

Projekt z użyciem SOAP WS  
Temat: System rezerwacji biletów kolejowych



Authors:  
Adam Bajguz  
Michał Kierzkowski

Białystok, 4.05.2020

---

Wykonujący:  
Adam Bajguz  
Michał Kierzkowski  
Grupa PS3

Prowadzący:  
dr hab. inż. Jacek Grekow

---

## Spis treści

1. Wprowadzanie.....	3
1.1. Zakres projektowy .....	3
1.2. Funkcjonalności .....	3
1.3. Definicje, Akronimy, Skróty.....	4
1.4. Użyte technologie .....	6
1.4.1. Po stronie serwera.....	6
1.4.2. Po stronie klienta .....	10
1.5. Swagger .....	11
1.6. ReDoc.....	11
2. Specyfikacja usługi Web Service .....	12
2.1. Standardy sieciowe .....	12
2.2. Adres usługi .....	12
2.3. OpenAPI Specification (fragment) .....	12
2.4. HATEOAS.....	18
2.4.1. Szczegóły stacji.....	18
2.4.2. Szczegóły trasy.....	19
2.5. Dostępne usługi .....	20
2.6. Opis wybranych usług .....	21
2.6.1. Operacja tworzenie biletu.....	21
2.6.2. Operacja pobranie listy biletów danego klienta.....	22
2.6.3. Operacja szczegóły stacji.....	23
2.6.4. Operacja usunięcie stacji .....	24
2.6.5. Operacja zmiana hasła .....	25
3. Instrukcja użytkownika aplikacji klienckiej .....	27

## 1.Wprowadzanie

Celem tego dokumentu jest przedstawienie funkcjonalności, technologii oraz zasad działania(w tym wymiany informacji) w usłudze sieciowej systemu rezerwacji biletów kolejowych oraz interfejsowym systemie klienckim.

### 1.1.Zakres projektowy

Zakresem projektu było stworzenie systemu, który pozwoli na przechowywanie bazy stacji kolejowych i tras, na jakich przemieszczają się pociągi, a także umożliwi zakup biletów na daną trasę oraz pobranie go w formacie PDF. W tym celu należało stworzyć dwie aplikacje jedną, jako WebService dostarczający wszystkie funkcjonalności w API, zaś drugą, jako aplikację kliencką umożliwiającą łatwy dostęp do zasobów serwisowych.

### 1.2.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)

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 loggera Serilog, 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

### 1.3. Definicje, Akronimy, Skróty

**Representational state transfer (REST)** – styl architektoniczny definiujący format przesyłanych danych, utworzony w 2000 roku przez Roya Fieldinga w ramach rozprawy doktorskiej, jako element standaryzacji protokołu HTTP. Używany ze względu na elastyczność, szybkość i prostotę. Nie jest protokołem – jako usługę RESTową można zdefiniować cache’owany, bezstanowy, komunikujący się na zasadzie klient-serwer serwis.

**Web Service (WS)** - podstawowa technologia wykorzystywana w architekturze SOA. Jest to usługa systemu informatycznego, która może być używana przez inne komponenty programowe, również komponenty zewnętrzne projektowanego systemu. Web Service w paradygmacie SOA stanowi najistotniejszy interfejs (ang. API – application programming interface) projektowanego systemu. Web Service jest również wykorzystywany wewnętrznie do komunikacji pomiędzy komponentami systemu.

**Hypermedia as the Engine of Application State (HATEOAS)** - jest składnikiem architektury aplikacji REST, który odróżnia ją od innych architektur aplikacji sieciowych. Dzięki HATEOAS klient wchodzi w interakcję z aplikacją sieciową, której serwery aplikacji dostarczają informacje dynamicznie poprzez hipermedia . Klient REST nie potrzebuje żadnej wiedzy na temat interakcji z aplikacją lub serwerem, poza ogólną znajomością hipermediów.

**Secure Sockets Layer (SSL)**– jest protokołem do zabezpieczania komunikacji odbywającej się w Internecie. Dzięki niemu, wszystkie informacje wysyłane pomiędzy przeglądarką internetową a serwerem są szyfrowane. Aby z niego skorzystać, wymagana jest instalacja certyfikatu SSL.

**HTTP Strict Transport Security (HSTS)** – mechanizm bezpieczeństwa sieci, który chroni strony przed atakami takimi, jak wymuszone zmniejszenie poziomu protokołu oraz przechwytywanie sesji. Dzięki niemu do serwerów można połączyć się tylko za pomocą przeglądarek, korzystających z bezpiecznych połączeń HTTPS, natomiast nigdy nie dopuszcza on połączeń na bazie niezabezpieczonego protokołu HTTP. HSTS jest uznawany za standardowy protokół przez IETF.

**JSON Web Token** - to internetowy standard do tworzenia, tokenów dostępu opartych na JSON. Tokeny są podpisywane przy użyciu prywatnego tajnego klucza lub klucza publicznego / prywatnego. Na przykład serwer może wygenerować token z roszczeniem „zalogowany, jako administrator” i przekazać go klientowi. Klient może następnie użyć tego tokena, aby udowodnić, że jest zalogowany, jako administrator.

**Model-view-viewmodel ( MVVM )** to wzorzec architektoniczny oprogramowania, który ułatwia oddzielenie rozwoju graficznego interfejsu użytkownika ( *widok* ) - czy to poprzez język znaczników lub kod GUI - od rozwoju logiki biznesowej lub back- logika końcowa ( *model* ), dzięki czemu widok nie jest zależny od żadnej konkretnej platformy modelu.

**Universal Windows Platform (UWP)** – interfejs API stworzony przez Microsoft i po raz pierwszy użyty w systemie Windows 10. Jego celem jest ułatwienie pisania aplikacji uniwersalnych, działających zarówno na komputerach, jak i telefonach, konsolach Xbox One oraz okularach HoloLens bez potrzeby ich przepisywania dla każdego rodzaju urządzeń.

**Globally unique identifier (GUID)**, identyfikator globalnie unikatowy – identyfikator obiektów między innymi w systemie Windows lub wszędzie, gdzie potrzebny jest unikatowy identyfikator.

**OpenAPI Specification** – pierwotnie znana jako Swagger Specification, jest specyfikacją plików interfejsu do odczytu maszynowego do opisywania, tworzenia, używania i wizualizacji RESTful web services.

**Postman** – bezpłatnym narzędziem, które możemy wykorzystać do testów API, które zapewnia możliwości automatyzacji testowania, debugowania a także współpracy wielu osób za pomocą kolekcji, środowisk i wsparcia dla ciągłej integracji

**Swagger** – to platforma oprogramowania typu open source wspierana przez duży ekosystem narzędzi, które pomagają programistom projektować, budować, dokumentować i wykorzystywać usługi sieciowe RESTful.

**HTTP verbs** – definiuje zestaw metod żądań w celu wskazania pożądanej akcji do wykonania dla danego zasobu np.

- GET - Metoda żąda reprezentacji określonego zasobu.
- POST - Metoda służąca przesłaniu encji do określonego zasobu, często powodując zmianę stanu.
- PUT - Metoda zastępuje wszystkie aktualne reprezentacje zasobu docelowego.
- DELETE - Metoda usuwa określony zasób.
- PATCH - Metoda służy do stosowania częściowych modyfikacji zasobu.

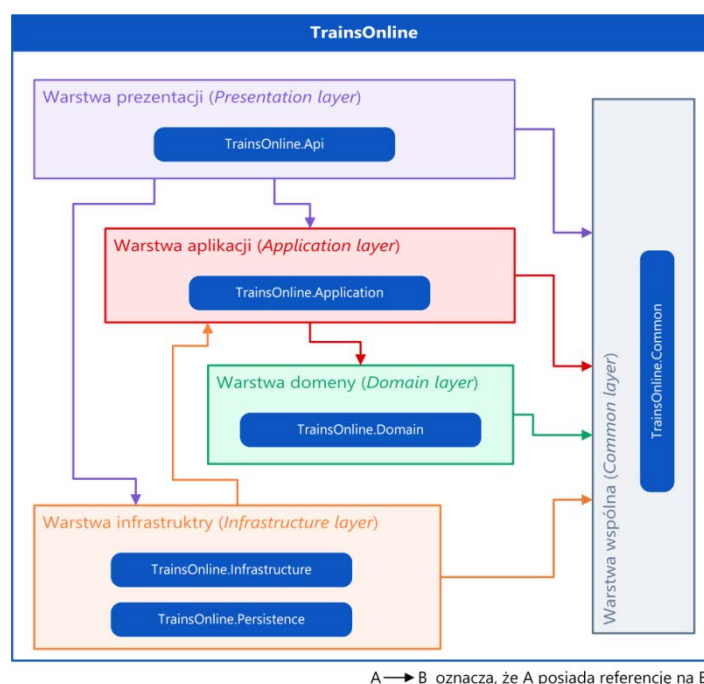
## 1.4. Użyte technologie

### 1.4.1. 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
CommandLineParser	2.8.0
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
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
Sentry.Serilog	2.1.1
Serilog.AspNetCore	3.2.0
Serilog.Enrichers.Memory	1.0.4
Serilog.Enrichers.Process	2.0.1
Serilog.Enrichers.Thread	3.2.0-dev-00747
Serilog.Exceptions	5.4.0
Serilog.Sinks.Async	1.4.0
Serilog.Sinks.Console	3.1.1
Serilog.Sinks.File	4.1.0
Shouldly	3.0.2

Nazwa	Wersja
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
xunit	2.4.1
xunit.runner.visualstudio	2.4.1



Wzorzec DDD wymaga również podziału projektu na następujące warstwy:

- wspólną – obejmuje wszystkie kwestie przekrojowe, czyli klasy i interfejsy wspólne dla różnych warstw;
- prezentacji – zawiera kontrolery, punkty dostępowe do REST API oraz aplikację Swagger;
- 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;
- domeny – zawiera dane, wyliczenia, wyjątki i logikę specyficzną dla domeny;
- 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.

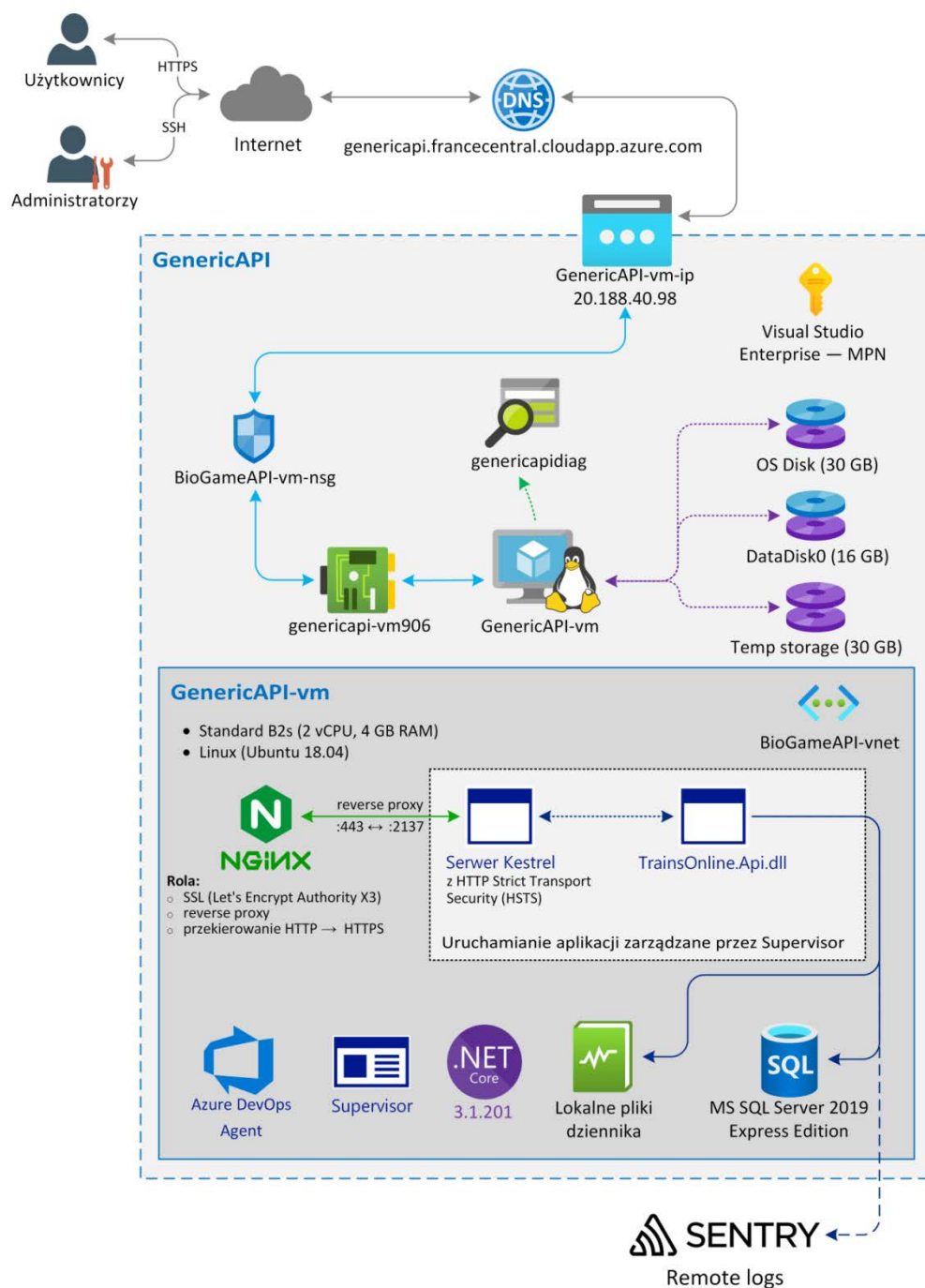
Rozwiązanie to jest wystarczające do zapewnienia czystego kodu oraz architektury w przypadku tego modułu. Ponadto nie wymaga implementowania m.in. koncepcji ograniczonych kontekstów (bounded contexts).

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.

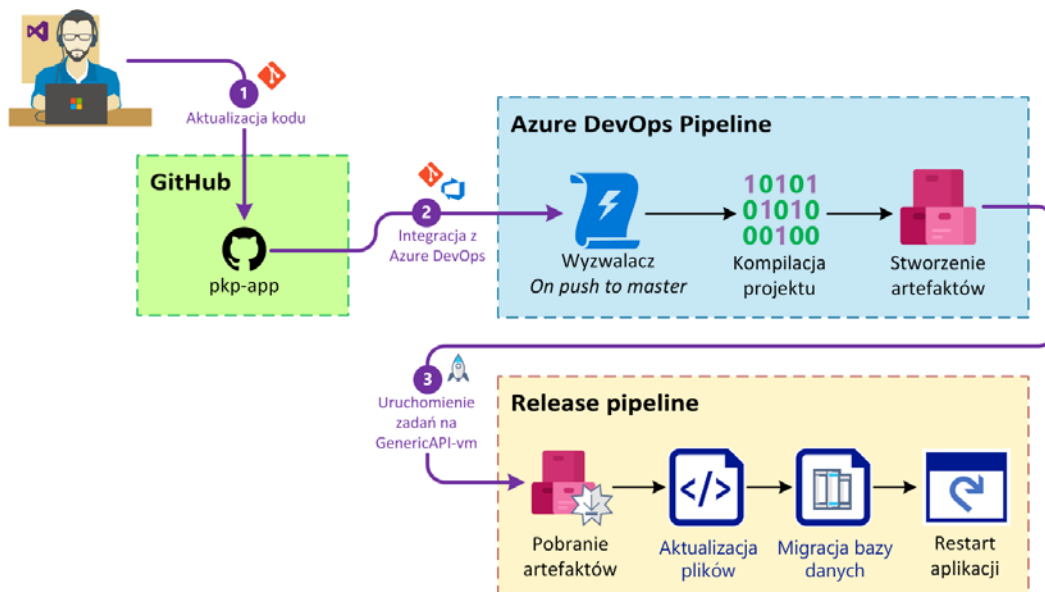
Zdalny dostęp do maszyny wirtualnej odbywa się poprzez protokół Secure Shell (SSH) z wykorzystaniem klucza RSA jako metody uwierzytelniania. Do połączenia się z maszyną wirtualną wykorzystano program Termius z pakietu GitHub Student Developer Pack. Dostarcza on wieloplatformowy terminal ze zintegrowanym 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.





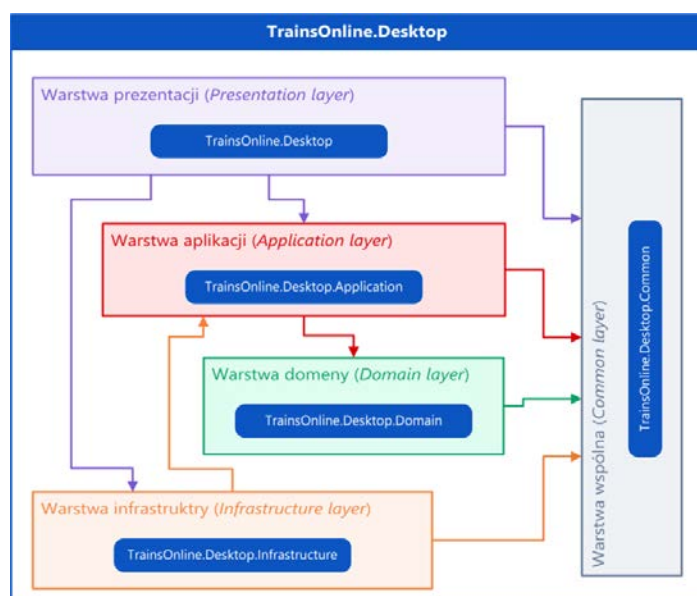
Zastosowanie ciągłej integracji (continuousintegration) oraz ciągłego dostarczania (continuousdelivery) umożliwiło łatwiejszą obsługę środowiska produkcyjnego TrainsOnline. Aktualizacje plików wykonywalnych, a także modyfikacje struktury bazy danych (migracje) odbywają się przy pomocy dwóch potoków:

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.



## 1.4.2. 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. Jediną różnicą jest brak warstwy Persistence.



A → B oznacza, że A posiada referencję na B

## 1.5. Swagger

Swagger  
trainsonline.swagger.io

test0@test.pl [User] [Admin] Logged in 2020-06-04T18:50:55.000Z Logout

Select a definition TrainsOnline v1.0.7434-20679

### TrainsOnline

4.0.7434-20679 QA53

[api/v1/swagger.json](#)

Backend Api for TrainsOnline  
© Adam Bajguz & Michał Kierzkowski 2020

Links:  
OpenAPI specification can be found at [api/v1/swagger.json](#)  
Swagger can be accessed through [api/index.html](#)  
ReDoc can be accessed through [api/redoc/index.html](#)

REST API base url is [api](#)  
SOAP API base url and docs [\[SOAP-API\]](#)  
App health can be checked under [/health](#)

Authorize

#### Authentication

User authentication and password reset

- POST [/api/user/login](#) Login a user
- POST [/api/user/reset-password](#) Send password reset link (Reset password step 1)
- POST [/api/user/reset-password/change](#) Reset user password (Reset password step 2)

#### Route

Create, update, and get route

- POST [/api/route/create](#) Create new route [Admin]
- GET [/api/route/get/{id}](#) Get route details
- PUT [/api/route/update](#) Updated route details [Admin]
- DELETE [/api/route/delete/{id}](#) Delete route [Admin]
- POST [/api/route/get-filtered](#) Get filtered routes
- GET [/api/route/get-all](#) Get all routes

## 1.6.ReDoc

Search...

Authentication

Authentication

Route

- Create new route [Admin]
- Get route details
- Updated route details [Admin]
- Delete route [Admin]
- Get filtered routes
- Get all routes

Station

Ticket

User

Documentation Powered by ReDoc

### Get filtered routes

Gets a filtered list of routes

AUTHORIZATIONS: Bearer

REQUEST BODY SCHEMA: application/json

```
{  "fromPattern": "string Nullable",  "toPattern": "string Nullable",  "maximumTicketPrice": "number <double> Nullable"}
```

#### Responses

200 Success

RESPONSE SCHEMA: text/plain

```
{  "routes": "Array of objects (RouteLookupModel) Nullable"}
```

### Get all routes

Gets a list of all routes

AUTHORIZATIONS: Bearer

#### Responses

200 Success

RESPONSE SCHEMA: text/plain

#### Request samples

Payload

Content type: application/json

```
{  "fromPattern": "string",  "toPattern": "string",  "maximumTicketPrice": 0}
```

#### Response samples

200

Content type: text/plain

No sample

#### Request samples

Payload

Content type: text/plain

```
{  "fromPattern": "string",  "toPattern": "string",  "maximumTicketPrice": 0}
```

#### Response samples

200

Content type: text/plain

No sample

## 2. Specyfikacja usługi Web Service

### 2.1. Standardy sieciowe

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 swaggenera.

### 2.2. Adres usługi

<https://genericapi.francecentral.cloudapp.azure.com/api>

### 2.3. OpenAPISpecification (fragment)

Cała specyfikacja ma rozmiar ponad 3000 linii JSON. Dlatego przedstawiamy fragment:

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "TrainsOnline",
    "description": "Backend Api for TrainsOnline<br>© Adam Bajguz &MichałKierzkowski 2020<br><hr><p><b>Links:</b><br>OpenAPI specification can be found at <a href=\"/api/v1/swagger.json\">/api/v1/swagger.json</a><br>Swagger can be accessed through <a href=\"/api/index.html\">/api/index.html</a><br>ReDoc can be accessed through <a href=\"/api/redoc/index.html\">/api/redoc/index.html</a><br><br>REST API base url is <a href=\"/api\">/api</a><br>SOAP API base url and docs <a href=\"/soap-api\">/soap-api</a><br><br>App health can be checked under <a href=\"/health\">/health</a><br></p>",
    "version": "1.0.7429.34723"
  },
  "paths": {
    "/api/user/login": {
      "post": {
        "tags": [
          "Authentication"
        ],
        "summary": "Login a user",
        "description": "Authenticates a user",
        "requestBody": {
          "content": {
            "application/json-patch+json": {
              "schema": {
                "$ref": "#/components/schemas/LoginRequest"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/LoginRequest"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/LoginRequest"
              }
            },
            "application/*+json": {
              "schema": {
                "$ref": "#/components/schemas/LoginRequest"
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "User authenticated",
            "content": {
              "text/plain": {
                "schema": {
                  "$ref": "#/components/schemas/JwtTokenModel"
                }
              },
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/JwtTokenModel"
                }
              },
              "text/json": {

```

```
        "schema": {
          "$ref": "#/components/schemas/JwtTokenModel"
        }
      },
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        }
      }
    }
  }
},
"/api/route/create": {
  "post": {
    "tags": [
      "Route"
    ],
    "summary": "Create new route [Admin]",
    "description": "Creates a new route",
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/CreateRouteRequest"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateRouteRequest"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateRouteRequest"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/CreateRouteRequest"
          }
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Route created",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/IdResponse"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/IdResponse"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/IdResponse"
          }
        }
      }
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ExceptionResponse"
          }
        }
      }
    }
  }
}
```

```
    }
  }
},
"401": {
  "description": "Unauthorized",
  "content": {
    "text/plain": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    }
  }
}
}
},
"/api/route/get/{id}": {
  "get": {
    "tags": [
      "Route"
    ],
    "summary": "Get route details",
    "description": "Gets route details",
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "string",
          "format": "uuid"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/GetRouteDetailsResponse"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/GetRouteDetailsResponse"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/GetRouteDetailsResponse"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          }
        }
      }
    }
  }
}
},
"/api/ticket/get-all-current-user": {
  "get": {
    "tags": [
      "Ticket"
    ],
    "summary": "Get all current user tickets [User]",
    "description": "Gets a list of all current user tickets",
    "responses": {
```

```
"200": {
  "description": "Success",
  "content": {
    "text/plain": {
      "schema": {
        "$ref": "#/components/schemas/GetUserTicketsListResponse"
      }
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/GetUserTicketsListResponse"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/GetUserTicketsListResponse"
      }
    }
  }
},
"401": {
  "description": "Unauthorized",
  "content": {
    "text/plain": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/ExceptionResponse"
      }
    }
  }
}
}
}
},
"/api/user/create": {
  "post": {
    "tags": [
      "User"
    ],
    "summary": "Create (register) a new user",
    "description": "Creates a new user (requires [Admin] role to create admin account)",
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserRequest"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserRequest"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserRequest"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserRequest"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "User created",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/IdResponse"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/IdResponse"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/IdResponse"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
```

```
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          }
        }
      },
      "401": {
        "description": "Unauthorized",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/ExceptionResponse"
            }
          }
        }
      }
    }
  },
  "components": {
    "schemas": {
      "LoginRequest": {
        "type": "object",
        "properties": {
          "email": {
            "type": "string",
            "nullable": true
          },
          "password": {
            "type": "string",
            "nullable": true
          }
        },
        "additionalProperties": false
      },
      "JwtTokenModel": {
        "type": "object",
        "properties": {
          "token": {
            "type": "string",
            "nullable": true
          },
          "lease": {
            "type": "string",
            "format": "date-span"
          },
          "validTo": {
            "type": "string",
            "format": "date-time"
          }
        },
        "additionalProperties": false
      },
      "HttpStatusCode": {
        "enum": [
          100, 101, 102, 103, 200, 201, 202, 203, 204, 205, 206, 207, 208, 226, 300, 301, 302, 303, 304, 305, 306, 307, 308, 400, 401,
          402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 421, 422, 423, 424, 426, 428, 429, 431, 451, 500, 501,
          502, 503, 504, 505, 506, 507, 508, 510, 511
        ],
        "type": "integer",
        "format": "int32"
      },
      "ExceptionResponse": {
        "type": "object",
        "properties": {
          "statusCode": {
            "$ref": "#/components/schemas/HttpStatusCode"
          },
          "message": {
            "type": "string",
            "nullable": true
          }
        },
        "errors": {
          "type": "object",
          "additionalProperties": false,

```



```
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "IdResponse": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string",
        "format": "uuid"
      }
    },
    "additionalProperties": false
  },
  "UserLookupModel": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string",
        "format": "uuid"
      },
      "email": {
        "type": "string",
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "securitySchemes": {
    "Bearer": {
      "type": "http",
      "description": "JWT Authorization header using the Bearer scheme. Use /api/login endpoint to retrieve a token.",
      "scheme": "bearer"
    }
  },
  "security": [
    {
      "Bearer": [ ]
    }
  ],
  "tags": [
    {
      "name": "Authentication",
      "description": "User authentication and password reset"
    },
    {
      "name": "Route",
      "description": "Create, update, and get route"
    },
    {
      "name": "Station",
      "description": "Create, update, and get station"
    },
    {
      "name": "Ticket",
      "description": "Create, update, and get ticket"
    },
    {
      "name": "User",
      "description": "Create, update, and get user"
    }
  ]
}
```

## 2.4. HATEOAS

### 2.4.1. Szczegóły stacji

GET `{{baseUri}}/api/station/get/id`

Status: 200 OK, Time: 1100 ms, Size: 2.82 KB

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Path Variables

KEY	VALUE	DESCRIPTION
id	73b98be1-c55f-4c9c-f248-08...	(Required)

Body

```
1 {
2   "Id": "73b98be1-c55f-4c9c-f248-08d7ec62224a",
3   "CreatedAt": "2020-04-21T18:59:35.451978",
4   "CreatedBy": "64b88c51-7f4c-454a-d853-08d7dfabe719",
5   "LastSavedOn": "2020-04-30T08:37:39.0129522",
6   "LastSavedBy": "64b88c51-7f4c-454a-d853-08d7dfabe719",
7   "Name": "Białystok",
8   "Latitude": 53.13272222,
9   "Longitude": 23.13361111,
10  "Departures": [
11    {
12      "Id": "4b488030-7184-4483-b4ae-08d7ecf9d55a",
13      "To": {
14        "Id": "90b303ff-9a57-4d2d-f24a-08d7ec62224a",
15        "Name": "Warszawa",
16        "Latitude": 53.22841667,
17        "Longitude": 21.00388889
18      },
19      "DepartureTime": "2020-05-10T09:07:10.745",
20      "Duration": "02:18:18",
21      "Distance": 193,
22      "TicketPrice": 30
23    },
24    {
25      "Id": "a5a93764-bdea-4ef8-b4af-08d7ecf9d55a",
26      "To": {
27        "Id": "0a3dded-d708-4ed3-9960-08d7ec62224a",
28        "Name": "Łódź",
29        "Latitude": 51.75422778,
30        "Longitude": 19.45583333
31      },
32      "DepartureTime": "2020-05-10T17:00:51.619",
33      "Duration": "04:20:00",
34      "Distance": 321,
35      "TicketPrice": 70
36    },
37    {
38      "Id": "95351513-bd08-4590-b4b0-08d7ecf9d55a",
39      "To": {
40        "Id": "b01bbe5c-97de-4c21-9963-08d7ec62224a",
41        "Name": "Katowice",
42        "Latitude": 50.25025,
43        "Longitude": 19.02277778
44      },
45      "DepartureTime": "2020-05-11T15:20:16.905",
46      "Duration": "07:38:00",
47      "Distance": 540,
48      "TicketPrice": 99
49    }
50  ],
51  "links": [
52    {
53      "method": "GET",
54      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/get-all",
55      "relation": "getStationsList",
56      "message": null
57    },
58    {
59      "method": "DELETE",
60      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/delete/73b98be1-c55f-4c9c-f248-08d7ec62224a",
61      "relation": "deleteStation",
62      "message": null
63    },
64    {
65      "method": "PUT",
66      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/update",
67      "relation": "updateStation",
68      "message": null
69    },
70    {
71      "method": "POST",
72      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/create",
73      "relation": "createStation",
74      "message": null
75    },
76    {
77      "method": "GET",
78      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/get/73b98be1-c55f-4c9c-f248-08d7ec62224a",
79      "relation": "self",
80      "message": "This is a GET self link."
81    }
82  ]
83 }
```

GET `{{baseUri}}/api/station/get/id`

Status: 200 OK, Time: 1100 ms, Size: 2.82 KB

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

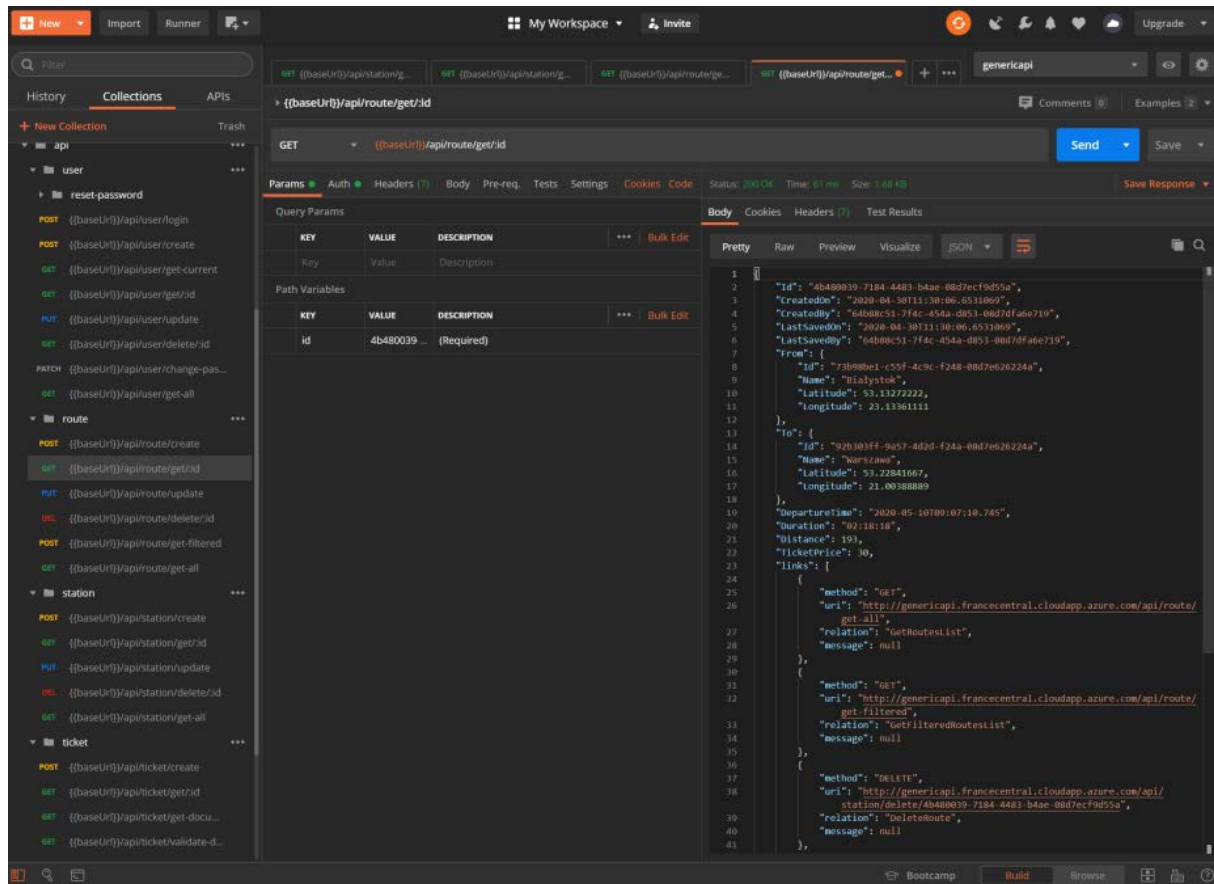
Path Variables

KEY	VALUE	DESCRIPTION
id	73b98be1-c55f-4c9c-f248-08...	(Required)

Body

```
84 {
85   "DepartureTime": "2020-05-11T15:20:16.905",
86   "Duration": "07:38:00",
87   "Distance": 540,
88   "TicketPrice": 99
89 },
90 "links": [
91   {
92     "method": "GET",
93     "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/get-all",
94     "relation": "getStationsList",
95     "message": null
96   },
97   {
98     "method": "DELETE",
99     "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/delete/73b98be1-c55f-4c9c-f248-08d7ec62224a",
100    "relation": "deleteStation",
101    "message": null
102  },
103  {
104    "method": "PUT",
105    "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/update",
106    "relation": "updateStation",
107    "message": null
108  },
109  {
110    "method": "POST",
111    "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/create",
112    "relation": "createStation",
113    "message": null
114  },
115  {
116    "method": "GET",
117    "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/get/73b98be1-c55f-4c9c-f248-08d7ec62224a",
118    "relation": "self",
119    "message": "This is a GET self link."
120  }
121 ]
122 }
```

## 2.4.2. Szczegóły trasy



The screenshot displays the Swagger UI for the TrainsOnline API. The left sidebar shows a collection of endpoints under the 'api' namespace, including endpoints for user management, route management, and station management. The main area shows the details of the selected endpoint: `GET /api/route/get/id`. The endpoint is a GET request that returns a JSON object representing a route. The response body is shown in a 'Pretty' format, displaying a JSON object with the following structure:

```

{
  "id": "4b480039-7184-4483-b4ae-08d7ecf9d55a",
  "createdOn": "2020-04-30T11:38:06.051000Z",
  "createdBy": "4b480039-7184-4483-b4ae-08d7ecf9d55a",
  "lastSavedOn": "2020-04-30T11:38:06.051000Z",
  "lastSavedBy": "4b480039-7184-4483-b4ae-08d7ecf9d55a",
  "from": {
    "id": "73b098be1-c55f-4c9c-f248-08d7e626224a",
    "name": "Bialystok",
    "latitude": 53.13272222,
    "longitude": 23.13361111
  },
  "to": {
    "id": "92b303ff-9a57-4d2d-f24a-08d7e626224a",
    "name": "Warszawa",
    "latitude": 52.22841667,
    "longitude": 21.00388889
  },
  "departureTime": "2020-05-10T09:07:10.745Z",
  "duration": "02:18:18",
  "distance": 103,
  "ticketPrice": 30,
  "links": [
    {
      "method": "GET",
      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/route/get-all",
      "relation": "GetRoutesList",
      "message": null
    },
    {
      "method": "GET",
      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/route/get-filtered",
      "relation": "GetFilteredRoutesList",
      "message": null
    },
    {
      "method": "DELETE",
      "uri": "http://genericapi.francecentral.cloudapp.azure.com/api/station/delete/4b480039-7184-4483-b4ae-08d7ecf9d55a",
      "relation": "DeleteRoute",
      "message": null
    }
  ]
}

```

### 2.5. Dostępne usługi

Usługi zostały podzielone na 5 grup użytkowych, reprezentowanych przez odpowiednie serwisy.

Grupa użytkowa	Dostępne usługi
Autoryzacja	<ul style="list-style-type: none"><li>• Logowanie</li><li>• Reset Hasła (krok 1 i 2)</li></ul>
Trasy kolejowe	<ul style="list-style-type: none"><li>• Utwórz trasę (A)</li><li>• Szczegóły trasy (U) (A)</li><li>• Modyfikacja trasy (A)</li><li>• Usuwanie trasy (A)</li><li>• Wypisanie wszystkich tras</li></ul>
Stacje kolejowe	<ul style="list-style-type: none"><li>• Utwórz stacji (A)</li><li>• Szczegóły stacji (U) (A)</li><li>• Modyfikacja stacji (A)</li><li>• Usuwanie stacji (A)</li><li>• Wypisanie wszystkich stacji</li></ul>
Bilety	<ul style="list-style-type: none"><li>• Tworzenie biletu (U) (A)</li><li>• Szczegóły biletu (U) (A)</li><li>• Tworzenie PDF biletu (U) (A)</li><li>• Modyfikacja biletu (A)</li><li>• Usuwanie biletu (A)</li><li>• Wypisywanie wszystkich biletów danego użytkownika (U) (A)</li><li>• Wypisywanie wszystkich biletów utworzonych w serwisie (A)</li></ul>
Użytkownik	<ul style="list-style-type: none"><li>• Tworzenie użytkownika</li><li>• Szczegóły użytkownika (U)</li><li>• Modyfikacja użytkownika (U)</li><li>• Usuwanie użytkownika (U)</li><li>• Zmiana hasła (U)</li><li>• Wypisani wszystkich użytkowników (A)</li></ul>

U - Usługa dostępna dla użytkownika serwisu

A -Usługa dostępna dla administratora serwisu

## 2.6. Opis wybranych usług

### 2.6.1. Operacja tworzenie biletu

Operacja tworzenie biletu służy do dodawania wpisu o zakupie biletu do bazy danych, wpis zawiera takie dane jak identyfikator klienta, identyfikator trasy oraz datę zakupu. Aby korzystać z operacji użytkownik musi być zalogowany

Opis szczegółowy:

Dane wejściowe:

- **UserId** – identyfikator użytkownika –32-znakowy ciąg heksadecymalny np. 3fa85f64-5717-4562-b3fc-2c963f66afa6, wygenerowany przy pomocy GUID
- **Routeld** – identyfikator trasy kolejowej –32-znakowy ciąg heksadecymalny np. 3fa85f64-5717-4562-b3fc-2c963f66afa6, wygenerowany przy pomocy GUID

Dane wyjściowe:

- **TicketId**– identyfikator nowo powstałego biletu– 32-znakowy ciąg heksadecymalny np. 3fa85f64-5717-4562-b3fc-2c963f66afa6, wygenerowany przy pomocy GUID
- **Komunikat o błędzie** – w razie błędnych danych lub braku uprawnień do korzystania z operacji pojawi się komunikat o błędzie

Komunikat wysyłany do usługi i zwracany z usługi:

The screenshot shows a REST client interface with the following details:

- Request Body:** A JSON object with "userId" and "routeId" fields, both containing 32-character hexadecimal strings.
- Request URL:** `https://genericapi.francecentral.cloudapp.azure.com/api/ticket/create`
- Server response:** A 200 status code with a JSON response body containing a "TicketId" field with a 32-character hexadecimal value.
- Response headers:** Includes "Connection: keep-alive", "Content-Encoding: br", "Content-Type: application/json; charset=utf-8", "Date: Sun, 16 May 2020 11:47:17 GMT", "Server: glns/1.16.0 (Ubuntu)", "Transfer-Encoding: chunked", and "Vary: Accept-Encoding".
- Request duration:** 382 ms.

## 2.6.2. Operacja pobranie listy biletów danego klienta

Operacja pobrania listy biletów danego klienta służy do wylistowania wszystkich zakupionych biletów przez klienta, wpis w liście zawiera takie dane jak identyfikator biletu, identyfikator trasy, dane stacji startowej i docelowej oraz datę zakupu biletu. Aby korzystać z operacji użytkownik musi być zalogowany.

Opis szczegółowy:

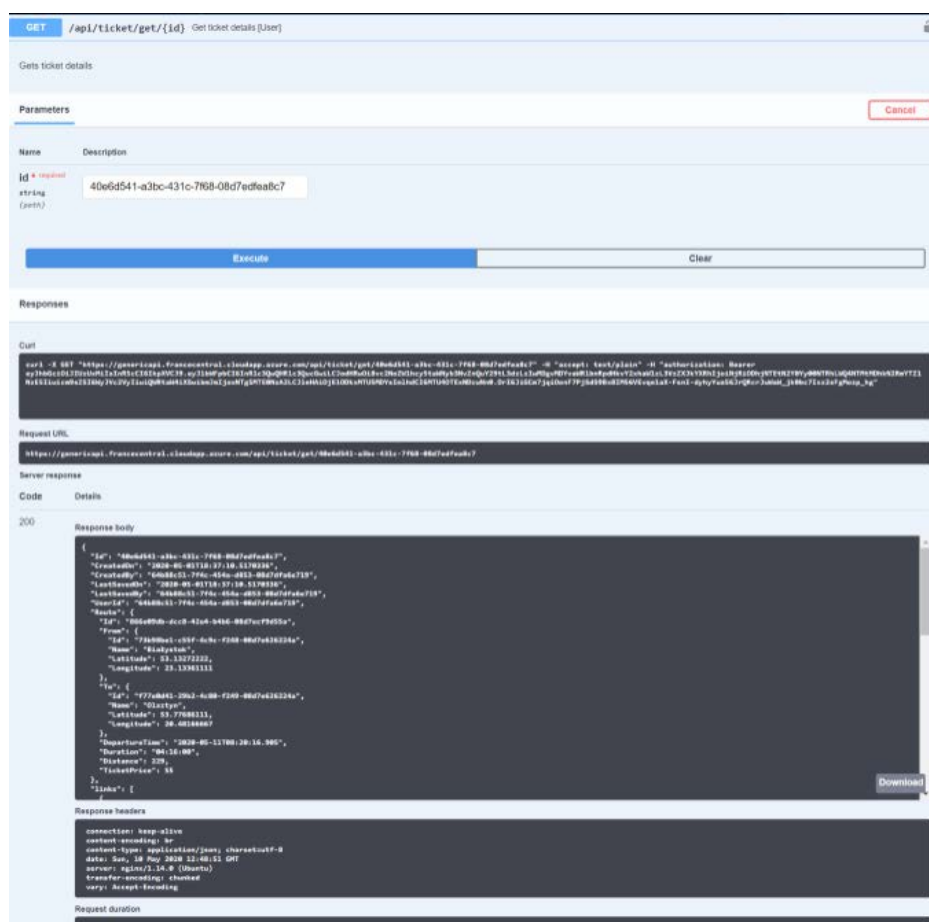
Dane wejściowe:

- **Userld** – identyfikator użytkownika – 32-znakowy ciąg heksadecymalny np. 3fa85f64-5717-4562-b3fc-2c963f66afa6, wygenerowany przy pomocy GUID

Dane wyjściowe:

- **Tickets**– lista biletów wraz z ich szczegółami  
lub
- **Komunikat o błędzie** – w razie błędnych danych lub braku uprawnień do korzystania z operacji pojawi się komunikat o błędzie

Komunikat wysyłany do usługi i zwracany z usługi:



## 2.6.3. Operacja szczegóły stacji

Operacja szczegóły stacji służy do wypisania wszystkich danych o wybranej stacji zawartych w bazie, szczegóły te obejmują dane takie jak nazwa, czy też położenie geograficzne stacji. Aby korzystać z operacji użytkownik musi być zalogowany.

Opis szczegółowy:

Dane wejściowe:

- **stationId** – identyfikator stacji– 32-znakowy ciąg heksadecymalny np. 3fa85f64-5717-4562-b3fc-2c963f66afa6, wygenerowany przy pomocy GUID

Dane wyjściowe:

- **Station**– wszystkie szczegóły stacji zawarte w bazie łącznie ze stacjami do których wychodzą trasy z danej stacji  
lub
- **Komunikat o błędzie** – w razie błędnych danych lub braku uprawnień do korzystania z operacji pojawi się komunikat o błędzie

Komunikat wysyłany do usługi i zwracany z usługi:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /api/station/get/{id} (Get station details)
- Parameters:** A table with one entry: 

Name	Description
id * required string (path)	73b98be1-c55f-4c9c-f248-88d7e626224a
- Execute:** A blue button to run the request.
- Responses:** A section showing the raw response data.
- Code:** 200
- Response body:** A JSON object containing station details and a list of departures.

```
{
  "Id": "73b98be1-c55f-4c9c-f248-88d7e626224a",
  "CreatedOn": "2020-04-11T18:59:35.451Z",
  "CreatedBy": "64b8c51-7f4c-454a-d853-88d7e6fde719",
  "LastSavedOn": "2020-04-30T08:37:39.8125Z",
  "LastSavedBy": "64b8c51-7f4c-454a-d853-88d7e6fde719",
  "Name": "Marszanna",
  "Latitude": 53.13272222,
  "Longitude": 23.13961111,
  "Departures": [
    {
      "Id": "4b488639-7184-6423-b4ae-88d7ecf9d55a",
      "To": {
        "Id": "92b382ff-9a57-4d2d-f24a-88d7e626224a",
        "Name": "Marszanna",
        "Latitude": 53.2201667,
        "Longitude": 21.0038889
      },
      "DepartureTime": "2020-05-18T09:07:10.745",
      "Duration": "02:18:18",
      "Distance": 153,
      "TicketPrice": 38
    },
    {
      "Id": "64a33764-bdea-4c78-bdaf-88d7ecf9d55a",
      "To": {
        "Id": "943d4ced-4768-4ed3-9968-88d7ec9d413",
        "Name": "Gdansk"
      }
    }
  ]
}
```







Komunikat błędu:

LOGS

DETAILS

400

Error: Bad Request

Response body

```
{
  "statusCode": 400,
  "message": "Validation failed: \n -- OldPassword: Old password is incorrect",
  "errors": [
    {
      "field": "OldPassword",
      "message": "Old password is incorrect",
      "errorCode": "AsyncPredicateValidator"
    }
  ]
}
```

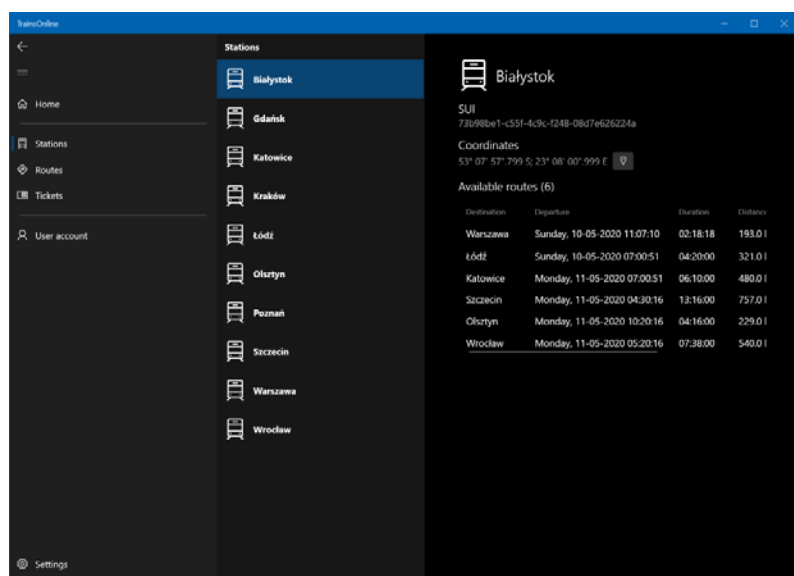
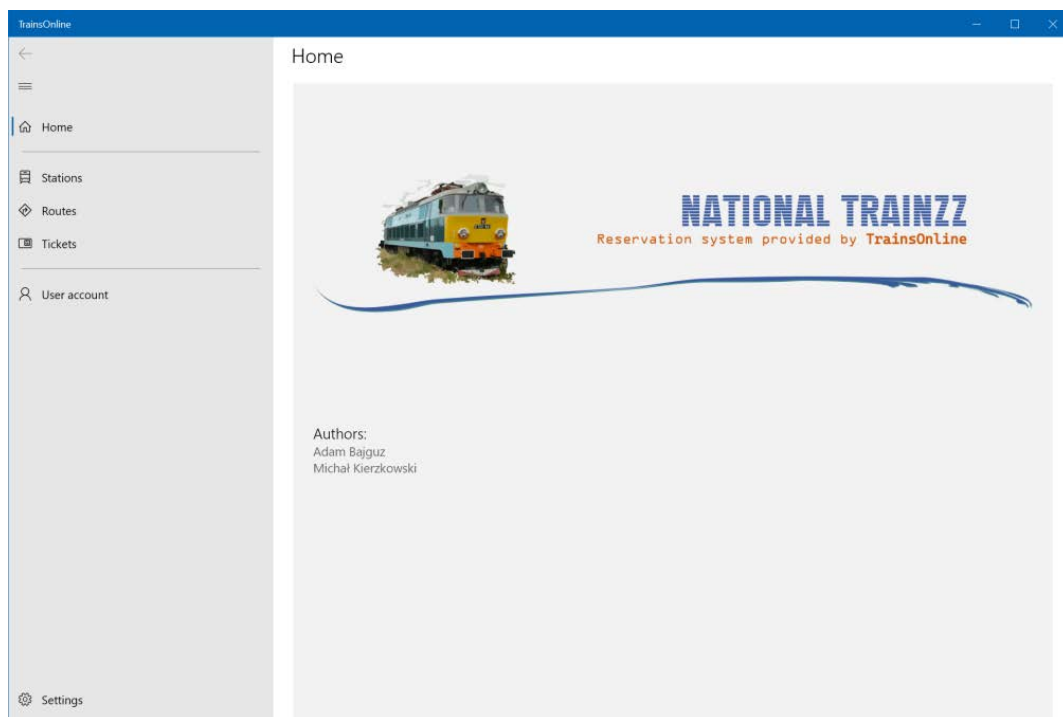
Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sun, 10 May 2020 12:25:04 GMT
server: nginx/1.14.0 (Ubuntu)
transfer-encoding: chunked
```

## 3. Instrukcja użytkownika aplikacji klienckiej

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.



# TrainsOnline

Posiada ona obsługę funkcjonalności API takich jak:

- logowanie i rejestracja;

The screenshot shows the 'Tickets' section of the TrainsOnline application. It features a sidebar with navigation links: Home, Stations, Routes, Tickets, and User account. The main content area is titled 'Tickets' and contains a message: 'Authorization is required to access this page'. Below this message are two tabs: 'Login' and 'Register'. The 'Login' tab is active, showing input fields for 'e-mail' and 'password', and a 'Login' button. The 'Register' tab is also visible, showing input fields for 'e-mail', 'password', 'name', 'surname', 'phone number', and 'address', along with a 'Register' button.

This screenshot shows the 'Register' tab of the TrainsOnline application. It includes the same sidebar as the previous screenshot. The main content area is titled 'Tickets' and contains a message: 'Authorization is required to access this page'. Below this message are two tabs: 'Login' and 'Register'. The 'Register' tab is active, showing input fields for 'e-mail', 'password', 'name', 'surname', 'phone number', and 'address', along with a 'Register' button.

- możliwość kupna biletów;

The screenshot shows the 'Routes' section of the TrainsOnline application. It features a sidebar with navigation links: Home, Stations, Routes, Tickets, and User account. The main content area is titled 'Routes' and contains a search bar with fields for 'Departure station', 'Arrival station', and 'Maximum price'. Below the search bar is a table of routes. The table has columns for 'Destination', 'Departure', 'Distance', 'Ticket price', and 'Actions'. The table is divided into two sections: 'Departure from: Białystok (6 items)' and 'Departure from: Warszawa (4 items)'. The routes are listed with their respective departure times, distances, and ticket prices.

Destination	Departure	Distance	Ticket price	Actions
Departure from: Białystok (6 items)				
Warszawa	Sunday, 10-05-2020 11:07:10	193.0 km	\$30.00	
Route details: Departure from 'Białystok' at 'Sunday, 10-05-2020 11:07:10' to 'Warszawa' Arrival in 'Warszawa' at 'Sunday, 10-05-2020 01:23:28' after '02:18:18'				
Departure station coordinates: '53° 07' 57" 799 S, 23° 08' 00" 999 E				
Arrival station coordinates: '53° 13' 42" 300 S, 21° 00' 14" 000 E				
Lódź	Sunday, 10-05-2020 07:00:51	321.0 km	\$70.00	
Katowice	Monday, 11-05-2020 07:00:51	480.0 km	\$90.00	
Szczecin	Monday, 11-05-2020 04:30:16	757.0 km	\$164.00	
Olsztyn	Monday, 11-05-2020 10:20:16	229.0 km	\$55.00	
Wrocław	Monday, 11-05-2020 05:20:16	540.0 km	\$99.00	
Departure from: Warszawa (4 items)				
Katowice	Monday, 11-05-2020 03:09:16	300.0 km	\$55.00	
Olsztyn	Monday, 11-05-2020 10:00:16	200.0 km	\$40.00	
Kraków	Monday, 11-05-2020 06:00:16	290.0 km	\$42.00	
Szczecin	Monday, 11-05-2020 07:30:16	560.0 km	\$102.00	

The screenshot shows the 'Tickets' section of the TrainsOnline application. It features a sidebar with navigation links: Home, Stations, Routes, Tickets, and User account. The main content area is titled 'Tickets' and contains a grid of ticket offers. Each offer is represented by a card with a train icon, the route name, the departure time, and the ticket price.

Route	Departure	Ticket price
Białystok → Olsztyn	Monday, 11-05-2020 10:20:16	\$55.00
Białystok → Olsztyn	Monday, 11-05-2020 10:20:16	\$55.00
Warszawa → Olsztyn	Monday, 11-05-2020 10:00:16	\$40.00
Białystok → Olsztyn	Monday, 11-05-2020 10:20:16	\$55.00
Warszawa → Olsztyn	Monday, 11-05-2020 10:00:16	\$40.00
Warszawa → Szczecin	Monday, 11-05-2020 07:30:16	\$102.00
Warszawa → Kraków	Monday, 11-05-2020 06:00:16	\$42.00
Białystok → Warszawa	Sunday, 10-05-2020 11:07:10	\$30.00

- pobranie biletu w formacie PDF;

TrainsOnline

←

☰

Home

Stations

Routes

Tickets

User account

Settings

Tickets

From

Warszawa

To

Kraków

Departure

Monday, 11-05-2020 06:00:16

Arrival

Monday, 11-05-2020 08:13:16

Duration

02:13:00

TicketPrice

\$42.00

Distance

290.0 km

Preview PDF

Download PDF

TICKET

TID

122\_gz0d6u2\_2g\_p728uax

TIME/STAMP

05/11/2020 21:08:11

ROUTE

WARSAWA - KRAKOW

DEPARTURE

05/11/2020 06:00:16

PASSENGER

NAME

Janina Janina

NAME

Janina Janina

E-MAIL

janina.janina@gmail.com

ADDRESS

Warszawa

PHONE

123456789

NOTE DETAILS

NOTE

123456789

CREATED ON

05/11/2020 21:08:11

Departure	Arrival	Travel time	Distance	Ticket price
Warszawa 05/11/2020 06:00:16	Krakow 05/11/2020 08:13:16	02:13:00	290 km	\$42

QR CODE

QR CODE

QR CODE



## 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:

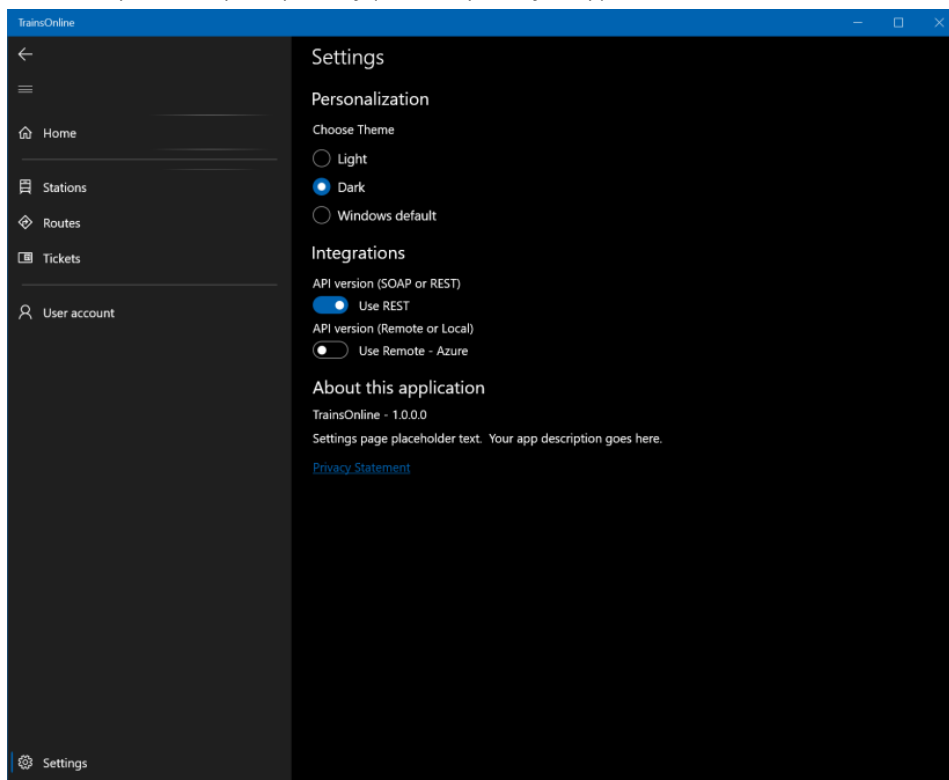


The verification QR Code allows you to verify authenticity of the ticket using a dedicated online service.

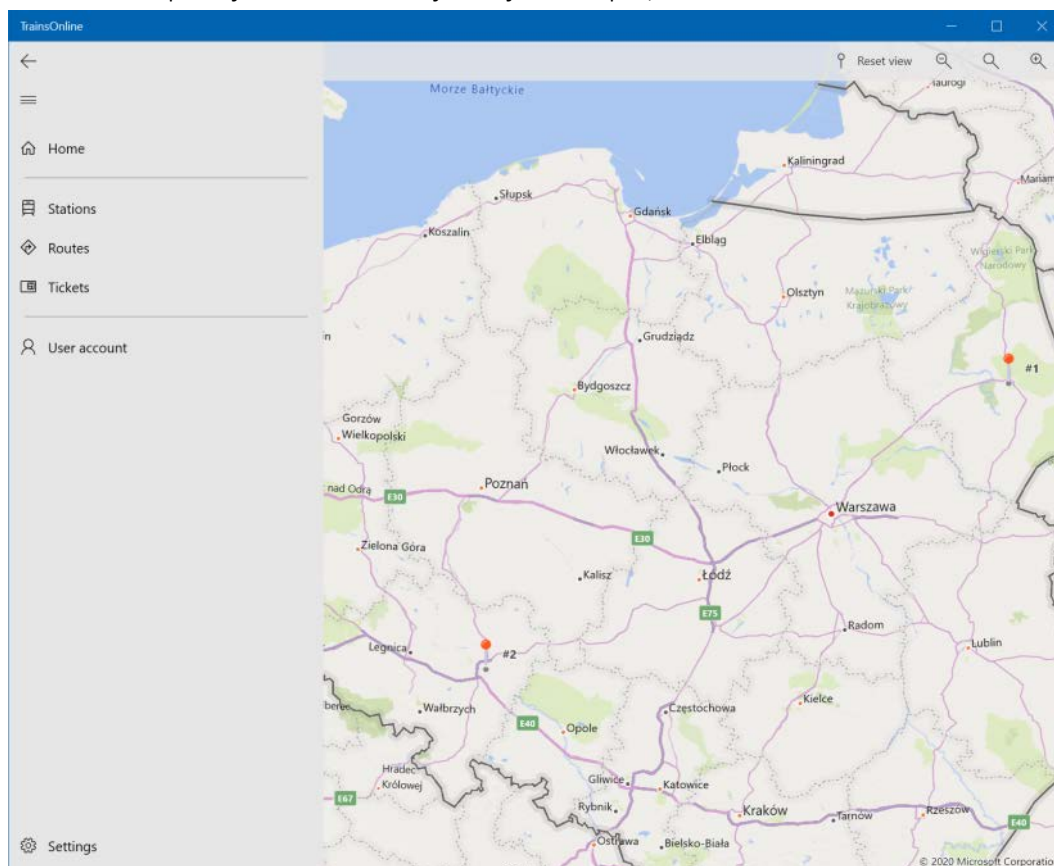
i wiele innych.

Poza spełnieniem podstawnych założeń aplikacja posiada dodatkowe funkcjonalności takie 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

Route details:

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

📍

Arrival station coordinates: '53° 46' 36".699 S; 20° 28' 54".000 E

📍



- dziennik logów

genericapi

All Environments

Last 14 days

Issues (4)

Sort by: Last Seen

Unresolved Issues

is:unresolved

	GRAPH:	24h	14d	EVENTS	USERS	ASSIGNEE
<div> <div>HTTP "POST" "/api/user/reset-password" responded 500 in 2565...</div> <div> <div>GENERICAPI-4</div> <div>a few seconds ago</div> </div> <div> <div>Serilog.AspNetCore.RequestLoggingMiddleware</div> </div> </div>				1	1	
<div> <div>System.Net.Mail.SmtpException</div> <div>TrainsOnline.Infrastructure.Main.Ema...</div> <div>The SMTP server requires a secure connection or the client was not authenti...</div> <div> <div>GENERICAPI-3</div> <div>a few seconds ago</div> </div> </div>				1	1	
<div> <div>Failed to determine the https port for redirect.</div> <div> <div>GENERICAPI-2</div> <div></div> </div> <div> <div>Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware</div> </div> </div>				1	1	
<div> <div>Server START: "PRODUCTION" mode enabled. --ef-migrate=False -...</div> <div> <div>GENERICAPI-1</div> <div>2 minutes ago</div> </div> </div>				2	1	

```

Mon May 04 19:33:38 UTC
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 'PKAppDbContext' 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].[IsAdmin], [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.Authentication
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.Authentication.Queries.GetResetPasswordToken.GetResetPasswordTokenQuery.Handler.Handle(GetResetPasswordTokenQuery re
at Mediatr.Pipeline.RequestExceptionProcessorBehavior'2.Handle(TRequest request, CancellationToken cancellationToken, RequestHandlerDelegate'1 next)
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.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.TaskOfActionResultExecutor.Execute(IActionResultTypeMapper mapper, ObjectMethodExecutor executor
at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeActionMethodAsync>g__Awaited|12_0(ControllerActionInvoker invoker, ValueTask'1 actionRe
at Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker.<InvokeNextActionFilterAsync>g__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>g__Awaited|13_0(ControllerActionInvoker invoker, Task lastTask, State
at Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeFilterPipelineAsync>g__Awaited|19_0(ResourceInvoker invoker, Task lastTask, State next, Scope s
at Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeAsync>g__Logged|17_1(ResourceInvoker invoker)
at Microsoft.AspNetCore.Routing.EndpointMiddleware.<Invoke>g__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> (0HLVGBRSEI0JC:00000001-/api/favicon-32x32.png) Request starting HTTP/1.1 GET http://genericapi.francecentral
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 193.99577 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.francecont
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

