64.

Środowisko

Java SE Development Kit w wersji 1.8

SQLite3

Bash

Sprzęt

Połączenie z internetem.

2 Konfiguracja

2.1 Aplikacja kliencka

Przed uruchomieniem aplikacji należy ją skompilować - służy do tego skrypt *compile* znajdujący się w katalogu *client*. Następnie należy uruchomić skrypt *setup* znajdujący się w tym samym katalogu w celu skonfigurowania aplikacji.

Podczas konfiguracji wymagane będzie podanie adresu serwera i portu, na którym serwer nasłuchuje.

2.2 Aplikacja serwerowa

Przed uruchomieniem serwera należy go skompilować wywołując skrypt *compile* znajdujący się w katalogu *server*. Następnie należy skonfigurować serwer za pomocą skryptu *setup*. Podczas konfiguracji wymagane będzie podanie portu, na którym będzie nasłuchiwał serwer.

Przewoźników można wyświetlać, dodawać i usuwać za pomocą skryptu *carriers*. Uruchomienie bez żadnych parametrów wyświetla pomoc. Można to również zrobić ręcznie, edytując plik *server/database/carriers.db*.

2.3 Aplikacja przewoźnika

Przed uruchomieniem aplikacji przewoźnika należy ją skompilować wywołując skrypt *compile* znajdujący się w katalogu *carrier*. Następnie należy skonfigurować aplikację za pomocą skryptu *setup*. Podczas konfiguracji wymagane będzie podanie nazwy przewoźnika (np. nazwa firmy przewozowej), numeru portu, na którym będzie nasłuchiwał serwer przewoźnika oraz nazwę bazy danych, która zostanie stworzona w celu przechowywania rozkładu jazdy.

W katalogu *carrier* znajduje się również skrypt *random_timetable*, który służy do wypełniania bazy danych losowym rozkładem jazdy. Podczas działania, skrypt zapyta o liczbę rekordów do wygenerowania. Skrypt można wywoływać zarówno przed, jak i po skonfigurowaniu aplikacji wielokrotnie - za każdym razem zostanie dodane do bazy tyle rekordów ile zostało podane podczas działania skryptu.

3 Uruchamianie

3.1 Aplikacja kliencka

Aby uruchomić skompilowaną i skonfigurowaną aplikację klienta należy w katalogu *client* uruchomić skrypt *run*.

3.2 Aplikacja serwerowa

Aby uruchomić skompilowaną i skonfigurowaną aplikację serwerową należy w katalogu *server* uruchomić skrypt *run*.

3.3 Aplikacja przewoźnika

Aby uruchomić skompilowaną i skonfigurowaną aplikację przewoźnika należy w katalogu *carrier* uruchomić skrypt *run*.

4 Obsługa

4.1 Aplikacja kliencka

Po uruchomieniu aplikacji, w oknie konsoli pojawi się logo projektu oraz początek sesji wyszukiwania połączenia.

Po słowie **FROM:** oraz **TO:** należy wpisać odpowiednio: miasto początkowe i końcowe podróży, używając wyłącznie liter alfabetu angielskiego, pauzy lub spacji. Lista dostępnych miast znajduje się na końcu dokumentacji.

Następnie po słowie **DATE:** należy wpisać datę początku podróży w formacie *YYYY-MM-DD*, np. *2015-01-26*.

Na koniec po słowie **TIME:** należy wpisać godzinę początku podróży w formacie *HH:MM*, np. *07:56*.

Jeżeli istnieje połączenie podane w zapytaniu, aplikacja znajdzie je, a następnie wypisze kolejne odcinki podróży.

Aby wyszukać kolejne połączenie, należy po linii: *"Would you like to find a new connection? If so, type 'yes':"* wpisać słowo *"yes"* bez cudzysłowu. Każde inne słowo, w szczególności pusta linia, kończy działanie aplikacji.

4.2 Aplikacja serwerowa

Obsługa serwera polega jedynie na sprawdzaniu, czy działa. Serwer jest odporny na różnego rodzaju błędy i wyjątki, aczkolwiek może się zdarzyć nieprzewidziany dotąd błąd, po którym aplikacja się zatrzyma. W celu zatrzymania aplikacji, w obecnej wersji należy wysłać jej jakiś mocny sygnał np. SIGINT (w konsoli zazwyczaj Ctrl+C).

4.3 Aplikacja przewoźnika

Obsługa aplikacji przewoźnika jest analogiczna do obsługi aplikacji serwera.

5 Odinstalowywanie

5.1 Aplikacja kliencka

W celu odinstalowania (lub też przywrócenia stanu aplikacji do stanu wyjściowego) należy uruchomić skrypt *uninstall* w odpowiednim podkatalogu projektu.

5.2 Aplikacja serwerowa

Patrz: Aplikacja kliencka

5.3 Aplikacja przewoźnika

Patrz: Aplikacja kliencka

6 Lista obsługiwanych miast

Poniżej znajduje się lista miast, które są aktualnie obsługiwane przez wyszukiwarke:

- Białystok
- Bielsko-Biala
- Bydgoszcz
- Bytom
- Chelm
- Elblag
- Gdansk
- Gliwice
- Gorzow Wielkopolski
- Kalisz
- Katowice
- Kielce
- Koszalin
- Krakow
- Legnica
- Lublin
- Lodz
- Nowy Sacz
- Olsztyn
- Opole
- Pila
- Poznan
- Rzeszow
- Slupsk
- Suwalki
- Szczecin
- Swinoujscie
- Tarnow

- Torun
- Walbrzych
- Warszawa
- Wloclawek
- Wroclaw
- Zakopane
- Zamosc
- Zielona Gora