

Analiza skali wykluczenia komunikacyjnego na obszarze Polski wraz z rekomendacjami zmian legislacyjnych w kontekście publicznego transportu zbiorowego T-INCLUDED

Zadanie 20

**Aktualizacja mapy wykluczenia, opracowanie
deweloperskiej dokumentacji wdrożeniowej**

**Utrzymanie aktualności systemu informacji wraz z opracowaniem
zasad dostępu**

30 kwiecień 2025 r.



Rzeczpospolita
Polska

NCBR 
Narodowe Centrum Badań i Rozwoju



POLITECHNIKA POZNAŃSKA



POLITECHNIKA
GDAŃSKA



Politechnika
Śląska

Politechnika
Warszawska

INFORMACJE O WYKONAWCY

Politechnika Poznańska

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

NIP: 7770003699

REGON: 000001608

e-mail: biuro.rektora@put.poznan.pl

strona: <http://www.put.poznan.pl>

ZESPÓŁ WYKONAWCÓW PRAC

Wydział Inżynierii Lądowej i Transportu Politechniki Poznańskiej

ul. Piotrowo 3, 61-138 Poznań

strona: <https://wilit.put.poznan.pl>

Wykonawcy:

- Andrzej Markiewicz – IT
- Michał Wojtał – IT
- Paweł Zmuda-Trzebiatowski – IT, ZST (*kierownik projektu na PP*)

Spis treści

1. Wprowadzenie	4
2. Testowanie funkcjonalności aktualizacji danych wejściowych w warunkach rzeczywistych	5
3. Opracowanie zasad dostępu do systemu informacji o wykluczeniu przez jednostki trzecie	10
4. Wykonanie specyfikacji standardów obowiązujących w podsystemie informacji o funkcjonowaniu PTZ w języku angielskim	12
5. Osiągnięcie kamieni milowych zadania	13
5.1. Aktualizacja mapy zagrożenia WK	13
5.2. Procedura dostępu do systemu przez jednostki trzecie	13
5.3. Tłumaczenie na język angielski specyfikacji rozszerzonych standardów danych dot. funkcjonowania PTZ w Polsce	13
6. Podsumowanie	14
Załączniki	17
Z1a. Technology for preparing API and its documentation	18
Z1b. WebAPI Documentation Contents in SWAGGER	20
1. General API Information (info)	20
2. List of Resources (Endpoints) (paths)	20
3. Parameters	20
4. Data Schemas (components/schemas)	21
5. Responses	21
Z1c. Examples of WebAPI documentation fragments prepared for the project	22
1. Endpoint: GET /api/Agency/{id}	22
2. Endpoint: DELETE /api/Agency/{id}	23
3. Endpoint: GET /api/Agency	23
4. Endpoint: POST /api/Agency	23
5. Endpoint: PUT /api/Agency	24
Z1d. List of Modules Containing Endpoints Described in the WebAPI Documentation	26
Z1e. Complete standard definition	28

1. Wprowadzenie

Zadanie 20 stanowi przedostatni, przedwdrożeniowy etap prac nad systemem informacji o zagrożeniu z transportu zbiorowego. Zadanie realizowano w powiązaniu z Zad. 21, w szczególności w zakresie szkolenia z systemu. W trakcie przedmiotowego szkolenia nie zgłoszono dalszych uwag dotyczących systemu. Należy jednak zauważyć, że informatyczne działania w zakresie utrzymania systemu były niezbędne do prawidłowego przebiegu szkolenia.

Ponadto prowadzono działania mające na celu osiągnięcie trzech kamieni milowych, tj.:

- aktualizacji mapy zagrożenia WK.
- opracowania procedury dostępu do systemu przez jednostki trzecie, takie jak administracja czy podmioty zainteresowane danymi o PTZ,
- przetłumaczenie na język angielski specyfikacji rozszerzonych standardów danych dotyczących funkcjonowania PTZ w Polsce, które są stosowane w opracowanym systemie.

W kolejnych rozdziałach scharakteryzowano przebieg prac oraz zaprezentowano uzyskane efekty tych prac.

2. Testowanie funkcjonalności aktualizacji danych wejściowych w warunkach rzeczywistych

Celem działania D1 było przetestowanie funkcjonalności systemu w zakresie obsługi możliwości aktualizacji danych wejściowych. W ramach działań aktualizacyjnych powrócono do przygotowanej w Zad. 3 listy danych GTFS dostępnych publicznie. W większości przypadków lista ta pozwoliła na pozyskanie nowych wersji rozkładów. Były jednak pewne wyjątki, m.in. przestał działać przyjazdy.pl. Jednak część danych pierwotnie pozyskanych z tego serwisu i tak udało się pozyskać bezpośrednio od organizatorów. Miało to miejsce np. w przypadku przewoźnika "Koleje Wielkopolskie".

Poniżej przedstawiono listę wybranych 55 podmiotów, dla których pozyskano i wprowadzone zaktualizowane dane rozkładowe:

1. Beskid (linia Wrocław-Wieluń)
2. CAR - BON Bonecki Travel MAREK BONECKI
3. Darbus Sp. Jawna Dariusz Stachura, Małgorzata Burda
4. DARTUR Usługi Transportowe DARIUSZ KUZAK
5. Edyta Puchała-Szymoniak Transport Osobowo - Towarowy
6. FIRMA PRZEWOZOWA "FB"
7. Firma Przewozowa BMC BOGDAN MASŁOWIEC
8. Firma Transportowo-Handlowa Kazimierz Czosnyka
9. Giewont J.Kurnyta i inni
10. KA w Świnoujściu
11. KM Szczecinek
12. Koleje Dolnośląskie
13. Koleje Śląskie
14. Kórnickie Przedsiębiorstwo Autobusowe "KOMBUS" Spółka z o.o.
15. Łódzka Kolej Aglomeracyjna
16. MalTrans Wiesław Malinowski
17. Marco Polo (Poznań)
18. Matmich-BUS Przewóz Osób Mateusz Michalak
19. Miejski Zakład Komunikacji w Koninie Spółka z o.o.

20. MPK Częstochowa
21. MPK Inowrocław
22. MPK w Łomży
23. MZK Ełk
24. MZK Stargard
25. MZK w Toruniu
26. NOVABUS Magdalena Woźniak
27. NOVABUS Magdalena Woźniak (Siechnice)
28. P.U.H. Koltrans Sp. J.
29. P.W. BESKID
30. PGKiM w Sandomierzu
31. PKS Bielsko
32. PKS Kalisz
33. PKS w Kluczborku sp. z o.o.
34. PKS w Łosicach sp. z o.o.
35. PKS w Strzelcach Opolskich S.A.
36. Pleszewskie Linie Autobusowe
37. PRZEWÓZ OSÓB Adam Grzywa
38. Przewóz Osób Jerzy Buzala
39. PRZEWÓZ OSÓB Stanisław Osika
40. RPK Bochnia
41. Stanisław Bodzioch Firma Przewozowa
42. "TELESFOR" TRANSPORT OSOBOWY T. Głowacki i P. Głowacki
43. Transport Osobowo Towarowy JANPOL
44. TRANSPORT OSOBOWY Maciej Piotr Naglak
45. Transport osobowy Mini-Bus Andrzej Domagała
46. UM Lidzbark Warmiński
47. UM Wrocław
48. Usługi Transportowe Towarowe i Osobowe Świder
49. Warszawska Kolej Dojazdowa
50. Zakład Komunalny w Kostrzynie
51. ZDZiT Olsztyn

52. Zielonogórski Związek Powiatowo-Gminny

53. ZTM Poznań

54. ZTM Rzeszów

55. Żegluga Ostródzko-Elbląska

Ostatecznie do wygenerowania mapy przygotowano 1322 źródła GTFS. 133 źródła nie zostały wykorzystane z uwagi na ich zastąpienie nowszymi i/lub poprawionymi wersjami lub występowanie nienaprawialnych błędów - głównie niewystarczającą ilość przenoszonych danych, tj. wybrakowanie.

Pozyskane źródła pozwoliły na wygenerowanie w skali kraju łącznie 66 milionów planów podróży. W efekcie stwierdzono, że zasięg transportu zbiorowego w Polsce dotyczy

- 86% punktów adresowych z dostępem do PTZ/PR.
- 14% to punkty adresowe, które nie znajdują się w pobliżu ani jednego przystanku - to tzw. białe plamy.

Natomiast po uwzględnieniu wszystkich składowych wykluczenia, **liczba wykluczonych PA w kraju wyniosła 39%, tj. 3 217 407 z 8 260 929 PA**. Odsetek wykluczonych PA oczywiście nie jest równomierny i wynosi:

- 51% na obszarach wiejskich (kody TERC = 2 lub 5)
- 19% na obszarach miejskich (kody TERC 1, 4, 8 lub 9).

Warto przy tym zauważyć, że w miastach z uwagi na ich charakter również występują różnice, tj.:

- w miastach będących gminami miejskimi (niejednokrotnie siedzibami powiatów) 11% PA jest wykluczonych,
- w miastach zlokalizowanych w gminach miejsko-wiejskich wykluczonych jest aż 40% PA.

Sytuacja w tych miasteczkach była zatem jedynie o 11 punktów procentowych niższa niż dla obszarów wiejskich.

Opracowany algorytm umożliwia wskazywanie przyczyn wykluczenia. W skali kraju wartości te wynosiły:

- brak aktywnego przystanku w pobliżu: 36% wykluczonych PA;
- niewystarczająca liczba połączeń do siedziby powiatu: 15% wykluczonych PA;
 - brak połączeń do siedziby powiatu pomimo istnienia oferty przewozowej: 5% wykluczonych PA;
- niewystarczająca liczba (pojedynczych) kursów spełniająca wszystkie składowe na raz: 8% wykluczonych PA;
- wynikające ze złożenia wielu podróży (przekroczenie dopuszczalnego budżetu w celu osiągnięcia 4 par połączeń do punktu docelowego, przy braku takiego przekroczenia na pojedynczym kursie): 2% wykluczonych PA;
- brak spełnienia przynajmniej jednej składowej WK dla pojedynczej podróży: 40% wykluczonych PA:
 - brak spełnienia wyłącznie składowej dystansu segmentów pieszych: 11%;
 - brak spełnienia wyłącznie składowej czasu podróży: 2%;
 - brak spełnienia wyłącznie składowej kosztu pojedynczej podróży: 11%;
 - brak spełnienia wyłącznie składowej niezawodności: 0,2%.

Udostępnione w trakcie trwania projektu wyniki Narodowego Spisu Powszechnego 2021 pozwoliły estymować informacje o wykluczonych PA na informacje o odsetku wykluczonej populacji. Wykorzystano do tego celu informacje o populacji w obwodach spisowych, co wymagało przygotowania odpowiedniej warstwy GIS oraz porównania ją z pozyskanymi z bazy Państwowego Rejestru Granic warstwy lokalizującej punkty adresowe. W związku z różnicami w tych bazach nie udało się odwzorować miejsca zamieszkania 41 576 osób, tj. 1 promila populacji wynoszącej wg udostępnionych danych NSP 2021 38 036 118 osób. Rozbieżność tę uznano za niewpływającą znacząco na rezultaty z wyjątkiem gminy Maszewo w województwie zachodniopomorskim, która w momencie pozyskiwania bazy lokalizacji punktów adresowych zawierała w PRG tylko jeden punkt adresowy (tj. występował błąd w tych danych źródłowych).

Szacunki demograficzne w przełożeniu na analizę wskaźnika WK w PA wskazały, że **wykluczone jest 27% mieszkańców Polski, tj. 10,1 mln osób**. Analiza w uznanych za

szczególnie zagrożone zjawiskiem wykluczenia z transportu zbiorowego podgrupach wiekowych nastolatków (osoby w wieku 10-19 lat) oraz seniorów (osoby w wieku poprodukcyjnym) nie wykazały znaczących różnic od średniej - wartości wykluczonej populacji wynosiły odpowiednio 26 i 24%. Zatem ilościowo **grupy te nie są bardziej narażone na wykluczenie**. Należy jednak pamiętać, że w przypadku jego wystąpienia może być ono dla tych grup bardziej dotkliwe w związku z częstym brakiem możliwości korzystania z samochodu, jako kierowca. Ponadto szacunki demograficzne potwierdziły dużą rozbieżność między mieszkańcami obszarów miejskich oraz wiejskich. Zagrożonych wykluczeniem było 11% mieszkańców miast (tj. 2,4 mln osób) oraz 50% mieszkańców obszarów wiejskich, tj. 7,7 mln osób.

Więcej wyników zawarto w raporcie z Zad. 21. Pliki źródłowe niezbędne do wykonania mapy wykluczenia dodano w formie załączników (standard csv oraz GeoPackage).

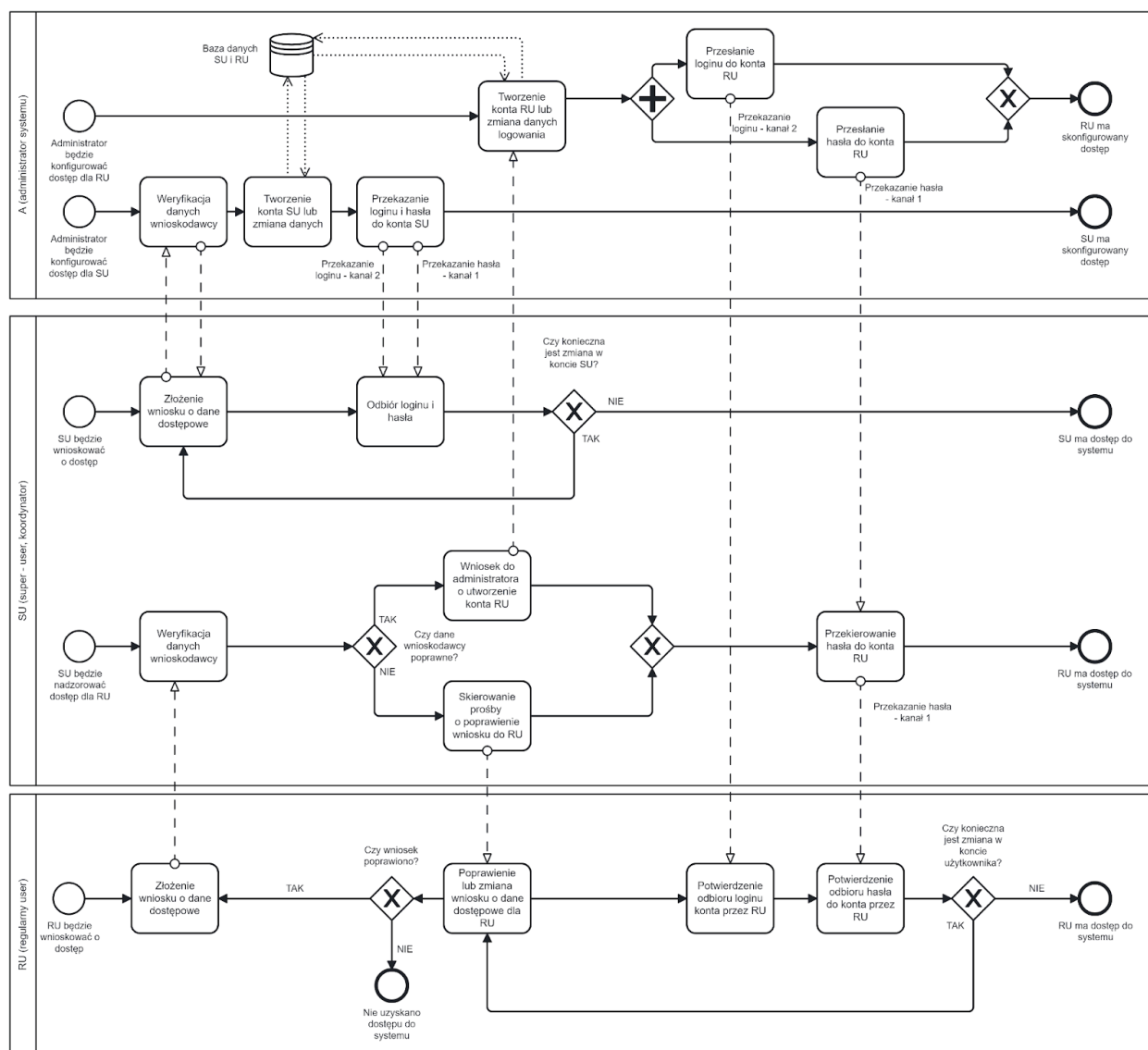
3. Opracowanie zasad dostępu do systemu informacji o wykluczeniu przez jednostki trzecie

W działaniu D2, w ramach prac projektowych prowadzono prace nad zasadami dostępu do systemu informacji o wykluczeniu przez jednostki trzecie, tj. np. jednostki samorządu terytorialnego, podmioty gospodarcze zainteresowane planowaniem podróży, itp. Ostatecznie wypracowano trzy poziomy dostępu do aplikacji dla różnego rodzaju podmiotów, takich jak jednostki samorządu terytorialnego czy podmioty gospodarcze zainteresowane planowaniem podróży:

- niezalogowany użytkownik (NU) - dostęp wyłącznie do mapy wykluczenia; przeznaczony dla ogółu podmiotów;
- regularny użytkownik (RU) - dostęp do przypisanych organizatorów; przeznaczony dla organizatorów, którzy mogą modyfikować własne informacje rozkładowe;
- super użytkownik, koordynator (SU) - dostęp do wszystkich organizatorów; przeznaczony dla podmiotów koordynujących, by umożliwić im wsparcie działań RU.

Ponadto możliwy jest dostęp administracyjny (administrator systemu A) do backendu aplikacji, który umożliwia wprowadzanie zmian, których nie przewidziano z poziomu interfejsu użytkownika, w tym takich, które mogłyby doprowadzić do uszkodzenia danych w przypadku stosowania przez niedoświadczonych użytkowników.

Użytkownicy regularni i super użytkownicy do zalogowania wymagają loginu i hasła, które może zostać utworzone przez administratorów systemu. Stąd w zakresie obowiązków przynajmniej części z nich powinny znajdować się odpowiednie zapisy. Rekomenduje się, by login i hasło do systemu były dostarczane różnymi kanałami komunikacji. Na rys. 1 przedstawiono w notacji BPMN zaproponowaną przez zespół autorski procedurę przyznawania dostępu do systemu obu tym grupom użytkowników. W razie potrzeby, w procesie tym można także uwzględnić procedurę odwołań sądowych w przypadku, gdy potencjalny UR otrzymał od SU decyzję odmowną przyznania takiego statusu lub zgłoszenie zostało zwrócone do poprawek, z którymi użytkownik się nie zgadza.



Rys. 1. Proces tworzenia kont dla użytkowników regularnych (RU) oraz super użytkowników - koordynatorów (SU) w systemie wraz z aktualizacją danych do logowania.

Należy zauważyć, że użytkownicy regularni oraz super użytkownicy mają dostęp do systemu nie tylko z poziomu frontendu (przeglądarki www), ale też przez API, które zostało szerzej opisane w raporcie z Zad.10, a w niniejszym raporcie znajduje się jego tłumaczenie. Ponadto same procedury zakładania kont użytkowników oraz nadawania im uprawnień były testowane i wykorzystywane zarówno na potrzeby własne zespołu projektowego (składającego się z czterech podmiotów), jak i w związku z różnego rodzaju konsultacjami opracowywanego systemu z podmiotami zewnętrznymi. Na potrzeby realizacji działań, na serwerze produkcyjnym utworzono łącznie 94 konta użytkowników.

4. Wykonanie specyfikacji standardów obowiązujących w podsystemie informacji o funkcjonowaniu PTZ w języku angielskim

W ramach działania D3 wykonano tłumaczenie do języka angielskiego specyfikacji standardów obowiązujących w podsystemie informacji o funkcjonowaniu wraz z opisem dostępu przez API. Założono, że przygotowanie dokumentacji w j. angielskim uprości wykorzystanie systemu przez podmioty zagraniczne, które mogą być nim zainteresowane np. ze względu na oferowanie oprogramowania dla organizatorów PTZ czy obsługę aktywności turystycznych na terenie Polski. Efekty działania został zamieszczone w postaci załącznika.

5. Osiągnięcie kamieni milowych zadania

5.1. Aktualizacja mapy zagrożenia WK

Przeprowadzone działania aktualizacyjne miały szerszy zakres niż wymagany w tym kamieniu milowym. Wskazano listę 55 organizatorów PTZ lub przewoźników względem zakładanych 20. **Kamień milowy został zatem osiągnięty.** Przy czym należy w tym miejscu zauważyć, że pełna i bieżąca aktualizacja wszystkich danych przy jednoczesnym założeniu względnie niskich kosztów tego procesu wymaga pełnej cyfryzacji obiegu informacji pasażerskiej w Polsce przy jednoczesnej koordynacji tego procesu. Brak koordynacji procesu może skutkować niską jakością danych, pomimo ich cyfryzacji, co zespół projektowy widział dla danych wykorzystanych do generacji mapy.

5.2. Procedura dostępu do systemu przez jednostki trzecie

W ramach utworzonego systemu założono występowanie trzech kategorii użytkowników oraz administratorów systemu, którzy m.in. mogą nadawać loginy oraz hasła nowym użytkownikom, wraz ze wskazywaniem zakresu dostępu do danych rozkładowych, tj. przypisywaniem organizatorów. **Można zatem uznać, że kamień milowy został osiągnięty.**

5.3. Tłumaczenie na język angielski specyfikacji rozszerzonych standardów danych dot. funkcjonowania PTZ w Polsce

Trzecim kamieniem milowym było uzyskanie jednego tłumaczenia standardów. Tłumaczenie to zawarto w załączniku do raportu. **Można zatem uznać, że kamień milowy został osiągnięty.**

6. Podsumowanie

W efekcie realizacji Zad. 20 zwiększyła się gotowość do wdrożenia opracowanego w ramach projektu systemu. W szczególności opracowano finalną wersję mapy wykluczenia, która może posłużyć do realizacji innych projektów.

Zaleca się jednak, by po wdrożeniu prace nad rozwojem systemu były kontynuowane, a powstała mapa wykluczenia była okresowo aktualizowana, gdyż z perspektywy uzyskanych rezultatów mniejsze znaczenie ma ostatecznie wyznaczona liczba wykluczonych osób (z perspektywy etycznej, czy ludzkiej, nawet sytuacja, w której wykluczona jest tylko jedna osoba, powinna być już nieakceptowalna), co możliwość obserwacji zmian poziomu wykluczenia w efekcie wdrażania zmian w polityce transportowej czy mobilnościowej. Ponadto warto zauważyć w tym miejscu, że opracowana definicja wykluczenia komunikacyjnego była konstruowana z myślą o niwelowaniu działań mających na celu wyłącznie "spełnienie narzuconego wskaźnika" bez zwiększania efektywności i jakości systemu PTZ z perspektywy użytkownika. Przyjęte podejście powinno być odporne m.in. na takie działania, jak:

- tworzenie oferty, która jest dostępna tylko dla wybranych mieszkańców JST w celu "udowodnienia" obsługi wszystkich mieszkańców tej JST (analiza prowadzona na poziomie punktów adresowych),
- tworzenie bardzo wielu linii, które składają się z pojedynczych kursów czy przeciwnie tworzenie pojedynczych linii, które zawierają wiele kursów w wielu wariantach ich tras - badana jest możliwość połączenia z miastem powiatowym,
- tworzenie nieskomunikowanych linii, które działają na małym obszarze, a w efekcie nie zapewniają dostępu do siedziby powiatu (promowanie tworzenia spójnego systemu PTZ bez jednoczesnego rozstrzygania (wymuszania) stopnia sformalizowania i formuły współpracy różnych organizatorów, a zatem zezwalanie na wybór najbardziej efektywnych rozwiązań w zależności od potrzeb podmiotów),
- tworzenie długich jednokierunkowych linii okólnych w celu spełnienia obszarowego kryterium liczby kursów - linie te nie będą spełniały kryterium niezawodnościowego i/lub czasu podróży.

Należy mieć jednak na względzie, że nawet tak sformułowany wskaźnik będzie mógł podlegać próbom sabotowania, szczególnie w przypadku, gdy z jego określonymi wartościami będzie wiązać się ze zróżnicowaniem wysokości dofinansowania. Ponadto, należy spodziewać się, że w przyszłości mogą rozwinąć się na większą skalę takie rozwiązania, jak transport na życzenie/żądanie czy technologie pozwalające oceniać występujące opóźnienia i wypadania kursów na podstawie danych lokalizacyjnych (obecnie dostępne głównie w przypadku kolei oraz większych aglomeracji). Stąd proponuje się kontynuowanie działań monitorujących oraz rozwojowych także w zakresie algorytmicznym projektu.

Warto także podkreślić, że na jakość analizy skali wykluczenia duży wpływ ma jakość danych, tj. rozkładów jazdy i cenników. W trakcie wykonywania projektu zespół projektowy spotykał się z przypadkami, gdy dane te, także pozyskane w formie cyfrowej, były wątpliwej jakości. Jakkolwiek na potrzeby projektu podjęto działania mające na celu poprawę tej jakości, to należy zauważyć, że działania te nie spowodowały poprawy jakości danych źródłowych udostępnianych przez ich producentów¹, a w efekcie nie miało większego wpływu na postrzeganie systemu transportowego przez użytkowników (pasażerów). Stąd warto jeszcze raz podkreślić rekomendację utworzenia jednostek koordynujących systemy transportu zbiorowego, w tym zajmujących się kuracją danych, także cyfrowych. W odczuciu zespołu autorskiego najbardziej odpowiedni do realizacji takich działań jest szczebel wojewódzki JST, gdyż z jednej strony ciągle jest on dość blisko regionalnych i lokalnych systemów transportowych (pracownicy mają dość dużą wiedzę w zakresie ich topografii oraz problemów), a jednocześnie jednostki te posiadają zasoby (ludzkie, finansowe), by móc sprostać takim zadaniom. Autorzy zauważyli wręcz, że w niektórych samorządach województw już podjęto działania w tym zakresie, np. inwentaryzacji przystanków czy budowania regionalnych portali rozkładów jazdy. Ponadto należy zauważyć, że wsparcie producentów informacji rozkładowej ma dużo większe znaczenie od budowy infrastruktury po stronie konsumenckiej ("portale rozkładów jazdy", "wyszukiwarki połączeń"), gdyż:

- bez wysokiej jakości danych rozwiązania te będą tak samo nieefektywne i wprowadzające w błąd, co obecne rozwiązania "papierowe",

¹ Przy czym niezależnie w tym celu zespół projektowy podejmował inne, niezależne inicjatywy, takie jak regionalna baza przystanków w województwie wielkopolskim.

- rozwiązania te będą ponadto oferowały niższą jakość niż rozwiązania komercyjne (gdzie kuracją danych często zajmuje się dostawca aplikacji),
- ponadto w celu zauważenia przez użytkowników rozwiązania te będą wymagały promocji (w efekcie wydawania na nią pieniędzy podatnika).

Zespół projektowy rekomenduje w tym zakresie prowadzenie monitoringu dostępnych rozwiązań komercyjnych i interwencję rynkową tylko w przypadku, gdy będą one niewystarczające pomimo polepszenia jakości danych źródłowych. Podobne działania należy podejmować w sytuacji, gdy zmieniają się preferencje użytkowników, np. wystąpi duży spadek do rozwiązań komercyjnych związany np. z obawami związanymi z prywatnością lub obawami o uwzględnianie ich interesów, a nie interesów innych podmiotów, co w efekcie będzie wpływać na generowane wyniki planów podróży, np. wspieranie droższych przewoźników, którzy opłacili wyższą pozycję w wynikach wyszukiwania. Nie wyłącza to jednak możliwości udostępniania publicznie części rozwiązań informatycznych, które i tak będą musiały powstać w celu umożliwienia podmiotom koordynującym skuteczną kurację cyfrowych danych transportowych. Nie wyłącza to także sensu prowadzenia innych działań mających na celu promocję cyfrowych standardów wymiany informacji transportowej, także poprzez szkolenia czy rozwój narzędzi umożliwiających, nie tylko pracownikom JST, ale też ogółowi społeczeństwa wgląd w dane i ich wykorzystanie, np. rozwój otwartej wtyczki GTFS-GO do otwartego oprogramowania QGIS, dzięki której można z pliku GTFS wygenerować i zwizualizować sieć transportu zbiorowego.

Załączniki

Załącznik 1 - Specyfikacja standardów obowiązujących w podsystemie informacji o funkcjonowaniu PTZ przygotowana w języku angielskim

The standard used in the Public Transport Information Subsystem includes an API specification that enables access to it.

The following sections present the concept behind the standard, the method of its definition, and an example demonstrating its use and complete definition.

Z1a. Technology for preparing API and its documentation

API (Application Programming Interface) is a set of rules that precisely define communication between programs or subprograms. It is one of the three types of computing interfaces (alongside user interfaces and physical interfaces). In essence, an API is a method of communication between different elements of software. There are various types of APIs. For the purposes of the solution used in this project, a **Web API** was created—a type of networked application programming interface that uses web architecture and protocols for communication between applications running on different devices within a network.

In the case of a Web API, the resources are not accessed by a human using a browser, but by another application. From a technical standpoint, access to resources is achieved by invoking methods on the Web API installed on the server side. An API can be consumed by different applications, such as web browsers, mobile apps, or desktop applications. In the case of a Web API, the protocol that governs communication between the server and enables access to resources is **HTTP (Hypertext Transfer Protocol)**. A **resource** should be understood as an object that is being accessed.

The communication between the client and server is based on sending a **request** and receiving a **response**. The HTTP protocol defines several methods that allow operations on resources. The method name is sent within the HTTP request.

For the API developed in this project, four standard HTTP methods were used:

- **GET** – to retrieve a resource
- **POST** – to create a resource
- **PUT** – to update a resource
- **DELETE** – to delete a resource

Web API methods are made available through **URI (Uniform Resource Identifier)** addresses, which make it easy to identify network resources. For example, a GET request to the address `/api/stop` returns a list of bus stops, while a GET request to

`/api/stop/{id}` returns the stop with the specified ID. Similarly, a POST request to `/api/stop` adds a new stop. Upon receiving a request, the Web API on the server performs the necessary operations and sends the result back to the client application. The format used for data exchange is **JSON**.

The HTTP protocol defines five groups of three-digit status codes for server responses to requests:

- **1XX** – informational
- **2XX** – success
- **3XX** – redirection
- **4XX** – client error
- **5XX** – server error

For the API developed for the project, the most commonly used server response codes were:

- **200 – OK:** The requested content was returned
- **201 – Created:** A resource was successfully created
- **204 – No Content:** The method succeeded, but the response body is empty
- **400 – Bad Request:** The request sent to the server was invalid
- **401 – Unauthorized:** Access is unauthorized
- **403 – Forbidden:** Access to the requested resource is denied
- **404 – Not Found:** The requested resource could not be found
- **409 – Conflict:** The operation attempted is prohibited due to a conflict
- **500 – Internal Server Error:** The server encountered an error while processing the request

For the designed Web API, documentation was prepared showing all available methods along with their URI endpoints, response bodies, and server response codes. The documentation was created using the **SWAGGER (OpenAPI)** tool.

Z1b. WebAPI Documentation Contents in SWAGGER

WebAPI documentation prepared in Swagger (OpenAPI) provides a complete description of how to use the API interface — both for humans (e.g., developers, testers) and machines (e.g., client code generators). The documentation is based on the OpenAPI specification and typically includes the following elements:

1. General API Information (**info**)

- **API name** (e.g., "Gospostrateg"),
- **API version** (e.g., "v1"),
- **API description** – general purpose and scope (optional),
- **Authors / Contact** – contact details for the team maintaining the API (optional),
- **License** – if applicable (e.g., MIT, Apache 2.0).

2. List of Resources (Endpoints) (**paths**)

Each API resource (e.g., `/api/Agency`) includes:

- **Path** (e.g., `/api/Agency/{id}`),
- **Available HTTP methods**: GET, POST, PUT, DELETE, etc.,
- **Operation description** – what the method does (e.g., "retrieves the PTZ operator/organizer with the specified ID"),
- **Tags** – grouping of methods (e.g., by controller or data type).

3. Parameters

Detailed description of required and optional parameters:

- **Query parameters** – e.g., `?page=1&size=20`,
- **Path parameters** – e.g., `/api/Agency/{id}`,
- **Header parameters** – e.g., `Authorization: Bearer <token>`,
- **Body (request body)** – data sent in the request payload (e.g., `application/json`).

Each parameter includes:

- Name,
- Data type (e.g., string, integer, boolean),
- Required status (`required: true/false`),
- Description (optional).

4. Data Schemas (**components/schemas**)

Defined data models used in requests and responses, such as:

- `AgencyDTO` – object representing a PTZ operator,
- `ProblemDetails` – standard error object,
- `UserDTO`, `ProjectDTO`, etc.

Each model defines:

- Fields (name, type, format – e.g., `int32`, `date-time`),
- Whether the field is required,
- Any constraints (e.g., length, range).

5. Responses

Description of possible HTTP responses:

- **HTTP status codes:** `200 OK`, `400 Bad Request`, `404 Not Found`, `500 Internal Server Error`, etc.,
- **Response description,**
- **Returned data type** – e.g., `application/json`,
- **Schema of returned data** – e.g., `AgencyDTO`, list of objects, etc.

Z1c. Examples of WebAPI documentation fragments prepared for the project

The WebAPI documentation in Swagger is provided in JSON format. For the purpose of this report, an example is shown of how the WebAPI documentation describes a set of operations used to manage objects containing information about PTZ (public transport) organizers. The endpoints are organized according to REST principles and support standard CRUD operations (Create, Read, Update, Delete), as shown in Figure 1.

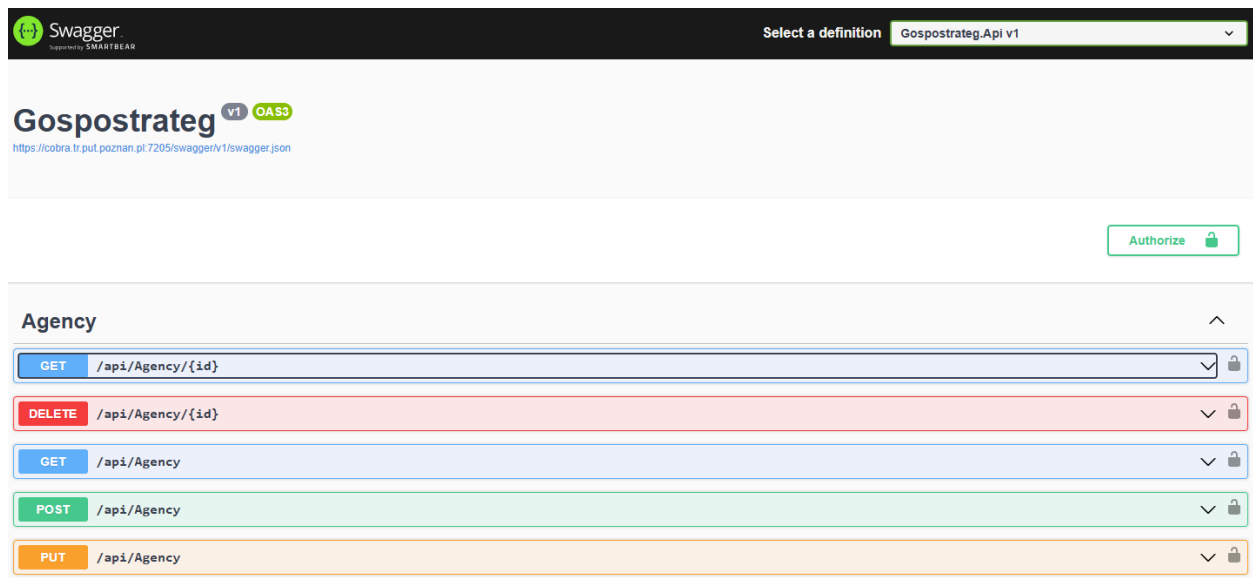


Fig. 1. Swagger WebAPI screenshot for the developed system.

1. Endpoint: GET /api/Agency/{id}

Used to retrieve detailed information about a PTZ organizer based on its identifier (parameter `id` in the path).

Parameters:

- `id` (integer, required) – organizer identifier.

Responses:

- `200 OK` – returns an `AgencyDTO` object in JSON format,
- `400 Bad Request` – validation error,

- **404 Not Found** – the agency with the given ID does not exist.

2. Endpoint: **DELETE /api/Agency/{id}**

Allows removing an organizer from the system.

Parameters:

- **id** (integer, required) – identifier of the organizer to delete.

Responses:

- **200 OK** – confirmation of deletion,
- **400 Bad Request** – request error (e.g., invalid ID).

3. Endpoint: **GET /api/Agency**

Returns a list of organizers available in the system.

Responses:

- **200 OK** – list of **AgencyDTO** objects,
- **400 Bad Request** – validation error,
- **404 Not Found** – no data found.

4. Endpoint: **POST /api/Agency**

Used to create a new agency.

Request body:

- An **AgencyDTO** object in **application/json** or **text/json** format.

Responses:

- **200 OK** – agency successfully created,
- **400 Bad Request** – input data error.

5. Endpoint: **PUT** /api/Agency

Allows editing an existing organizer in the system.

Request body:

- An **AgencyDTO** object containing updated data.

Responses:

- **200 OK** – update successful,
- **400 Bad Request** – incorrect data or structure.

Additionally, the **AgencyDTO** object, which represents a PTZ (public transit) organizer, is described in the data schemas according to Table 1. Fields marked as **Nullable: yes** can be empty or absent in the data. This object plays a key role in creating, editing, and displaying organizer/operator data via the API. As can be observed, some fields are specific to the Polish context, such as the distinction between operator and organizer, or the use of the NIP (Polish tax identification number).

Table 1. Structure of the *AgencyDTO* object

Field Name	Data Type	Nullable	Description
id	integer (int32)	no	Unique identifier of the organizer in the system.
city	string	yes	Name of the city where the organizer operates.
district	string	yes	District or administrative area.
email	string	yes	Organizer's email address.
idExternal	string	yes	External organizer ID (e.g., from another system).
municipality	string	yes	Municipality to which the organizer is assigned.
name	string	yes	Full name of the organizer.
phone	string	yes	Contact phone number.
postalCode	string	yes	Organizer's postal code.
primaryNumber	string	yes	Building number (main).
secondaryNumber	string	yes	Apartment or secondary number.
streetName	string	yes	Street name.
type	string	yes	Type (e.g., operator, transport organizer).
url	string	yes	Official website of the organizer.
idFeed	integer (int32)	no	Identifier of the associated GTFS data feed.
urlFare	string	yes	Link to fare or pricing information.
organiserId	integer (int32)	yes	ID of the parent organizer (if applicable).
nip	string	yes	Organizer's NIP (tax identification number).
serviceNotes	string	yes	Notes or additional information about services.
gtfsSourceUrl	string	yes	Source URL of GTFS data.

Z1d. List of Modules Containing Endpoints Described in the WebAPI Documentation

Agency

Handles the management of public transport organizers (PTZ). Enables the creation, editing, retrieval, and deletion of data related to these entities, including contact details, addresses, and external identifiers (e.g., GTFS).

Calendar

Manages the transit calendar – allows defining weekdays and date ranges during which trips are active. Used when assigning schedules to trips.

CalendarDate

Enables the introduction of calendar exceptions, such as additional service days or exclusions (e.g., holidays, non-service days). Allows modifying the default calendar rules.

CommunicationExclusion

Responsible for operations related to representing the phenomenon of transport exclusion.

Export

Enables exporting system data into the GTFS format structure.

FareAttributes

Manages fare-related data: pricing, zones, fare types. This module allows for creating, editing, and deleting fare attributes assigned to PTZ operators.

Geoposition

Allows searching for geographic locations based on text queries (e.g., street names, towns, addresses). Used to support spatial data input.

GTFS

Module for managing GTFS data – allows importing feeds, assigning them to PTZ organizers, browsing, and updating related data.

Organiser

Manages data related to public transport organizers – such as municipal offices or transport associations. Enables integration with the national GUS register, assignment to operators, and editing of contact information.

Route

Handles the management of transit routes. Allows defining, updating, and associating routes with PTZ operators. Routes serve as the basis for creating trips and schedules.

Stop

Manages stop data: location, names, identifiers, and their association with routes and trips. Allows for creating, editing, and deleting transit stops.

StopTime

Enables the detailed definition of arrival and departure times at stops within trips. Supports both single and batch operations.

Trip

Manages public transport trips – specific route executions at defined times. Handles creation, editing, assignment to routes and calendars, and cloning of trips.

Users

Responsible for managing system users. Supports authentication, session (token) refresh, logout, and access to user and session data.

As can be observed, the modules can be divided into two groups:

- **Those directly related to the GTFS structure** (e.g., *Agency, Trip, Stop, Route*)
- **Additional modules** that support other functionalities, such as user authentication or retrieving the full GTFS data repository.

Z1e. Complete standard definition

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "Gospostrateg",
    "version": "v1"
  },
  "paths": {
    "/api/Agency/{id}": {
      "get": {
        "tags": [
          "Agency"
        ],
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int32"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/AgencyDTO"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/ProblemDetails"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/ProblemDetails"
                }
              }
            }
          }
        }
      },
      "delete": {
```

```
"tags": [
  "Agency"
],
"parameters": [
  {
    "name": "id",
    "in": "path",
    "required": true,
    "schema": {
      "type": "integer",
      "format": "int32"
    }
  }
],
"responses": {
  "200": {
    "description": "Success"
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
}
},
"/api/Agency": {
  "get": {
    "tags": [
      "Agency"
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/AgencyDTO"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
```

```
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "post": {
      "tags": [
        "Agency"
      ],
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/AgencyDTO"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/AgencyDTO"
            }
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/AgencyDTO"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "put": {
    "tags": [
      "Agency"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/AgencyDTO"
          }
        }
      },
      "text/json": {
```

```
        "schema": {
          "$ref": "#/components/schemas/AgencyDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/AgencyDTO"
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/Calendar": {
    "get": {
      "tags": [
        "Calendar"
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/CalendarDTO"
              }
            }
          }
        },
        "400": {
          "description": "Bad Request",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        }
      }
    }
  }
}
```

```
        }
      }
    }
  },
  "post": {
    "tags": [
      "Calendar"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CalendarWithDatesDTO"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CalendarWithDatesDTO"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/CalendarWithDatesDTO"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "put": {
    "tags": [
      "Calendar"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CalendarDTO"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CalendarDTO"
          }
        }
      }
    }
  }
}
```



```
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/CalendarDTO"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Calendar/{tripId}": {
  "get": {
    "tags": [
      "Calendar"
    ],
    "parameters": [
      {
        "name": "tripId",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/CalendarsByTripDto"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
}
```

```

    }
  },
  "404": {
    "description": "Not Found",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
},
"/api/Calendar/{id}": {
  "delete": {
    "tags": [
      "Calendar"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
},
"/api/CalendarDate": {
  "get": {
    "tags": [
      "CalendarDate"
    ],
    "parameters": [
      {
        "name": "calendarId",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ]
  }
}

```

```
    }
  }
],
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "application/json": {
        "schema": {
          "type": "string",
          "format": "date-time"
        }
      }
    }
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "404": {
    "description": "Not Found",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
},
"post": {
  "tags": [
    "CalendarDate"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      }
    }
  }
},
},
```

```
"responses": {
  "200": {
    "description": "Success"
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
},
"put": {
  "tags": [
    "CalendarDate"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/CalendarDateDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/CalendarDate/Active": {
  "get": {
    "tags": [
      "CalendarDate"
    ]
  }
}
```

```
    ],
    "parameters": [
      {
        "name": "calendarIds",
        "in": "query",
        "schema": {
          "type": "array",
          "items": {
            "type": "integer",
            "format": "int32"
          }
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "type": "string",
              "format": "date-time"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/CalendarDate/{id}": {
    "delete": {
      "tags": [
        "CalendarDate"
      ],
      "parameters": [
        {
          "name": "id",
          "in": "path",
          "required": true,
          "schema": {
```

```

        "type": "integer",
        "format": "int32"
    }
}
],
"responses": {
    "200": {
        "description": "Success"
    },
    "400": {
        "description": "Bad Request",
        "content": {
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/ProblemDetails"
                }
            }
        }
    }
}
},
"/api/CommunicationExclusion": {
    "get": {
        "tags": [
            "CommunicationExclusion"
        ],
        "parameters": [
            {
                "name": "NELatitude",
                "in": "query",
                "schema": {
                    "type": "number",
                    "format": "double"
                }
            },
            {
                "name": "NELongitude",
                "in": "query",
                "schema": {
                    "type": "number",
                    "format": "double"
                }
            },
            {
                "name": "SWLatitude",
                "in": "query",
                "schema": {
                    "type": "number",
                    "format": "double"
                }
            },
            {
                "name": "SWLongitude",
                "in": "query",
                "schema": {
                    "type": "number",
                    "format": "double"
                }
            }
        ]
    }
},

```

```

        {
            "name": "zoomLevel",
            "in": "query",
            "schema": {
                "type": "integer",
                "format": "int32"
            }
        }
    ],
    "responses": {
        "200": {
            "description": "Success",
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/CommunicationExclusionDTO"
                    }
                }
            }
        },
        "400": {
            "description": "Bad Request",
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/ProblemDetails"
                    }
                }
            }
        },
        "404": {
            "description": "Not Found",
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/ProblemDetails"
                    }
                }
            }
        }
    }
},
"/api/CommunicationExclusion/AllPoints": {
    "get": {
        "tags": [
            "CommunicationExclusion"
        ],
        "responses": {
            "200": {
                "description": "Success",
                "content": {
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/File"
                        }
                    }
                }
            },
            "400": {

```

```
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    },
    "404": {
      "description": "Not Found",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  },
  "/api/Export/Structured": {
    "get": {
      "tags": [
        "Export"
      ],
      "parameters": [
        {
          "name": "agencyId",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        },
        {
          "name": "routeId",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        },
        {
          "name": "isReturningDirection",
          "in": "query",
          "schema": {
            "type": "boolean"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/Export/Unstructured": {
```



```
"get": {
  "tags": [
    "Export"
  ],
  "parameters": [
    {
      "name": "agencyId",
      "in": "query",
      "schema": {
        "type": "integer",
        "format": "int32"
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Success"
    }
  }
},
"/api/FareAttributes/{agencyId}": {
  "get": {
    "tags": [
      "FareAttributes"
    ],
    "parameters": [
      {
        "name": "agencyId",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/FareAttributesDTO"
              }
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
}
```

```
    },
    "404": {
      "description": "Not Found",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  },
  "/api/FareAttributes": {
    "post": {
      "tags": [
        "FareAttributes"
      ],
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/FareAttributesDTO"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/FareAttributesDTO"
            }
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/FareAttributesDTO"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "put": {
    "tags": [
      "FareAttributes"
    ],
    "requestBody": {
      "content": {
```

```
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/FareAttributesDTO"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/FareAttributesDTO"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/FareAttributesDTO"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/FareAttributes/{id}": {
  "delete": {
    "tags": [
      "FareAttributes"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
}
```

```
    }
  }
}
},
"/api/Geoposition/{query}": {
  "get": {
    "tags": [
      "Geoposition"
    ],
    "parameters": [
      {
        "name": "query",
        "in": "path",
        "required": true,
        "schema": {
          "type": "string"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/AgencyDTO"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
},
"/api/Gtfs/{organiserId}": {
  "post": {
    "tags": [
      "Gtfs"
    ]
  }
}
```

```
    ],
    "parameters": [
      {
        "name": "organiserId",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      },
      {
        "name": "exportable",
        "in": "query",
        "schema": {
          "type": "boolean"
        }
      }
    ],
    "requestBody": {
      "content": {
        "multipart/form-data": {
          "schema": {
            "type": "object",
            "properties": {
              "formFile": {
                "type": "string",
                "format": "binary"
              }
            }
          },
          "encoding": {
            "formFile": {
              "style": "form"
            }
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/Gtfs": {
    "post": {
      "tags": [
        "Gtfs"
      ]
    }
  }
}
```

```
    ],
    "parameters": [
      {
        "name": "organiserId",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      },
      {
        "name": "exportable",
        "in": "query",
        "schema": {
          "type": "boolean"
        }
      }
    ],
    "requestBody": {
      "content": {
        "multipart/form-data": {
          "schema": {
            "type": "object",
            "properties": {
              "formFile": {
                "type": "string",
                "format": "binary"
              }
            }
          },
          "encoding": {
            "formFile": {
              "style": "form"
            }
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/Gtfs/{agencyId}": {
    "get": {
      "tags": [
        "Gtfs"
      ]
    }
  }
}
```

```
    ],
    "parameters": [
      {
        "name": "agencyId",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  },
  "/api/Gtfs/Organiser/{organiserId}": {
    "get": {
      "tags": [
        "Gtfs"
      ],
      "parameters": [
        {
          "name": "organiserId",
          "in": "path",
          "required": true,
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/Organiser": {
    "get": {
      "tags": [
        "Organiser"
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/OrganiserDTO"
              }
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
```

```
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        },
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    },
    "post": {
      "tags": [
        "Organiser"
      ],
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/OrganiserDTO"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/OrganiserDTO"
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/OrganiserDTO"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Success"
        },
        "400": {
          "description": "Bad Request",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        }
      }
    },
    "put": {
```



```
"tags": [
  "Organiser"
],
"requestBody": {
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/OrganiserDTO"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/OrganiserDTO"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/OrganiserDTO"
      }
    }
  }
},
"responses": {
  "200": {
    "description": "Success"
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
}
},
"/api/Gus/{nip}": {
  "get": {
    "tags": [
      "Organiser"
    ],
    "parameters": [
      {
        "name": "nip",
        "in": "path",
        "required": true,
        "schema": {
          "type": "string"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
```

```

        "$ref": "#/components/schemas/DanePodmiotu"
      }
    }
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "404": {
    "description": "Not Found",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
}
},
"/api/Organiser/{id}": {
  "delete": {
    "tags": [
      "Organiser"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
}
}

```

```
,
"/api/Route": {
  "get": {
    "tags": [
      "Route"
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/RouteDTO"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "post": {
    "tags": [
      "Route"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/RouteDTO"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/RouteDTO"
          }
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/RouteDTO"
        }
      }
    }
  }
}
```

```
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"put": {
  "tags": [
    "Route"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/RoutedDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/RoutedDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/RoutedDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Route/Agency": {
  "get": {
```

```
"tags": [
  "Route"
],
"parameters": [
  {
    "name": "agencyId",
    "in": "query",
    "schema": {
      "type": "integer",
      "format": "int32"
    }
  }
],
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/RouteViewDTO"
        }
      }
    }
  },
  "400": {
    "description": "Bad Request",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "404": {
    "description": "Not Found",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
},
"/api/Route/{id}": {
  "delete": {
    "tags": [
      "Route"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ]
  }
}
```

```

    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Stop/{id}": {
  "get": {
    "tags": [
      "Stop"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/StopDTO"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "404": {
    "description": "Not Found",
    "content": {

```

```

        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      },
    },
  },
  "delete": {
    "tags": [
      "Stop"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/Stop": {
    "get": {
      "tags": [
        "Stop"
      ],
      "parameters": [
        {
          "name": "NELatitude",
          "in": "query",
          "schema": {
            "type": "number",
            "format": "double"
          }
        },
        {
          "name": "NELongitude",
          "in": "query",
          "schema": {
            "type": "number",

```

```
        "format": "double"
      }
    },
    {
      "name": "SWLatitude",
      "in": "query",
      "schema": {
        "type": "number",
        "format": "double"
      }
    },
    {
      "name": "SWLongitude",
      "in": "query",
      "schema": {
        "type": "number",
        "format": "double"
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/StopDTO"
          }
        }
      }
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "404": {
      "description": "Not Found",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  },
  "post": {
    "tags": [
      "Stop"
    ],
    "requestBody": {
      "content": {
        "application/json": {
```



```

        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"put": {
  "tags": [
    "Stop"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/StopDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {

```

```
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
    },
  },
  "/api/StopTime": {
    "get": {
      "tags": [
        "StopTime"
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/StopTimeDTO"
              }
            }
          }
        },
        "400": {
          "description": "Bad Request",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        }
      }
    },
    "post": {
      "tags": [
        "StopTime"
      ],
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/StopTimeDTO"
            }
          }
        }
      }
    }
  }
}
```

```

    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/StopTimeDTO"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/StopTimeDTO"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"put": {
  "tags": [
    "StopTime"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/StopTimeDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/StopTimeDTO"
        }
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/StopTimeDTO"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {

```

```
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "/api/StopTime/Trip": {
    "get": {
      "tags": [
        "StopTime"
      ],
      "parameters": [
        {
          "name": "tripId",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/StopTimeViewDTO"
                }
              }
            }
          }
        },
        "400": {
          "description": "Bad Request",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        }
      }
    }
  }
}
```

```

},
"/api/StopTimes": {
  "put": {
    "tags": [
      "StopTime"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/StopTimeViewDTO"
            }
          }
        },
        "text/json": {
          "schema": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/StopTimeViewDTO"
            }
          }
        },
        "application/*+json": {
          "schema": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/StopTimeViewDTO"
            }
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
}

```

```
,
"/api/StopTime/{id}": {
  "delete": {
    "tags": [
      "StopTime"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ]
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Trip": {
  "get": {
    "tags": [
      "Trip"
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/TripDTO"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  }
},
```

```
"404": {
  "description": "Not Found",
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ProblemDetails"
      }
    }
  }
},
"post": {
  "tags": [
    "Trip"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"put": {
  "tags": [
    "Trip"
  ],
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      }
    }
  }
}
```

```

        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/TripDTO"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Trip/Route": {
  "get": {
    "tags": [
      "Trip"
    ],
    "parameters": [
      {
        "name": "routeId",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/TripViewDto"
            }
          }
        }
      }
    }
  },
  "400": {
    "description": "Bad Request",
    "content": {

```



```
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      },
      "404": {
        "description": "Not Found",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        }
      }
    }
  },
  "/api/Trip/{id}": {
    "delete": {
      "tags": [
        "Trip"
      ],
      "parameters": [
        {
          "name": "id",
          "in": "path",
          "required": true,
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        },
        "400": {
          "description": "Bad Request",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        }
      }
    }
  },
  "/api/Trip/Clone": {
    "post": {
      "tags": [
        "Trip"
      ],
      "requestBody": {
        "content": {
```

```
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/CloneTripDTO"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/CloneTripDTO"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/CloneTripDTO"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/Users/authenticate": {
  "post": {
    "tags": [
      "Users"
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/AuthenticateRequest"
          }
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/AuthenticateRequest"
        }
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/AuthenticateRequest"
      }
    }
  }
},
"responses": {
  "200": {
```

```
        "description": "Success"
      }
    }
  },
  "/api/Users/refresh-token": {
    "post": {
      "tags": [
        "Users"
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/Users/revoke-token": {
    "post": {
      "tags": [
        "Users"
      ],
      "requestBody": {
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/RevokeTokenRequest"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/RevokeTokenRequest"
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/RevokeTokenRequest"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/Users": {
    "get": {
      "tags": [
        "Users"
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  }
},
```

```
"/api/Users/{id}": {
  "get": {
    "tags": [
      "Users"
    ],
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  },
  "/api/Users/{id}/refresh-tokens": {
    "get": {
      "tags": [
        "Users"
      ],
      "parameters": [
        {
          "name": "id",
          "in": "path",
          "required": true,
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "components": {
    "schemas": {
      "AgencyDTO": {
        "type": "object",
        "properties": {
          "id": {
            "type": "integer",
            "format": "int32"
          },
          "city": {
            "type": "string",
            "nullable": true
          }
        }
      }
    }
  }
}
```

```
"district": {
  "type": "string",
  "nullable": true
},
"email": {
  "type": "string",
  "nullable": true
},
"idExternal": {
  "type": "string",
  "nullable": true
},
"municipality": {
  "type": "string",
  "nullable": true
},
"name": {
  "type": "string",
  "nullable": true
},
"phone": {
  "type": "string",
  "nullable": true
},
"postalCode": {
  "type": "string",
  "nullable": true
},
"primaryNumber": {
  "type": "string",
  "nullable": true
},
"secondaryNumber": {
  "type": "string",
  "nullable": true
},
"streetName": {
  "type": "string",
  "nullable": true
},
"type": {
  "type": "string",
  "nullable": true
},
"url": {
  "type": "string",
  "nullable": true
},
"idFeed": {
  "type": "integer",
  "format": "int32"
},
"urlFare": {
  "type": "string",
  "nullable": true
},
"organiserId": {
  "type": "integer",
  "format": "int32",
  "nullable": true
}
```

```
    },
    "nip": {
      "type": "string",
      "nullable": true
    },
    "serviceNotes": {
      "type": "string",
      "nullable": true
    },
    "gtfsSourceUrl": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": false
},
"AuthenticateRequest": {
  "required": [
    "password",
    "username"
  ],
  "type": "object",
  "properties": {
    "username": {
      "type": "string"
    },
    "password": {
      "type": "string"
    }
  },
  "additionalProperties": false
},
"CalendarDTO": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "name": {
      "type": "string",
      "nullable": true
    },
    "symbol": {
      "type": "string",
      "nullable": true
    },
    "starting": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    },
    "ending": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    },
    "exclusion": {
      "type": "boolean"
    }
  }
}
```

```
    },
    "autogenerated": {
      "type": "boolean"
    }
  },
  "additionalProperties": false
},
"CalendarDateDTO": {
  "type": "object",
  "properties": {
    "date": {
      "type": "array",
      "items": {
        "type": "string",
        "format": "date-time"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"CalendarWithDatesDTO": {
  "type": "object",
  "properties": {
    "calendar": {
      "$ref": "#/components/schemas/CalendarDTO"
    },
    "dates": {
      "type": "array",
      "items": {
        "type": "string",
        "format": "date-time"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"CalendarsByTripDto": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "format": "int32"
    },
    "exclusion": {
      "type": "boolean"
    },
    "starting": {
      "type": "string",
      "format": "date-time"
    },
    "ending": {
      "type": "string",
      "format": "date-time"
    },
    "name": {
      "type": "string",
      "nullable": true
    }
  },
}
```

```
        "symbol": {
            "type": "string",
            "nullable": true
        }
    },
    "additionalProperties": false
},
"CloneTripDTO": {
    "type": "object",
    "properties": {
        "tripId": {
            "type": "integer",
            "format": "int32"
        },
        "clonesNumber": {
            "type": "integer",
            "format": "int32"
        },
        "minutesInterval": {
            "type": "integer",
            "format": "int32"
        }
    }
},
"additionalProperties": false
},
"CommunicationExclusionDTO": {
    "type": "object",
    "properties": {
        "xPos": {
            "type": "number",
            "format": "float"
        },
        "ypos": {
            "type": "number",
            "format": "float"
        },
        "teryt": {
            "type": "string",
            "nullable": true
        },
        "simcId": {
            "type": "string",
            "nullable": true
        },
        "ulicaId": {
            "type": "string",
            "nullable": true
        },
        "miejscowosc": {
            "type": "string",
            "nullable": true
        },
        "ulica": {
            "type": "string",
            "nullable": true
        },
        "prngId": {
            "type": "string",
            "nullable": true
        }
    },
    "additionalProperties": false
}
```



```
"gminaId": {
  "type": "string",
  "nullable": true
},
"communicationExclusionValue": {
  "type": "number",
  "format": "float"
},
"availability": {
  "type": "string",
  "nullable": true
},
"tripTime": {
  "type": "string",
  "nullable": true
},
"cost": {
  "type": "string",
  "nullable": true
},
"frequency": {
  "type": "string",
  "nullable": true
},
"reliability": {
  "type": "string",
  "nullable": true
},
"handicap": {
  "type": "string",
  "nullable": true
},
"type": {
  "type": "string",
  "nullable": true
},
"number": {
  "type": "string",
  "nullable": true
}
},
"additionalProperties": false
},
"DanePodmiotu": {
  "type": "object",
  "properties": {
    "regon": {
      "type": "string",
      "nullable": true
    },
    "nip": {
      "type": "string",
      "nullable": true
    },
    "statusNip": {
      "type": "string",
      "nullable": true
    },
    "nazwa": {
      "type": "string",
```

```
    "nullable": true
  },
  "wojewodztwo": {
    "type": "string",
    "nullable": true
  },
  "powiat": {
    "type": "string",
    "nullable": true
  },
  "gmina": {
    "type": "string",
    "nullable": true
  },
  "miejscowosc": {
    "type": "string",
    "nullable": true
  },
  "kodPocztowy": {
    "type": "string",
    "nullable": true
  },
  "ulica": {
    "type": "string",
    "nullable": true
  },
  "nrNieruchomosci": {
    "type": "string",
    "nullable": true
  },
  "nrLokalu": {
    "type": "string",
    "nullable": true
  },
  "typ": {
    "type": "string",
    "nullable": true
  },
  "silosID": {
    "type": "integer",
    "format": "int32"
  },
  "dataZakonczeniaDzialalnosci": {
    "type": "string",
    "nullable": true
  },
  "miejscowoscPoczty": {
    "type": "string",
    "nullable": true
  },
  "errors": {
    "type": "array",
    "items": {
      "$ref": "#/components/schemas/ErrorMessage"
    },
    "nullable": true
  }
},
"additionalProperties": false
},
```

```
"ErrorModel": {
  "type": "object",
  "properties": {
    "errorCode": {
      "type": "string",
      "nullable": true
    },
    "errorMessagePl": {
      "type": "string",
      "nullable": true
    },
    "errorMessageEn": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": false
},
"FareAttributesDTO": {
  "type": "object",
  "properties": {
    "fareIdExternal": {
      "type": "string",
      "nullable": true
    },
    "fareId": {
      "type": "integer",
      "format": "int32"
    },
    "price": {
      "type": "number",
      "format": "float"
    },
    "currencyType": {
      "type": "string",
      "nullable": true
    },
    "paymentMethod": {
      "type": "string",
      "nullable": true
    },
    "transfers": {
      "type": "string",
      "nullable": true
    },
    "agencyId": {
      "type": "integer",
      "format": "int32"
    },
    "transferDuration": {
      "type": "integer",
      "format": "int32"
    }
  },
  "additionalProperties": false
},
"File": {
  "type": "object",
  "additionalProperties": false
},
```

```
"OrganiserDTO": {
  "type": "object",
  "properties": {
    "organiserId": {
      "type": "integer",
      "format": "int32"
    },
    "organiserName": {
      "type": "string",
      "nullable": true
    },
    "voivodeship": {
      "type": "string",
      "nullable": true
    },
    "county": {
      "type": "string",
      "nullable": true
    },
    "type": {
      "type": "string",
      "nullable": true
    },
    "nameOfLocalGovernmentUnit": {
      "type": "string",
      "nullable": true
    },
    "nameOfOffice": {
      "type": "string",
      "nullable": true
    },
    "city": {
      "type": "string",
      "nullable": true
    },
    "postalCode": {
      "type": "string",
      "nullable": true
    },
    "postOffice": {
      "type": "string",
      "nullable": true
    },
    "streetName": {
      "type": "string",
      "nullable": true
    },
    "streetNumber": {
      "type": "string",
      "nullable": true
    },
    "telephonePrefix": {
      "type": "string",
      "nullable": true
    },
    "telephoneNumber": {
      "type": "string",
      "nullable": true
    },
    "telephoneNumberAdditional": {
```

```
        "type": "string",
        "nullable": true
    },
    "telephoneNumberInternal": {
        "type": "string",
        "nullable": true
    },
    "faxNumberPrefix": {
        "type": "string",
        "nullable": true
    },
    "faxNumber": {
        "type": "string",
        "nullable": true
    },
    "faxNumberInternal": {
        "type": "string",
        "nullable": true
    },
    "emailGeneral": {
        "type": "string",
        "nullable": true
    },
    "emailContact": {
        "type": "string",
        "nullable": true
    },
    "websiteUrl": {
        "type": "string",
        "nullable": true
    },
    "esp": {
        "type": "string",
        "nullable": true
    },
    "nip": {
        "type": "string",
        "nullable": true
    },
    "serviceNotes": {
        "type": "string",
        "nullable": true
    },
    "gtfsSourceUrl": {
        "type": "string",
        "nullable": true
    }
},
"additionalProperties": false
},
"ProblemDetails": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "nullable": true
        },
        "title": {
            "type": "string",
            "nullable": true
        }
    }
}
```

```
    },
    "status": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "detail": {
      "type": "string",
      "nullable": true
    },
    "instance": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": { }
},
"RevokeTokenRequest": {
  "type": "object",
  "properties": {
    "token": {
      "type": "string",
      "nullable": true
    }
  }
},
"additionalProperties": false
},
"RoutedTO": {
  "type": "object",
  "properties": {
    "routeId": {
      "type": "integer",
      "format": "int32"
    },
    "agencyId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "continuousDropoff": {
      "type": "string",
      "nullable": true
    },
    "continuousPickup": {
      "type": "string",
      "nullable": true
    },
    "routeDesc": {
      "type": "string",
      "nullable": true
    },
    "routeExternalId": {
      "type": "string",
      "nullable": true
    },
    "routeLongName": {
      "type": "string",
      "nullable": true
    },
    "routeShortName": {
```

```
        "type": "string",
        "nullable": true
    },
    "routeType": {
        "type": "string",
        "nullable": true
    },
    "routeUrl": {
        "type": "string",
        "nullable": true
    },
    "routeColor": {
        "type": "string",
        "nullable": true
    },
    "routeTextColor": {
        "type": "string",
        "nullable": true
    },
    "permissionNo": {
        "type": "string",
        "nullable": true
    },
    "timetableNotes": {
        "type": "string",
        "nullable": true
    },
    "suspended": {
        "type": "boolean",
        "nullable": true
    },
    "specialRegularService": {
        "type": "boolean",
        "nullable": true
    }
},
"additionalProperties": false
},
"RouteViewDTO": {
    "type": "object",
    "properties": {
        "routeId": {
            "type": "integer",
            "format": "int32"
        },
        "agencyId": {
            "type": "integer",
            "format": "int32",
            "nullable": true
        },
        "continuousDropoff": {
            "type": "string",
            "nullable": true
        },
        "continuousPickup": {
            "type": "string",
            "nullable": true
        },
        "routeDesc": {
            "type": "string",
```

```
        "nullable": true
      },
      "routeExternalId": {
        "type": "string",
        "nullable": true
      },
      "routeLongName": {
        "type": "string",
        "nullable": true
      },
      "routeShortName": {
        "type": "string",
        "nullable": true
      },
      "routeType": {
        "type": "string",
        "nullable": true
      },
      "routeUrl": {
        "type": "string",
        "nullable": true
      },
      "routeColor": {
        "type": "string",
        "nullable": true
      },
      "routeTextColor": {
        "type": "string",
        "nullable": true
      },
      "permissionNo": {
        "type": "string",
        "nullable": true
      },
      "timetableNotes": {
        "type": "string",
        "nullable": true
      },
      "suspended": {
        "type": "boolean",
        "nullable": true
      },
      "specialRegularService": {
        "type": "boolean",
        "nullable": true
      },
      "routeStartDate": {
        "type": "string",
        "format": "date-time",
        "nullable": true
      },
      "routeEndDate": {
        "type": "string",
        "format": "date-time",
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "StopDTO": {
```



```
"type": "object",
"properties": {
  "id": {
    "type": "integer",
    "format": "int32"
  },
  "city": {
    "type": "string",
    "nullable": true
  },
  "code": {
    "type": "string",
    "nullable": true
  },
  "district": {
    "type": "string",
    "nullable": true
  },
  "idExternal": {
    "type": "string",
    "nullable": true
  },
  "latitude": {
    "type": "number",
    "format": "double",
    "nullable": true
  },
  "longitude": {
    "type": "number",
    "format": "double",
    "nullable": true
  },
  "municipality": {
    "type": "string",
    "nullable": true
  },
  "name": {
    "type": "string",
    "nullable": true
  },
  "postalCode": {
    "type": "string",
    "nullable": true
  },
  "primaryNumber": {
    "type": "string",
    "nullable": true
  },
  "secondaryNumber": {
    "type": "string",
    "nullable": true
  },
  "streetName": {
    "type": "string",
    "nullable": true
  },
  "wheelchairSupport": {
    "type": "string",
    "nullable": true
  },
}
```

```
"locationType": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"parentStation": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"platformCode": {
  "type": "string",
  "nullable": true
},
"levelId": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"accessibilityId": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"ttsStopName": {
  "type": "string",
  "nullable": true
},
"shelter": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"bench": {
  "type": "integer",
  "format": "int32",
  "nullable": true
},
"manager": {
  "type": "string",
  "nullable": true
},
"stopDescription": {
  "type": "string",
  "nullable": true
},
"stopUrl": {
  "type": "string",
  "nullable": true
},
"landmarks": {
  "type": "string",
  "nullable": true
},
"inUse": {
  "type": "boolean",
  "nullable": true
}
},
"additionalProperties": false
```

```
},
"StopTimeDTO": {
  "type": "object",
  "properties": {
    "stopTimesId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "arrivalTime": {
      "$ref": "#/components/schemas/TimeSpan"
    },
    "continuousPickup": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "continuousDropoff": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "departureTime": {
      "$ref": "#/components/schemas/TimeSpan"
    },
    "dropoffType": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "pickupType": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "stopHeadsign": {
      "type": "string",
      "nullable": true
    },
    "timetableNotes": {
      "type": "string",
      "nullable": true
    },
    "stopId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "stopSequence": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "tripId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    }
  },
  "additionalProperties": false
}
```

```
},
"StopTimeViewDTO": {
  "type": "object",
  "properties": {
    "stopTimesId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "arrivalTime": {
      "$ref": "#/components/schemas/TimeSpan"
    },
    "continuousPickup": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "continuousDropoff": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "departureTime": {
      "$ref": "#/components/schemas/TimeSpan"
    },
    "dropoffType": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "pickupType": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "stopHeadsign": {
      "type": "string",
      "nullable": true
    },
    "timetableNotes": {
      "type": "string",
      "nullable": true
    },
    "stopId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "stopSequence": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "tripId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "wheelchairSupport": {
      "type": "string",
```

```
        "nullable": true
    },
    "stopLatitude": {
        "type": "number",
        "format": "double",
        "nullable": true
    },
    "stopLongitude": {
        "type": "number",
        "format": "double",
        "nullable": true
    },
    "stopCode": {
        "type": "string",
        "nullable": true
    },
    "stopName": {
        "type": "string",
        "nullable": true
    }
},
"additionalProperties": false
},
"TimeSpan": {
    "type": "object",
    "properties": {
        "ticks": {
            "type": "integer",
            "format": "int64"
        },
        "days": {
            "type": "integer",
            "format": "int32",
            "readOnly": true
        },
        "hours": {
            "type": "integer",
            "format": "int32",
            "readOnly": true
        },
        "milliseconds": {
            "type": "integer",
            "format": "int32",
            "readOnly": true
        },
        "minutes": {
            "type": "integer",
            "format": "int32",
            "readOnly": true
        },
        "seconds": {
            "type": "integer",
            "format": "int32",
            "readOnly": true
        },
        "totalDays": {
            "type": "number",
            "format": "double",
            "readOnly": true
        }
    },
    "additionalProperties": false
}
```

```
"totalHours": {
  "type": "number",
  "format": "double",
  "readOnly": true
},
"totalMilliseconds": {
  "type": "number",
  "format": "double",
  "readOnly": true
},
"totalMinutes": {
  "type": "number",
  "format": "double",
  "readOnly": true
},
"totalSeconds": {
  "type": "number",
  "format": "double",
  "readOnly": true
}
},
"additionalProperties": false
},
"TripDTO": {
  "type": "object",
  "properties": {
    "tripId": {
      "type": "integer",
      "format": "int32"
    },
    "bikesAllowed": {
      "type": "boolean"
    },
    "calendarDateId": {
      "type": "integer",
      "format": "int32"
    },
    "directionId": {
      "type": "boolean"
    },
    "routeId": {
      "type": "integer",
      "format": "int32"
    },
    "serviceId": {
      "type": "integer",
      "format": "int32"
    },
    "predefinedCalendarId": {
      "type": "integer",
      "format": "int32"
    },
    "tripHeadsign": {
      "type": "string",
      "nullable": true
    },
    "tripIdExternal": {
      "type": "string",
      "nullable": true
    }
  },
}
```

```
"tripShortName": {
  "type": "string",
  "nullable": true
},
"blockId": {
  "type": "string",
  "nullable": true
},
"timetableNotes": {
  "type": "string",
  "nullable": true
},
"wheelchairAccessible": {
  "type": "boolean"
},
"suspended": {
  "type": "boolean"
},
"selectedCalendars": {
  "type": "array",
  "items": {
    "$ref": "#/components/schemas/CalendarDTO"
  },
  "nullable": true
}
},
"additionalProperties": false
},
"TripViewDto": {
  "type": "object",
  "properties": {
    "tripId": {
      "type": "integer",
      "format": "int32"
    },
    "bikesAllowed": {
      "type": "boolean"
    },
    "calendarDateId": {
      "type": "integer",
      "format": "int32"
    },
    "directionId": {
      "type": "boolean"
    },
    "routeId": {
      "type": "integer",
      "format": "int32"
    },
    "serviceId": {
      "type": "integer",
      "format": "int32"
    },
    "predefinedCalendarId": {
      "type": "integer",
      "format": "int32"
    },
    "tripHeadsign": {
      "type": "string",
      "nullable": true
    }
  }
}
```

```

    },
    "tripIdExternal": {
      "type": "string",
      "nullable": true
    },
    "tripShortName": {
      "type": "string",
      "nullable": true
    },
    "blockId": {
      "type": "string",
      "nullable": true
    },
    "timetableNotes": {
      "type": "string",
      "nullable": true
    },
    "wheelchairAccessible": {
      "type": "boolean"
    },
    "suspended": {
      "type": "boolean"
    },
    "selectedCalendars": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/CalendarDTO"
      },
      "nullable": true
    },
    "weekdays": {
      "type": "string",
      "nullable": true
    },
    "startDate": {
      "type": "string",
      "nullable": true
    },
    "endDate": {
      "type": "string",
      "nullable": true
    },
    "firstArrival": {
      "type": "string",
      "nullable": true
    },
    "lastArrival": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": false
},
"securitySchemes": {
  "Bearer": {
    "type": "apiKey",
    "description": "JWT Authorization header using the Bearer scheme. \r\n
Enter 'Bearer' [space] and then your token in the text input below.\r\n
Example: 'Bearer 12345abcdef'",

```



```
        "name": "Authorization",
        "in": "header"
      }
    },
    "security": [
      {
        "Bearer": [ ]
      }
    ]
  }
}
```