

Projekt 1: baza danych typu NoSQL

Temat: Rozszerzenie aplikacji TrainsOnline o raportowanie informacji o przebiegu trasy pociągu w bazie MongoDB

Białystok, 4.05.2020

Wykonujący:
Adam Bajguz
Michał Kierzkowski
Grupa PS3

Prowadzący:
dr hab. inż. Agnieszka Drużdżel prof. PB

Spis treści

1. Wprowadzanie	3
2. Funkcjonalności	4
3. Projekt i implementacja aplikacji	5
3.1. Model danych	5
3.2. Architektura aplikacji i wykorzystane technologie	6
3.2.1. Po stronie serwera	6
3.2.2. Po stronie klienta	12
2.3. Profil użytkownika końcowego	22

1. Wprowadzanie

Zakresem projektu było rozszerzenie aplikacji TrainsOnline. Istniejący system pozwalał na przechowywanie bazy stacji kolejowych i tras, na jakich przemieszczają się pociągi, a także umożliwił zakup biletów na daną trasę oraz pobranie go w formacie PDF. System rozszerzono o trzy funkcjonalności opierające się o nierelacyjną bazę danych:

1. logger pociągu,
2. cache
3. analityka.

Pierwsze z rozszerzeń tj. logger pociągu pozwala na gromadzenie danych z trasy pociągu np. data i położenie geograficzne wszystko będzie zapisywane w bazie MongoDB. Po każdej trasie do bazy generowany jest raport zawierający takie dane jak średnia prędkość pociągu. W tym celu należało rozszerzyć Webservice dostarczający wszystkie funkcjonalności w API oraz aplikację kliencką umożliwiającą łatwy dostęp do zasobów serwisowych, a także stworzyć trzecią aplikację symulującą pociąg.

Drugie rozszerzenia, rozwiązano problem z dostępem do danych wymagających złożonych operacji na bazie lub generowania plików PDF poprzez zaimplementowanie mechanizmu cechowania danych opartego na nierelacyjnej bazie typu klucz-wartość.

Ostatnie z rozszerzeń, tj. rozszerzenie WebSerwisu o analitykę rozwiązano umożliwiło zbieranie danych analitycznych dotyczących ruchu sieciowego do WebAPI w taki sposób aby możliwa była analiza ilości wejść (zapytań) do poszczególnych adresów w serwisie, a także zebranie informacji o stosowanych klientach (np. przeglądarek) oraz wykrycie zapytań od hakerów.

2. Funkcjonalności

System posiada funkcjonalności podstawowe takie jak:

- rejestracja klienta;
- logowanie na serwer;
- wyświetlanie bazy tras kolejowych (Miasto od , Miasto do, dzień, godzina);
- wyszukiwarkę tras;
- kupno biletu;
- odbiór potwierdzenia kupna w formacie PDF;
- sprawdzenie rezerwacji na podstawie podanego numeru biletu;
- dodawanie, usuwanie, modyfikowanie tras oraz stacji (funkcjonalność dostępna na Swagger lub SOAP UI tylko dla użytkowników z prawami administratora)
- logowanie zdarzeń z trasy pociągu,
- tworzenie raportów
- prosta analityka/statystyki wykorzystania zasobów z API

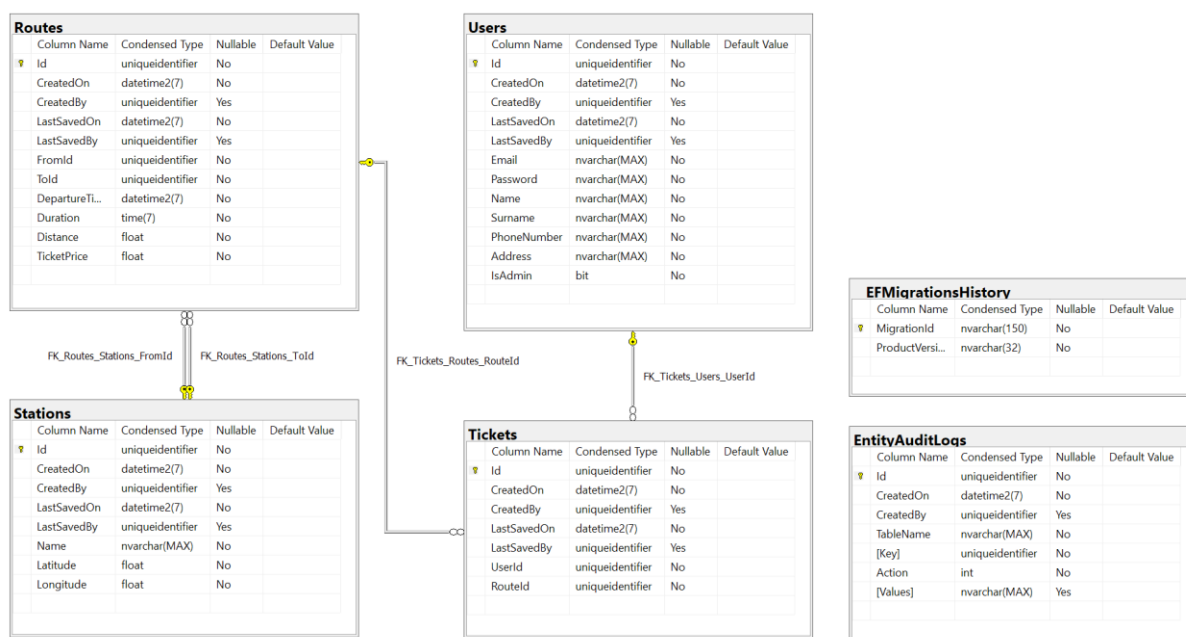
a także funkcjonalności dodatkowe:

- przypominanie hasła na podany podczas rejestracji adres email;
- szczegóły danej stacji kolejowej;
- prezentowanie na mapie położenia stacji;
- autoryzacja oparta na JSON Web Token;
- generowanie kodów QR które służą do dodania wydarzenia do kalendarza o planie podróży oraz weryfikacji biletu
- dziennik logów zapisywany wewnętrznie na serwerze przy pomocy loggeraSerilog, a także możliwość zdalnego przechowywania logów na platformie Sentry
- wykorzystanie algorytmu Jaro-Winkler podczas wyszukiwania
- zgodność z OpenAPI 3.0.1
- wykorzystanie Swagger, SwaggerUI, ReDoc

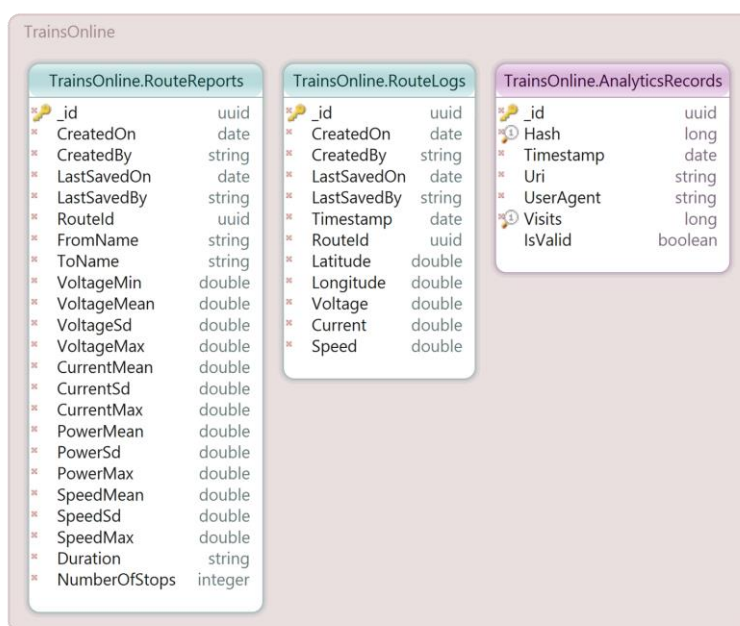
3. Projekt i implementacja aplikacji

3.1. Model danych

Aplikacja wykorzystuje dwa modele danych: relacyjny i nierelacyjny. Model relacyjny składa się z 6 tabel, zawierających dane o użytkowniku, trasie, biletach, stacjach, historii zmian w elementach bazy (entity audit log).



Z kolei model nierelacyjny składa się z 3 kolekcji przechowywanych w MonogDb, przechowujących logi z pociągu, raporty z tras pociągów, a także dane analityczne związane z ruchem po WebAPI i całej aplikacji serwerowej.



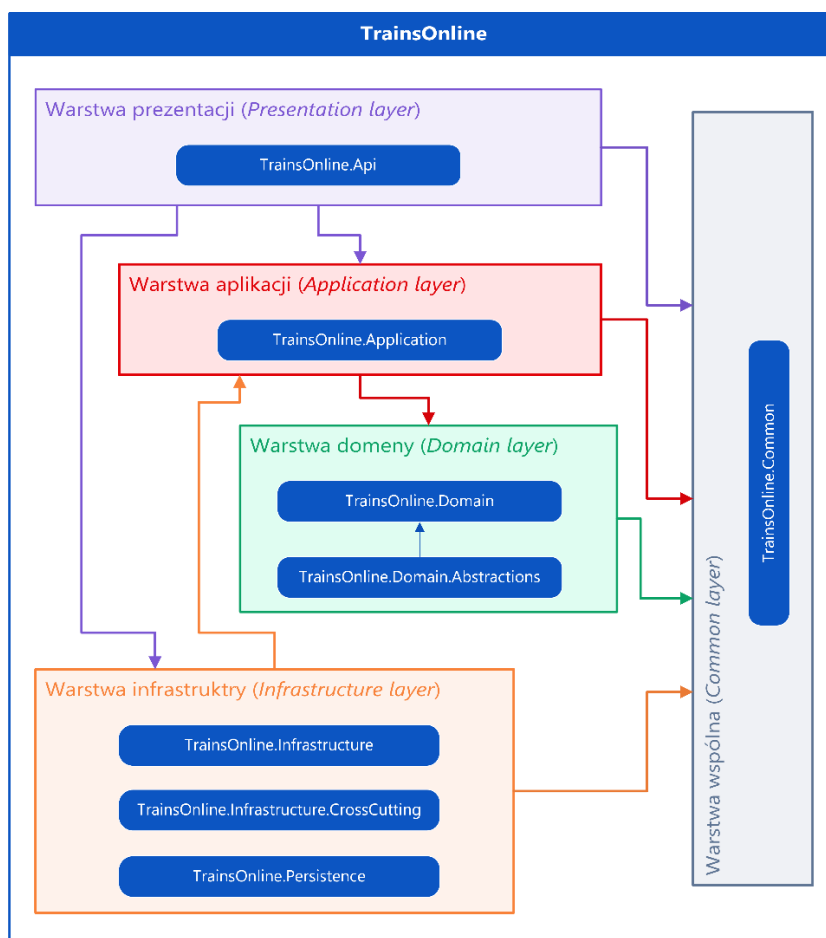
3.2. Architektura aplikacji i wykorzystane technologie

3.2.1. Po stronie serwera

Moduł TrainsOnline został zaimplementowany z użyciem podejścia określonego jako Clean Architecture, którego celem jest zapewnienie wysokiej skalowalności aplikacji i uniknięcie tworzenia monolitycznej architektury. Podejście to zrealizowano przy pomocy wzorca architektonicznego Domain Driven Design (DDD). Do stworzenia serwera aplikacji użyto: C# 8.0, NET Core 3.1.0, ASP.NET Core MVC, Microsoft SQL Server 2017. W implementacji zastosowano także następujące biblioteki:

Nazwa	Wersja
AutoMapper.Extensions.Microsoft.DependencyInjection	7.0.0
AspNetCore.HealthChecks.MongoDB	3.1.1
CommandLineParser	2.8.0
Ceras	4.1.7
CSharpVitamins.ShortGuid	1.0.1
FluentValidation.AspNetCore	8.6.2
F23.StringSimilarity	3.1.0
GemBox.Document	31.0.1175
System.Drawing.Common	5.0.0-preview.3.20214.6
SoapCore	1.1.0-alpha
QRCoder	1.3.9
Humanizer.Core	2.8.26
Sciensoft.Hateoas	3.1.0
MediatR	8.0.1
MediatR.Extensions.Microsoft.DependencyInjection	8.0.0
Microsoft.AspNetCore.Authentication.JwtBearer	3.1.3
Microsoft.AspNetCore.Mvc	2.2.0
Microsoft.EntityFrameworkCore	3.1.3
Microsoft.EntityFrameworkCore.Design	3.1.3
Microsoft.EntityFrameworkCore.InMemory	3.1.
Microsoft.EntityFrameworkCore.SqlServer	3.1.3
Microsoft.EntityFrameworkCore.Tools	3.1.3
Microsoft.Extensions.CommandLineUtils	1.1.1
Microsoft.Extensions.Configuration.EnvironmentVariables	3.1.0
Microsoft.Extensions.Configuration.Json	3.1.0
Microsoft.NET.Test.Sdk	16.4.0
MongoDB.Bson	2.10.4
MongoDB.Driver	2.10.4
MongoDB.Driver.Core	2.10.4
Newtonsoft.Json	12.0.3
Sentry.Serilog	2.1.3
Serilog	2.9.0
Serilog.AspNetCore	3.4.0-dev-00168

Nazwa	Wersja
Serilog.Enrichers.Memory	1.0.4
Serilog.Enrichers.Process	2.0.1
Serilog.Enrichers.Thread	3.2.0-dev-00747
Serilog.Exceptions	5.5.0
Serilog.Sinks.Async	1.4.0
Serilog.Sinks.Console	3.1.1
Serilog.Sinks.File	4.1.0
SerilogTimings	2.3.0
Shouldly	3.0.2
Swashbuckle.AspNetCore	5.4.1
Swashbuckle.AspNetCore.Newtonsoft	5.4.1
Swashbuckle.AspNetCore.ReDoc	5.4.1
Swashbuckle.AspNetCore.Annotations	5.4.1
System.IdentityModel.Tokens.Jwt	5.6.0
System.Data.HashFunction.Core	2.0.0
System.Data.HashFunction.MurmurHash	2.0.0
xunit	2.4.1
xunit.runner.visualstudio	2.4.1



A → B oznacza, że A posiada referencję na B

Wzorzec DDD wymaga również podziału projektu na następujące warstwy:

- a) wspólną – obejmuje wszystkie kwestie przekrojowe, czyli klasy i interfejsy wspólne dla różnych warstw;
- b) prezentacji – zawiera kontrolery, punkty dostępowe do REST API oraz aplikację Swagger;
- c) aplikacji – zawiera całą logikę. Warstwa jest zależna wyłącznie od warstwy domeny, są w niej zdefiniowane interfejsy, np. repozytorium lub jednostki pracy (Unit of Work), których implementacja znajduje się w innych warstwach. Oznacza to, że DDD wykorzystuje wzorzec odwrócenia sterowania (inversion of control). Przykładowo, jeśli aplikacja musi uzyskać dostęp do usługi powiadomień, do warstwy aplikacji zostanie dodany nowy interfejs, a implementacja zostanie utworzona w ramach warstwy infrastruktury;
- d) domeny – zawiera dane, wyliczenia, wyjątki i logikę specyficzną dla domeny;
- e) infrastruktury – zawiera klasy dostępu do zasobów zewnętrznych, takich jak: baza danych, systemy plików i poczta elektroniczna. Klasy te opierają się na interfejsach zdefiniowanych w warstwie aplikacji. W warstwie znajduje się również podwarstwa Persistence, w której zawarta jest cała konfiguracja bazy danych.

W projekcie z uwagi na małą liczbę obiektów domenowych, a tym samym tabel w bazie danych, zastosowano podejście grupowania klas w:

- i. projekty (biblioteki) w oparciu o warstwy określone przez DDD (package by layer),
- ii. funkcjonalności (package by feature) na poziomie folderów i przestrzeni nazw w poszczególnych warstwach.

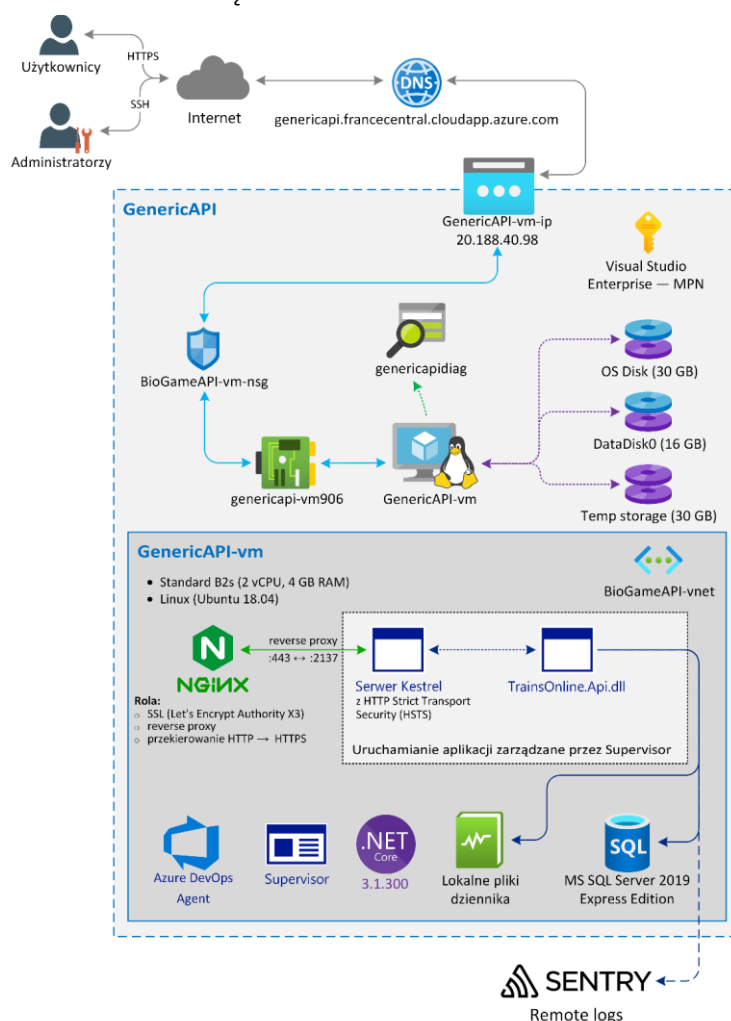
Rozwiązanie te jest wystarczające do zapewnienia czystego kodu oraz architektury w przypadku tego modułu. Ponadto nie wymaga implementowania m.in. koncepcji ograniczonych kontekstów (bounded contexts).

Moduł TrainsOnline został wdrożony na platformie Azure na maszynie wirtualnej typu Standard B2s, wyposażonej w dwa wirtualne procesory oraz 4 GB pamięci operacyjnej. Do maszyny podpięte są dwa dyski SSD typu Premium SSD, charakteryzujące się maksymalną liczbą operacji na sekundę (IOPS) równą 120 i przepływnością maksymalną na poziomie 25 MB/s, oraz jeden dysk tymczasowy o maksymalnej wartości IOPS równej 1600 i maksymalnej przepływności 15 MB/s :

- OS Disk – dysk systemowy o pojemności 30 GB z systemem operacyjnym Ubuntu w wersji 18.04 oraz aplikacjami: .NET Core 3.1.0 Runtime, Microsoft SQL Server 2017 Express Edition, nginx, Azure DevOps Agent, supervisor;
- DataDisk0 – dysk o pojemności 16 GB zawierający aplikację "" oraz pliki dziennika aplikacji, tzw. logi;
- Temp storage – dysk o pojemności 30 GB służący do przechowywania danych tymczasowych, np. plików tymczasowych serwera SQL, który ulega wyczyszczeniu m.in. po ponownym uruchomieniu maszyny wirtualnej.

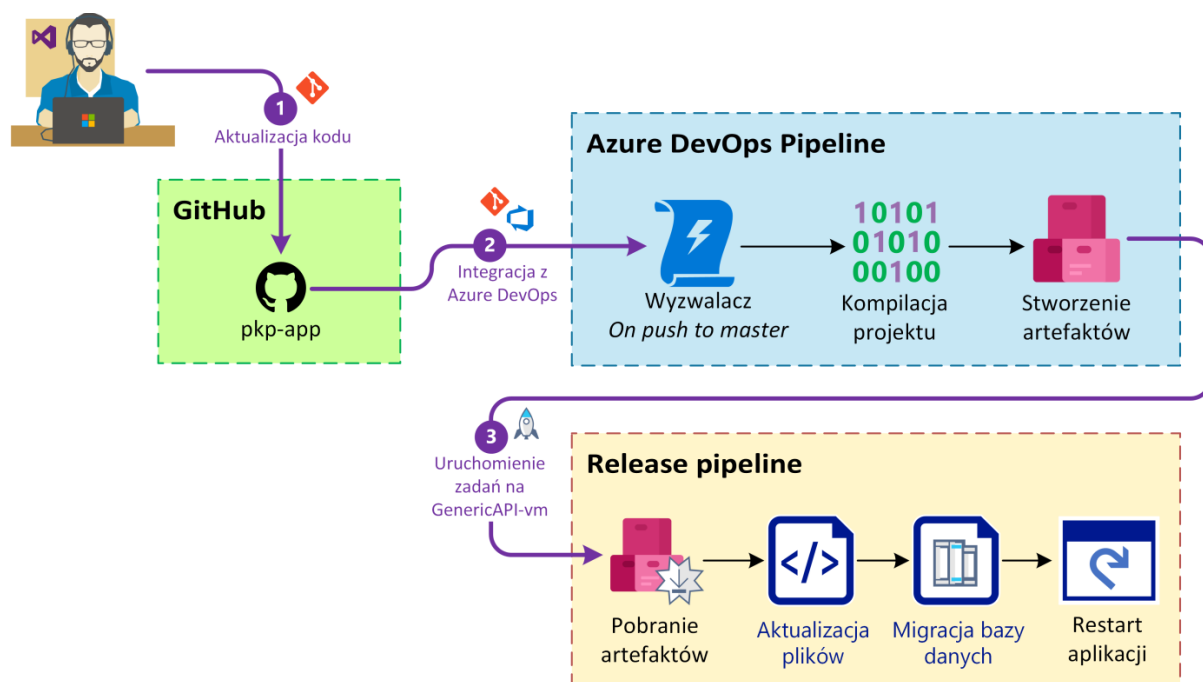
Usługa „TrainsOnline” zaimplementowana została jako usługa sieciowa (Web Service) z użyciem protokołu SOAP. Usługa dostępna jest poprzez protokół HTTPS. Usługa dostępna jest pod adresem: <https://genericapi.francecentral.cloudapp.azure.com/api>, który prowadzi do strony zawierającej wszelkie dostępne endpointy na serwerze oraz dokumentację za pomocą swaggersa.

Zdalny dostęp do maszyny wirtualnej odbywa się poprzez protokół Secure Shell (SSH) z wykorzystaniem klucza RSA jako metody uwierzytelniania. Do połączenia się z maszyną wirtualną wykorzystano program Termius z pakietu GitHub Student Developer Pack. Dostarcza on wieloplatformowy terminal ze zintegrowanym klientem SSH. Dostęp przez użytkowników (klientów) do API odbywa się za pośrednictwem protokołu Hypertext Transfer ProtocolSecure (HTTPS). Działanie aplikacji można sprawdzić pod adresem <https://genericapi.francecentral.cloudapp.azure.com/soap-api>. Certyfikat na potrzeby protokołu HTTPS uzyskano za pomocą programu Certbot, urzędem certyfikacji jest Let'sEncrypt Authority X3. Aplikacja TrainsOnline uruchomiona jest na porcie 2137 za pomocą serwera Kestrel, który został użyty jako serwer graniczny. Komunikacja aplikacji z Internetem odbywa się za pomocą serwera nginx skonfigurowanego jako reverseproxy pomiędzy portami o numerach 443 oraz 2137. Aplikacja komunikuje się również z platformą Sentry, na której zapisywane są wszelkie zdarzenia błędów.



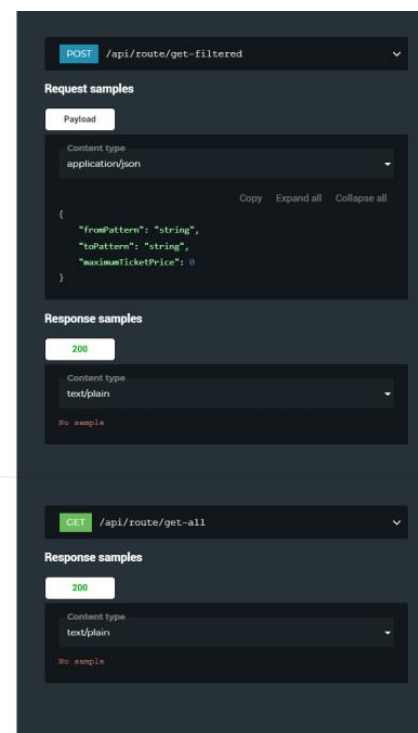
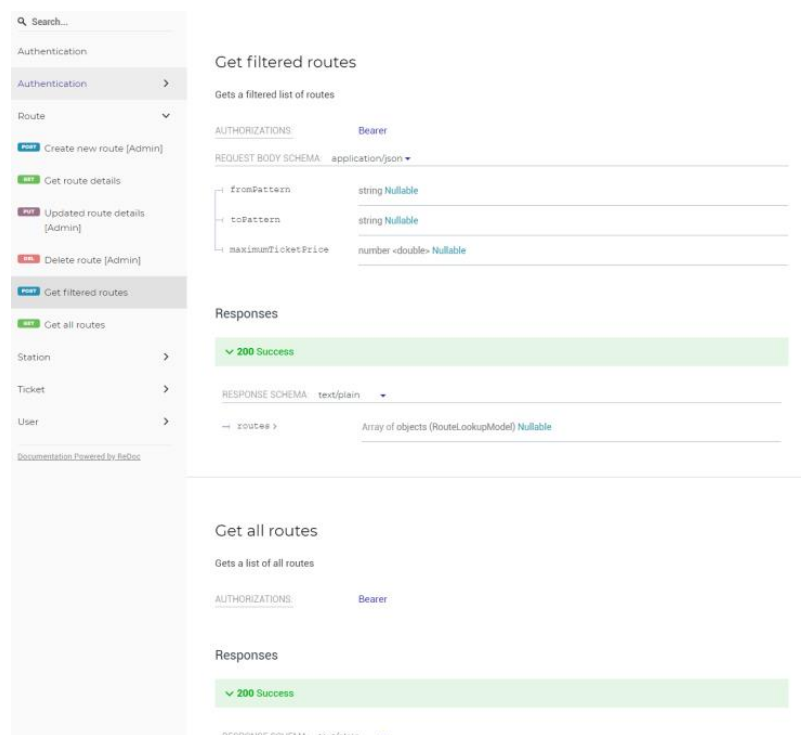
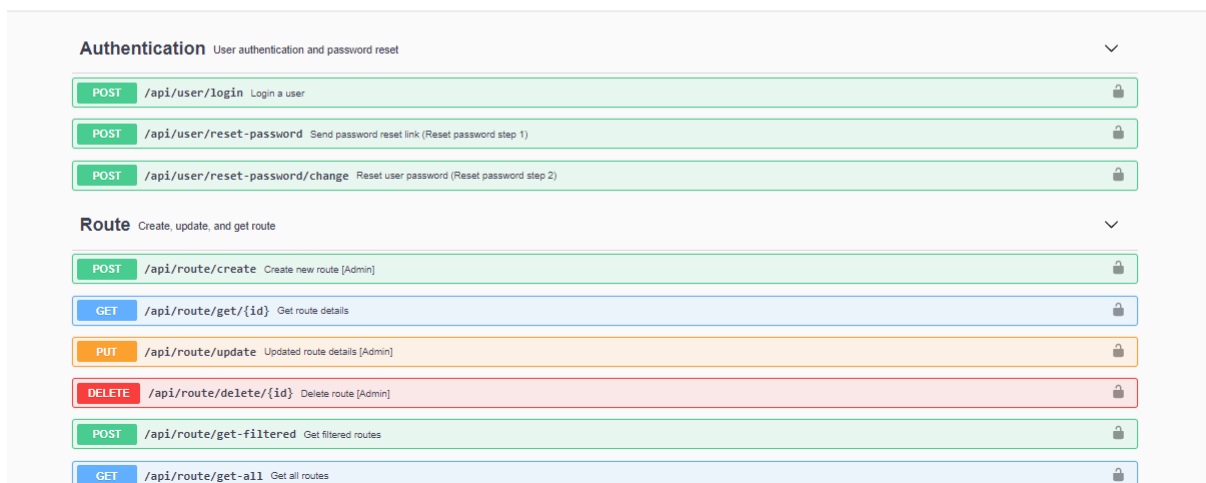
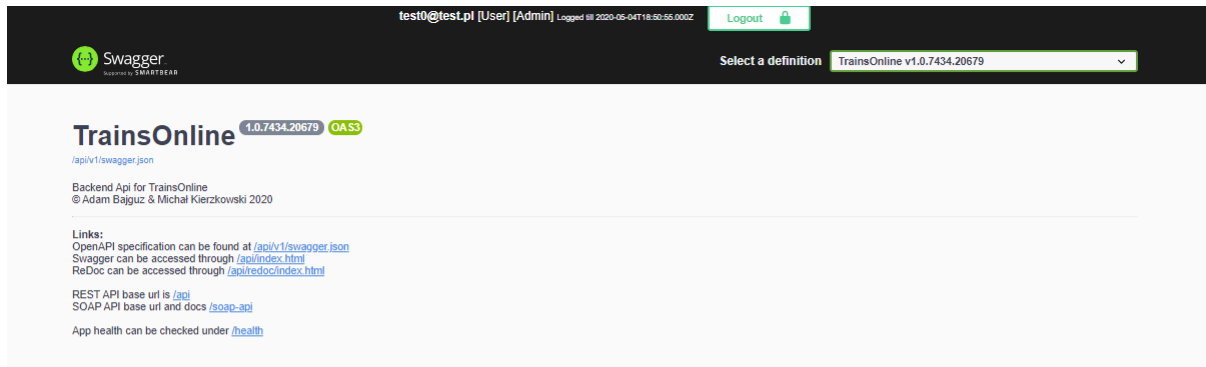
Zastosowanie ciągłej integracji (continuousintegration) oraz ciągłego dostarczania (continuousdelivery) umożliwiło łatwiejszą obsługę środowiska produkcyjnego TrainsOnline. Aktualizacje plików wykonywalnych, a także modyfikacje struktury bazy danych (migracje) odbywają się przy pomocy dwóch potoków:

1. uruchamianego na AzureDevOps w momencie pojawienia się zmian w gałęzi master repozytorium pkp-app, który buduje aplikację oraz przygotowuje paczkę z artefaktami;
2. uruchamianego na maszynie wirtualnej z poziomu AzureDevOps (z udziałem programu AzureDevOps Agent), którego zadaniem jest m.in. pobranie paczki artefaktów i aktualizacja plików.



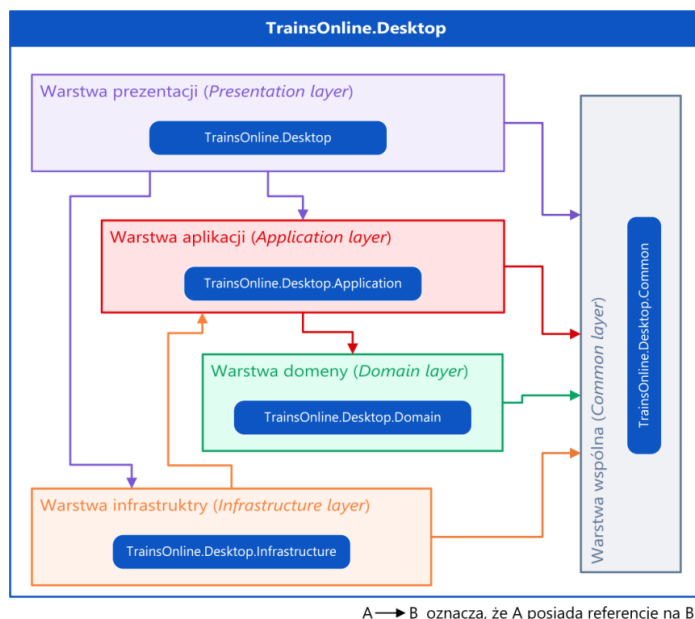
TrainsOnline

WebAPI zostało udokumentowane za pomocą Swaggera i ReDoc. Ponadto Swagger pozwala na łatwe testowanie i dostęp do funkcjonalności nawet przez niedoświadczonych użytkowników.

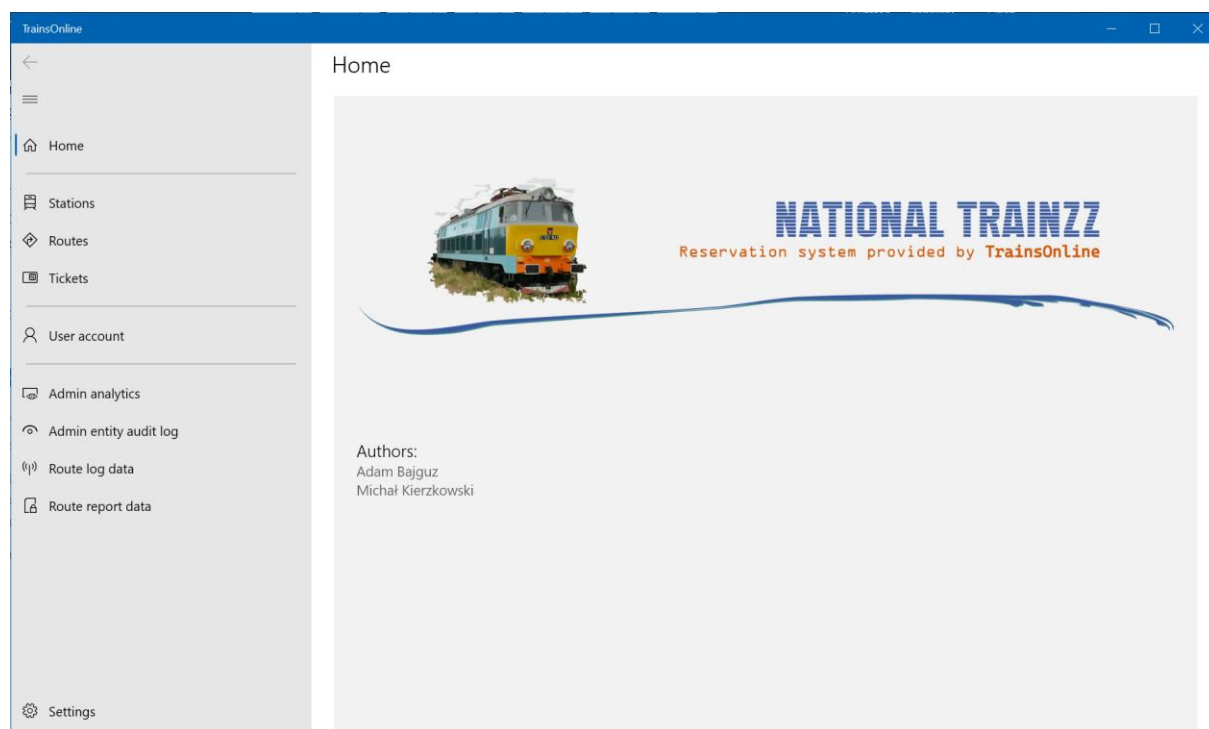


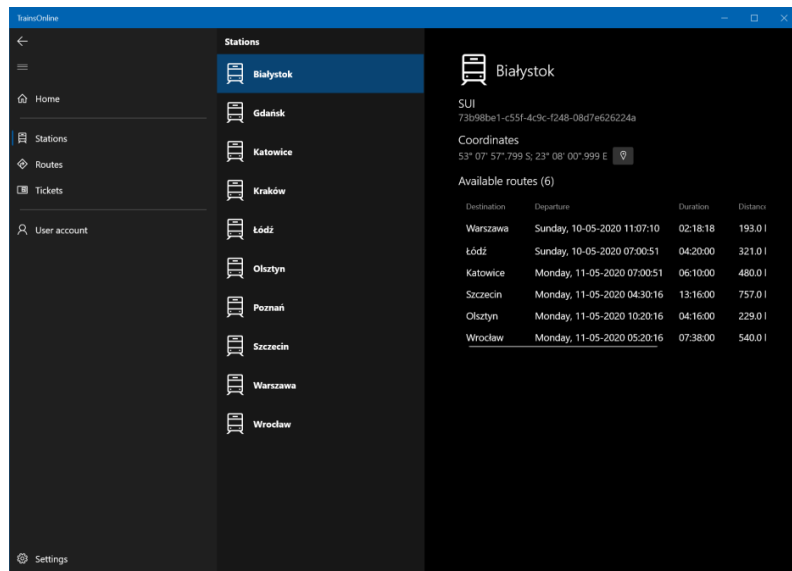
3.2.2. Po stronie klienta

Aplikacja TrainsOnline.Desktop jest aplikacją UWPzawierającą całą logikę niezbędną do komunikacji z serwerem rezerwacji biletów kolejowych. Technologia UWP została zastosowana w celu implementacji łatwego i przyjaznego dla użytkownika interfejsu okienkowego o spójnym wyglądzie. Architektura aplikacji wykorzystuje wzorec MVVM w celu zapewnienia oddzielenia interfejsu użytkownika od logiki biznesowej. Do implementacji MVVM zastosowano bibliotekę Caliburn.Micro. Ponadto w implementacji zastosowano podejście DDD – podobnie jak w aplikacji serwerowej. Jedyną różnicą jest brak warstwy Persistence.



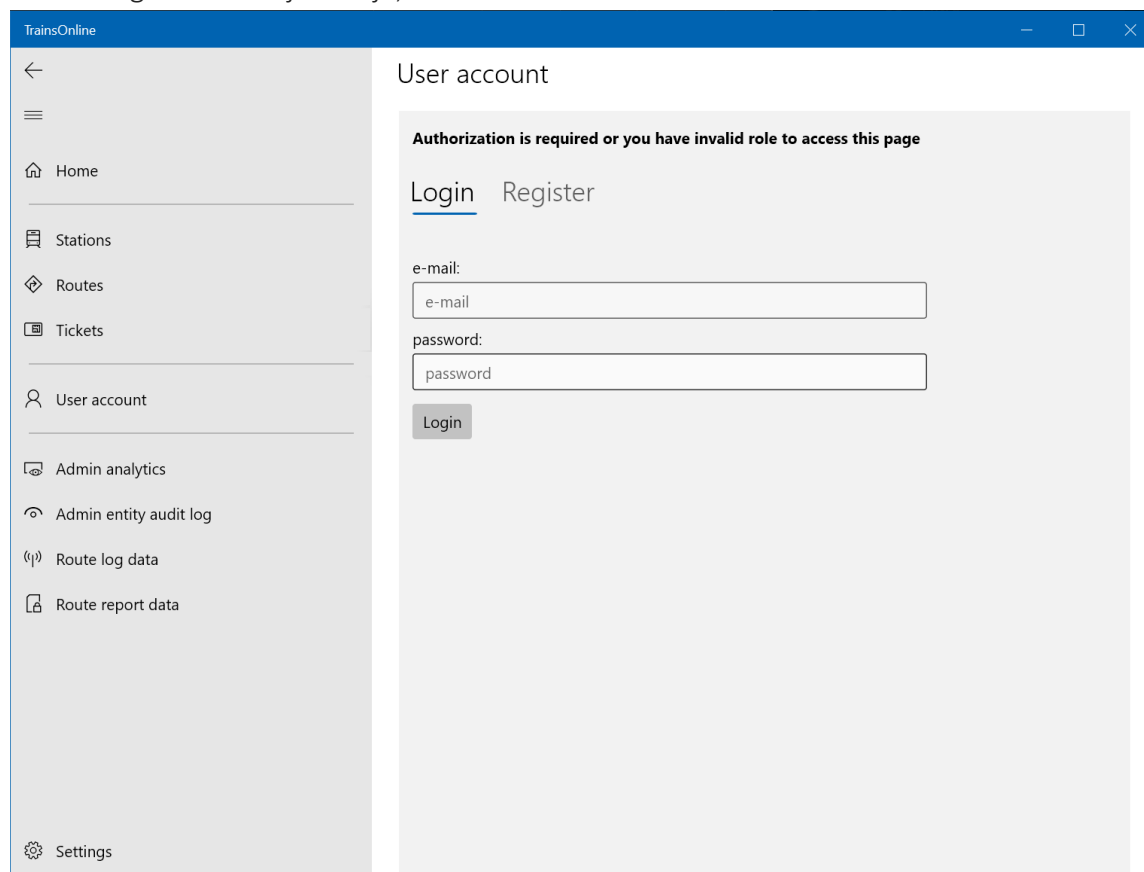
Program kliencki jest prostą i intuicyjną w obsłudze responsywną aplikacją okienkową powstałą w technologii Universal Windows Platform (UWP) i działa zarówno na komputerach, jak i telefonach z system Windows.





Posiada ona obsługę funkcjonalności API takich jak:

- logowanie i rejestracja;



TrainsOnline

←

≡

Home

Stations

Routes

Tickets

User account

Admin analytics

Admin entity audit log

Route log data

Route report data

Settings

User account

Authorization is required or you have invalid role to access this page

Login

Register

e-mail:

e-mail

password:

password

name:

Name

surname:

Surname

phone number:

Phone number

address:

Address

Register

- analityka

14

Admin analytics			
Timestamp	Url	Visits	Id
Url: /api/index.html (7 items)			
Saturday, 30-05-2020	/api/index.html	3	(9AFCDA89-021C-4A48-B1D6-C0AA8FC42ACE)
Sunday, 31-05-2020	/api/index.html	1	(8E5D773F-CA11-41C0-B177-981F07039D50)
Monday, 01-06-2020	/api/index.html	1	(8F4FADA7-57DC-4BF1-A1AA-4CA0860685B8)
Thursday, 04-06-2020	/api/index.html	4	(07B8ADA1-A474-48F7-BFF2-F5218B60C03F)
Friday, 05-06-2020	/api/index.html	10	(937C9FE7-BA98-409F-8347-1C4A5D860BD0)
Saturday, 06-06-2020	/api/index.html	1	(FB477759-D0C5-441A-8779-480BFEF5C502)
Saturday, 13-06-2020	/api/index.html	12	(54740A87-83DB-4C5C-9A49-E2595AEF1F52)
Url: /api/v1/swagger.json (7 items)			
Saturday, 30-05-2020	/api/v1/swagger.json	4	(B203A500-F9A0-4CDE-ACF6-C8EEA5D70592)
Sunday, 31-05-2020	/api/v1/swagger.json	1	(FF76FA59-D074-473F-BE2D-078F33B18A52)
Monday, 01-06-2020	/api/v1/swagger.json	1	(36B94306-9C0C-4DBF-BEA1-FC51356E588B)
Thursday, 04-06-2020	/api/v1/swagger.json	4	(8FAD6ADC-06CE-4AD4-8F23-2A47E29B47CF)
Friday, 05-06-2020	/api/v1/swagger.json	10	(90D1035B-BC77-4ECC-AD3E-395C237A2306)
Saturday, 06-06-2020	/api/v1/swagger.json	1	(3451A956-9487-43EB-B48A-8D6D6A9BF6C5)
Saturday, 13-06-2020	/api/v1/swagger.json	14	(784E4873-1A69-41BA-A55C-7E60D81A8280)
Url: /api/favicon-32x32.png (4 items)			
Saturday, 30-05-2020	/api/favicon-32x32.png	2	(A5C9BEDB-E6E9-4951-97E9-ABD997162A9E)
Thursday, 04-06-2020	/api/favicon-32x32.png	2	(E4A12418-6C87-4F02-8FC7-BC25B4723B83)
Friday, 05-06-2020	/api/favicon-32x32.png	5	(EED259D5-6653-4FA8-AB92-F2AB239C6D46)
Saturday, 13-06-2020	/api/favicon-32x32.png	4	(EA4FA752-F743-4E1A-8818-6227A7C8A879)
Url: /api/user/login (7 items)			
Saturday, 30-05-2020	/api/user/login	2	(966C94D2-476E-4A09-A4D0-C93259170468)
Sunday, 31-05-2020	/api/user/login	2	(D7484BA3-0CA6-4822-839A-C7C4A545AEF6)
Monday, 01-06-2020	/api/user/login	1	(6325D400-19A2-425A-B881-540A1E785885)

- logi zmian w bazie relacyjnej

TrainsOnline

Home

Stations

Routes

Tickets

User account

Admin analytics

Admin entity audit log

Route log data

Route report data

Settings

Admin entity audit log

Id	CreatedOn	CreatedBy	TableName	Action
Url: 39523232-c81e-493f-a001-08d7ef53c69f (10 items)				
(B6F42D7E-4C6C-411D-1E34-08D7F4E04BAE)	Sunday, 10-05-2020 02:55:03	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(C4D04E75-468E-48A8-1E33-08D7F4E04BAE)	Sunday, 10-05-2020 02:54:52	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(73BF4ED4-7577-438D-1E32-08D7F4E04BAE)	Sunday, 10-05-2020 02:54:41	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(C5527CA8-F61D-4EE2-8C0B-08D7EDFEA8D3)	Sunday, 03-05-2020 01:22:44	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(56898190-1267-4B0E-8C0A-08D7EDFEA8D3)	Sunday, 03-05-2020 01:22:42	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(32B55FD6-73CB-41ED-8C09-08D7EDFEA8D3)	Sunday, 03-05-2020 01:22:37	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(6FD97844-ECA7-47CD-8C0B-08D7EDFEA8D3)	Sunday, 03-05-2020 01:20:07	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(E2922858-875C-4EFF-8C07-08D7EDFEA8D3)	Sunday, 03-05-2020 01:19:35	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(CC9C3F88-3942-43DB-8C06-08D7EDFEA8D3)	Sunday, 03-05-2020 01:19:21	(39523232-C81E-493F-A001-08D7EF53C69F)	Users	Modified
(521F8CF9-57DD-41FF-8C05-08D7EDFEA8D3)	Sunday, 03-05-2020 01:18:58		Users	Added
Url: 57858492-a326-43a2-6c19-08d7f4e09e18 (2 items)				
(2A4A0A0B-7644-41B8-1E31-08D7F4E04BAE)	Sunday, 10-05-2020 02:50:39	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Stations	Deleted
(330554C8-D3CD-4FAS-1E30-08D7F4E04BAE)	Sunday, 10-05-2020 02:49:45	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Stations	Added
Url: 25dedc5c-90d9-493c-2859-08d7f4e04ba5 (1 item)				
(4F30D9A9-5C07-4369-1E2F-08D7F4E04BAE)	Sunday, 10-05-2020 02:47:27	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Tickets	Added
Url: 4ef8c620-6b1d-4a69-3ad4-08d7f0dd972b (1 item)				
(ED1C29A8-4844-4CC0-424B-08D7F0DD9732)	Tuesday, 05-05-2020 12:18:01	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Tickets	Added
Url: 61675d42-388e-4458-fd5c-08d7f05cf3c5 (1 item)				
(FE65FFC8-3A26-4076-OCF2-08D7F05CF3CD)	Monday, 04-05-2020 08:57:11	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Users	Added
Url: 64b88c51-7f4c-454a-d853-08d7dfa6e719 (3 items)				
(78FE4A71-DF70-4B54-F9AA-08D7F057B57E)	Monday, 04-05-2020 08:26:24	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Users	Modified
(8733657C-3873-4230-F9A9-08D7F057B57E)	Monday, 04-05-2020 08:25:46	(64888C51-7F4C-454A-D853-08D7DFA6E719)	Users	Modified
(637E9CE4-C347-4B04-9EEF-08D7DFA6E72C)	Monday, 13-04-2020 02:33:43		Users	Added

- logi z tras:

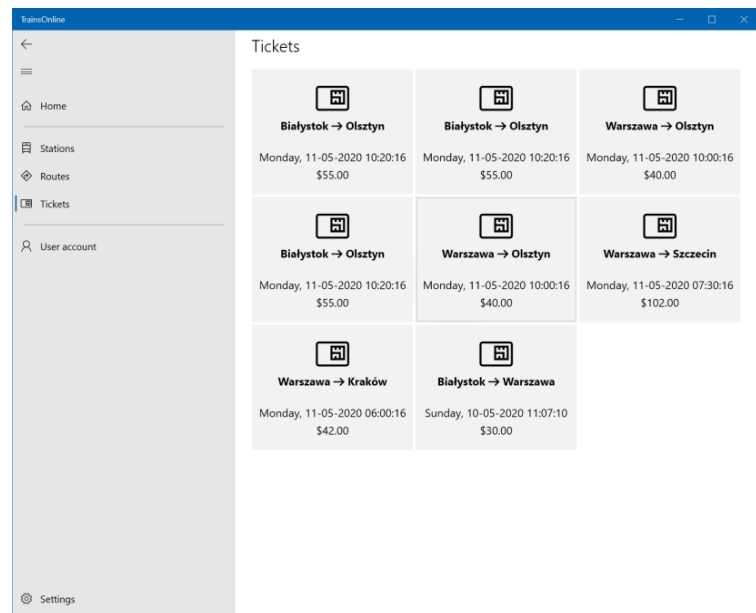
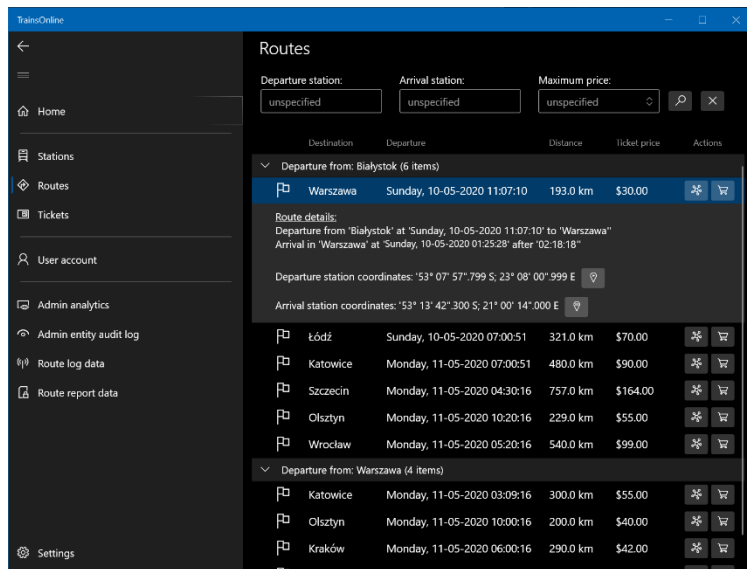
Route log data					
	Timestamp	Latitude	Longitude	Speed	
Route: 7083d789-bcda-4b2f-b4b7-08d7ecf9d55a (458 items)					
	Monday, 11-05-2020 10:58:16	51.1083	17.035	41	
Route report details:					
Id (21F0BF91-DD3E-4D68-8301-44FDC78F925)					
Voltage 3075					
Current 378					
Speed 41					
	Monday, 11-05-2020 10:57:16	51.1127	17.0483	79	
	Monday, 11-05-2020 10:56:16	51.1171	17.0616	71	
	Monday, 11-05-2020 10:55:16	51.1215	17.0749	75	
	Monday, 11-05-2020 10:54:16	51.126	17.0883	0	
	Monday, 11-05-2020 10:53:16	51.1304	17.1016	99	
	Monday, 11-05-2020 10:52:16	51.1348	17.1149	60	
	Monday, 11-05-2020 10:51:16	51.1392	17.1282	43	
	Monday, 11-05-2020 10:50:16	51.1436	17.1415	25	
	Monday, 11-05-2020 10:49:16	51.1481	17.1548	82	
	Monday, 11-05-2020 10:48:16	51.1525	17.1682	83	
	Monday, 11-05-2020 10:47:16	51.1569	17.1815	95	
	Monday, 11-05-2020 10:46:16	51.1613	17.1948	42	
	Monday, 11-05-2020 10:45:16	51.1657	17.2081	57	
	Monday, 11-05-2020 10:44:16	51.1702	17.2214	27	
	Monday, 11-05-2020 10:43:16	51.1746	17.2347	30	
	Monday, 11-05-2020 10:42:16	51.179	17.2481	58	
	Monday, 11-05-2020 10:41:16	51.1834	17.2614	0	

- raporty z tras

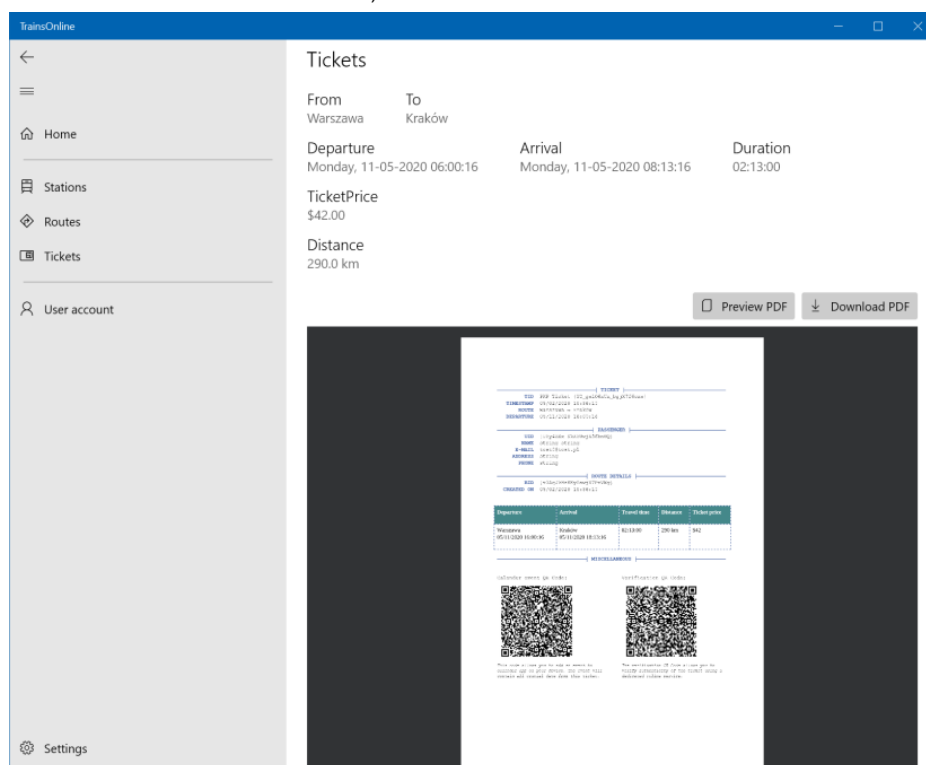
Route report data							
	CreatedOn	LastSavedOn	FromName	ToName	Duration	SpeedMean	
Route: 6d45eac0-16fc-4e56-b4b5-08d7ecf9d55a (2 items)							
	Friday, 05-06-2020 05:25:10	Friday, 05-06-2020 05:25:10	Białystok	Szczecin	13:15:00	46.2249	
Route report details:							
Id: {18B11B76-C4A5-470B-B5A7-236A2EE2259B}							
VoltageMin: 2801							
VoltageMean: 2998.26							
VoltageSd: 115.19							
VoltageMax: 3198							
CurrentMean: 341.984							
CurrentSd: 174.397							
CurrentMax: 648							
PowerMean: 1.02535e+006							
PowerSd: 20068.7							
PowerMax: 2.0723e+006							
SpeedMean: 46.2249							
SpeedSd: 31.0224							
SpeedMax: 99							
Duration: 13:15:00							
NumberOfStops: 290							
	Friday, 05-06-2020 05:13:13	Friday, 05-06-2020 05:13:13	Białystok	Szczecin	00:15:00	59.1875	
Route: 866e09db-dcc8-42e4-b4b6-08d7ecf9d55a (1 item)							
	Friday, 05-06-2020 05:23:02	Friday, 05-06-2020 05:23:02	Białystok	Olsztyn	04:15:00	47.1016	
Route: 7083d789-bcda-4b2f-b4b7-08d7ecf9d55a (2 items)							
	Friday, 05-06-2020 05:21:44	Friday, 05-06-2020 05:21:44	Białystok	Wroclaw	07:37:00	49.9476	
	Friday, 05-06-2020 05:12:36	Friday, 05-06-2020 05:12:36	Białystok	Wroclaw	00:37:00	54.3684	
Route: 35b50a28-bce5-4c6d-b4b4-08d7ecf9d55a (1 item)							
	Friday, 05-06-2020 05:14:34	Friday, 05-06-2020 05:14:34	Warszawa	Szczecin	00:15:00	43.0625	


TrainsOnline

- możliwość kupna biletów;



- pobranie biletu w formacie PDF;





NATIONAL TRAINZZ TICKET

Provided by TrainsOnline

TICKET

TID PKP Ticket {e2214a8f-b926-40ba-4d16-08d7e6b99c7a}

TIMESTAMP 2020-04-22 12:35:16

ROUTE Gdańsk → Poznań

DEPARTURE 2020-04-22 10:40:28

PASSENGER

UID {b2ac5452-e45c-4ed3-4d76-08d7dfa3a155}

NAME string string

E-MAIL test0@test.pl

ADDRESS string

PHONE string

ROUTE DETAILS


RID {29d3fa8b-77bf-40aa-1ccf-08d7e6a9aa76}

CREATED ON 2020-04-22 12:35:16

Departure	Arrival	Travel time	Distance	Ticket price
Gdańsk 2020-04-22 10:40:28	Poznań 2020-04-22 12:40:28	02:00:00	100 km	\$1000


MISCELLANEOUS

Calendar event QR Code:



This code allows you to add an event to calendar app on your device. The event will contain all crucial data from this ticket.

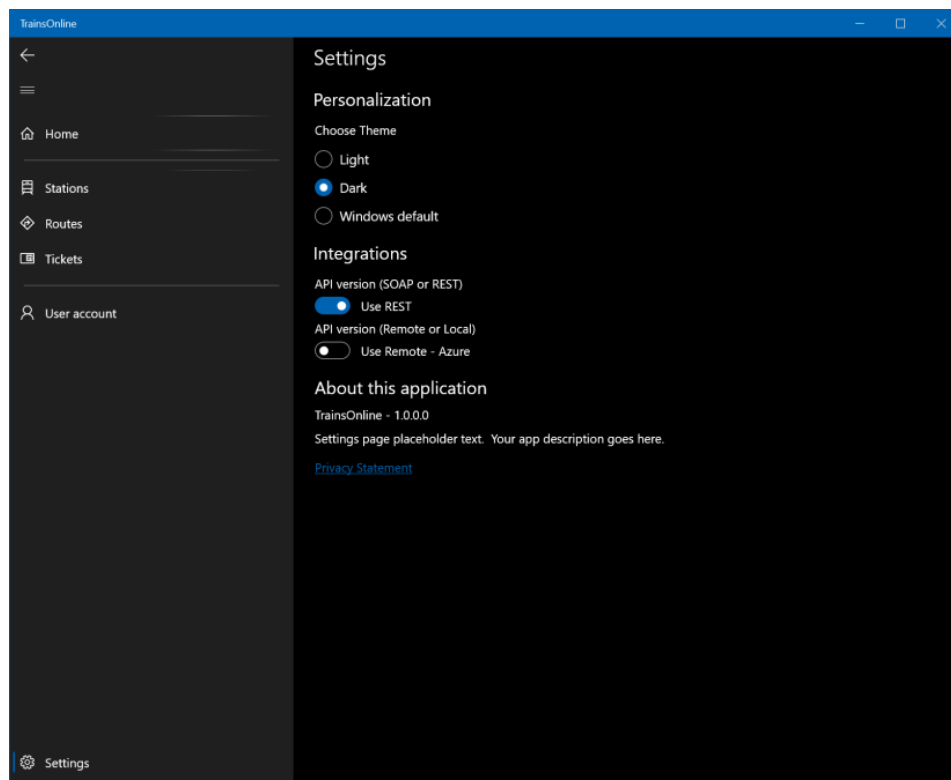
Verification QR Code:



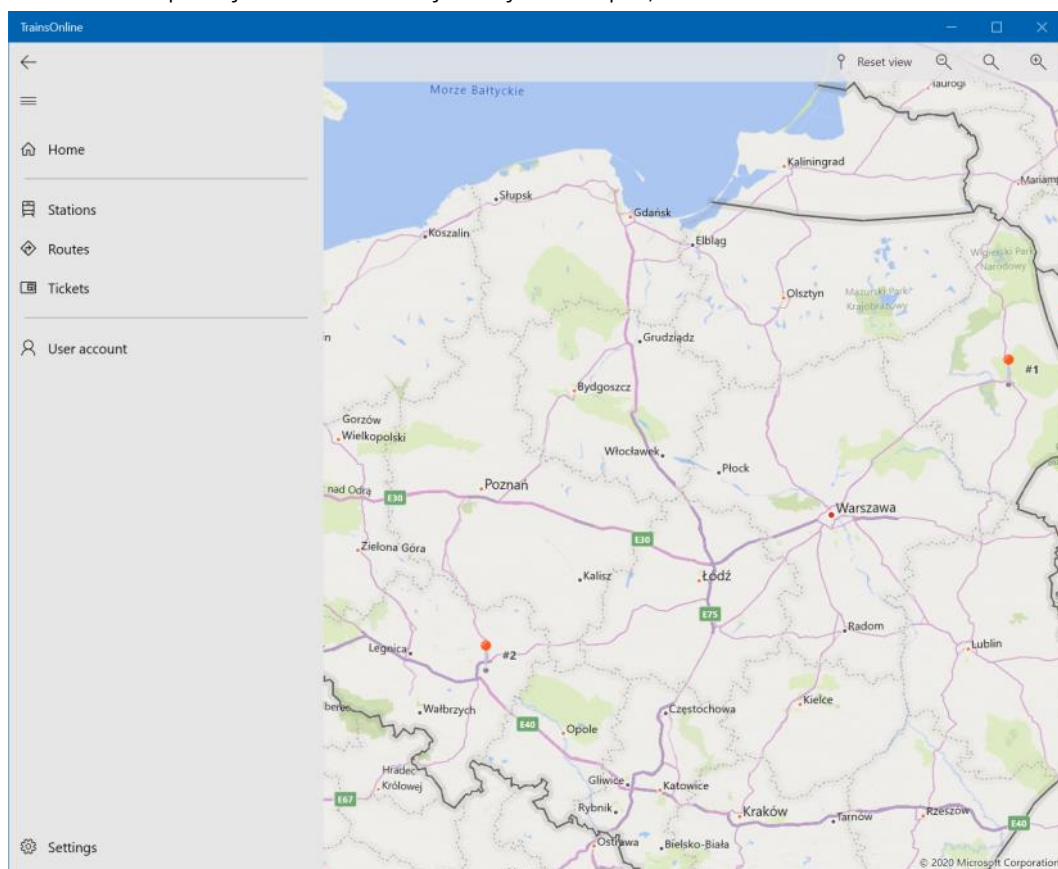
The verification QR Code allows you to verify authenticity of the ticket using a dedicated online service.

Poza spełnieniem podstawnych założeń aplikacja posiada dodatkowe funkcjonalności taki jak:

- możliwość wyboru stylu aplikacji(ciemny lub jasny)



- możliwość podejrzenia lokalizacji stacji na mapie;



- generowanie kodów QR

Calendar event QR Code:



This code allows you to add an event to calendar app on your device. The event will contain all crucial data from this ticket.

Verification QR Code:



The verification QR Code allows you to verify authenticity of the ticket using a dedicated online service.

- wykorzystania algorytmu Jaro-Winkler

←

☰

🏠

📅

🔍

📄

👤

Routes

Departure station:

ziaklystok

Arrival station:

olsztn

Maximum price:

unspecified

🔍

✕

	Destination	Departure	Distance	Ticket price
▼	Departure from: Białystok (1 item)			
📄	Olsztyn	Monday, 11-05-2020 10:20:16	229.0 km	\$55.00
<u>Route details:</u> Departure from 'Białystok' at 'Monday, 11-05-2020 10:20:16' to 'Olsztyn' Arrival in 'Olsztyn' at 'Monday, 11-05-2020 02:36:16' after '04:16:00'				
Departure station coordinates: '53° 07' 57".799 S; 23° 08' 00".999 E <div>📍</div>				
Arrival station coordinates: '53° 46' 36".699 S; 20° 28' 54".000 E <div>📍</div>				

TrainsOnline

- dziennik logów zdarzeń na serwerze

genericapi

All Environments

Last 14 days

Issues (4)

Sort by: Last Seen

Unresolved Issues

is:unresolved

☐ Resolve

☐ Ignore

Merge

...

▶

GRAPH: 24h 14d

EVENTS

USERS

ASSIGNEE

☐ HTTP "POST" "/api/user/reset-password" responded 500 in 2565...
GENERICAPI-4 a few seconds ago Serilog.AspNetCore.RequestLoggingMiddleware

☐ System.Net.Mail.SmtpException TrainsOnline.Infrastructure.Main.Ema...
The SMTP server requires a secure connection or the client was not authenti...
GENERICAPI-3 a few seconds ago - a few seconds old

☐ Failed to determine the https port for redirect.
GENERICAPI-2 Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware

☐ Server START: "PRODUCTION" mode enabled. --ef-migrate=False -...
GENERICAPI-1 2 minutes ago - 2 minutes old

Mon May 04 19:33:38 UTC

/home/aba/DataDisk0/GenericAPI/logs/production_20200504.log

generic logs

LOG

2020-05-04 19:11:40.169 +00:00 [INF] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Entity Framework Core 3.1.3 initialized 'PKPAppDbContext' u

2020-05-04 19:11:41.135 +00:00 [INF] <dotnet 55795:13> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Executed DbCommand (126ms) [Parameters=[@_data_Email_0='?']

SELECT TOP(1) [u].[Id], [u].[Address], [u].[CreatedBy], [u].[CreatedOn], [u].[Email], [u].[IsActive], [u].[LastSavedBy], [u].[LastSavedOn], [u].[Name], [u].[Password

FROM [Users] AS [u]

WHERE [u].[Email] = @_data_Email_0

2020-05-04 19:11:41.574 +00:00 [INF] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Executed action TrainsOnline.Api.Controllers.Authentication

2020-05-04 19:11:41.575 +00:00 [INF] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Executed endpoint 'TrainsOnline.Api.Controllers.Authenticat

2020-05-04 19:11:41.579 +00:00 [ERR] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Unhandled exception in CustomExceptionHandlerMiddleware

System.Net.Mail.SmtpException: The SMTP server requires a secure connection or the client was not authenticated. The server response was: 5.7.0 Authentication Requ

at System.Net.Mail.MailCommand.CheckResponse(SmtpStatusCode statusCode, String response)

at System.Net.Mail.MailCommand.EndSend(IAsyncResult result)

at System.Net.Mail.SendMailAsyncResult.SendMailFromCompleted(IAsyncResult result)

--- End of stack trace from previous location where exception was thrown ---

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw(Exception source)

at System.Net.Mail.SendMailAsyncResult.End(IAsyncResult result)

at System.Net.Mail.SmtpClient.SendMailCallback(IAsyncResult result)

--- End of stack trace from previous location where exception was thrown ---

at TrainsOnline.Infrastructure.Main.Email.EmailService.SendEmail(String email, String subject, String message) in /home/vsts/work/1/s/backend/TrainsOnline.Infras

at TrainsOnline.Application.Handlers.AuthenticationHandlers.Queries.GetResetPasswordToken.GetResetPasswordTokenQuery.Handler.Handle(GetResetPasswordTokenQuery re

at Mediatr.Pipeline.RequestExceptionProcessorBehavior`2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate`1 next)

at Mediatr.Pipeline.RequestExceptionActionProcessorBehavior`2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate`1 next)

at Mediatr.Pipeline.RequestExceptionActionProcessorBehavior`2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate`1 next)

at Mediatr.Pipeline.RequestPostProcessorBehavior`2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate`1 next)

at Mediatr.Pipeline.RequestPreProcessorBehavior`2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate`1 next)

at TrainsOnline.Api.Controllers.AuthenticationController.ResetPasswordStep1(SendResetPasswordRequest request) in /home/vsts/work/1/s/backend/TrainsOnline.Api/Con

at Microsoft.AspNetCore.Mvc.Infrastructure.ActionMethodExecutor.TaskOfIActionResultExecutor.Execute(IActionResultTypeMapper mapper, ObjectMethodExecutor executor

at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeActionMethodAsync>__Awaited|12_0(ControllerActionInvoker invoker, ValueTask`1 actionRe

at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeNextActionFilterAsync>__Awaited|10_0(ControllerActionInvoker invoker, Task lastTask, S

at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.Rethrow(ActionExecutedContextSealed context)

at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.Next(State& next, Scope& scope, Object& state, Boolean& isCompleted)

at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeInnerFilterAsync>__Awaited|13_0(ControllerActionInvoker invoker, Task lastTask, State

at Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeFilterPipelineAsync>__Awaited|19_0(ResourceInvoker invoker, Task lastTask, State next, Scope s

at Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeAsync>__Logged|17_1(ResourceInvoker invoker)

at Microsoft.AspNetCore.Routing.EndpointMiddleware.<Invoke>__AwaitRequestTask|6_0(Endpoint endpoint, Task requestTask, ILogger logger)

at Microsoft.AspNetCore.Authorization.AuthorizationMiddleware.Invoke(HttpContext context)

at Microsoft.AspNetCore.Authentication.AuthenticationMiddleware.Invoke(HttpContext context)

at Microsoft.AspNetCore.Diagnostics.StatusCodePagesMiddleware.Invoke(HttpContext context)

at Microsoft.AspNetCore.ResponseCompression.ResponseCompressionMiddleware.Invoke(HttpContext context)

at TrainsOnline.Api.CustomMiddlewares.Exceptions.CustomExceptionHandlerMiddleware.Invoke(HttpContext context) in /home/vsts/work/1/s/backend/TrainsOnline.Api/Cus

2020-05-04 19:11:41.615 +00:00 [ERR] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) HTTP POST /api/user/reset-password responded 500 in 2565.49

2020-05-04 19:11:41.627 +00:00 [INF] <dotnet 55795:12> (0HLVGBRSEI0J9:00000001-/api/user/reset-password) Request finished in 2611.9695ms 500 application/json

2020-05-04 19:14:52.759 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JA:00000001-/api/index.html) Request starting HTTP/1.1 GET http://genericapi.francecentral.clouda

2020-05-04 19:14:52.826 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JA:00000001-/api/index.html) HTTP GET /api/index.html responded 200 in 63.544454 ms

2020-05-04 19:14:52.826 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JA:00000001-/api/index.html) Request finished in 67.5295ms 200 text/html;charset=utf-8

2020-05-04 19:14:52.930 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JB:00000001-/api/v1/swagger.json) Request starting HTTP/1.1 GET http://genericapi.francecentral.c

2020-05-04 19:14:52.933 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JB:00000001-/api/v1/swagger.json) Request starting HTTP/1.1 GET http://genericapi.francecentral.c

2020-05-04 19:14:52.934 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JB:00000001-/api/v1/swagger.json) Successfully validated the token.

2020-05-04 19:14:52.944 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JC:00000001-/api/favicon-32x32.png) Sending file. Request path: '/favicon-32x32.png'. Physical pa

2020-05-04 19:14:52.945 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JC:00000001-/api/favicon-32x32.png) HTTP GET /api/favicon-32x32.png responded 200 in 10.885127 ms

2020-05-04 19:14:52.945 +00:00 [INF] <dotnet 55795:24> (0HLVGBRSEI0JC:00000001-/api/favicon-32x32.png) Request finished in 11.6189ms 200 image/png

2020-05-04 19:14:53.124 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JB:00000001-/api/v1/swagger.json) HTTP GET /api/v1/swagger.json responded 200 in 193.90577 ms

2020-05-04 19:14:53.125 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JB:00000001-/api/v1/swagger.json) Request finished in 195.0167ms 200 application/json;charset=utf

2020-05-04 19:15:00.574 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JD:00000001-/api/user/reset-password) Request starting HTTP/1.1 POST http://genericapi.francecent

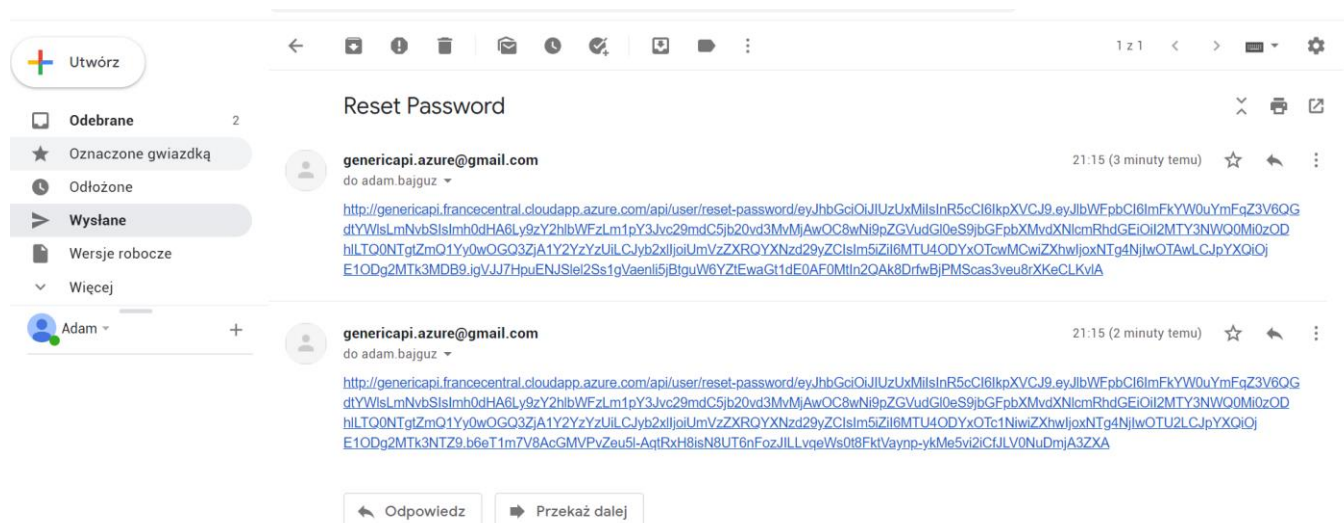
2020-05-04 19:15:00.581 +00:00 [INF] <dotnet 55795:23> (0HLVGBRSEI0JD:00000001-/api/user/reset-password) CORS policy execution failed.

L10857 99% 0 hits

??View Help

Press e/E to move forward/backward through error messages

- wysyłanie emaili



2.3. Profil użytkownika końcowego

Użytkownikiem końcowym aplikacji mogą być przedsiębiorstwa zajmujące się transportem kolejowym i przewozem osób. Aplikacja może posłużyć do zbierania danych z tras, celem ich późniejszej analizy, a także tworzenia dokumentacji końcowej dla przewoźników i podsumowywania działalności firmy w danych okrasach czasowych.

Dane mogą posłużyć także do optymalizacji tras oraz analizy sytuacji specjalnych np. zmiany napięcia w trakcji kolejowej i analizy przyczyny uszkodzeń taboru. W przypadku właściwego wdrożenia na pociągach była by również używana przez maszynistów to kontroli sposobu prowadzenia pociągu, a także mogła by posłużyć jako system ostrzegania np. przed kolizją pociągów.