

# Projekt – web serwisy

## System Rezerwacji Biletów Lotniczych

---

Rozproszone systemy internetowe – Politechnika Białostocka

**Autorzy:**

Bartosz Ciereszyński

Piotr Zajączkowski

*Białystok, 29.04.2024 r.*

# 1 WPROWADZENIE

## 1.1 CEL PROJEKTU

Celem projektu jest utworzenie web serwisu oraz aplikacji klienckiej korzystającej z tego web serwisu.

## 1.2 TEMAT PROJEKTU I WYMAGANIA

Tematem projektu jest *System Rezerwacji Biletów Lotniczych*, w ramach którego spełnione zostały muszą poniższe wymagania funkcjonalne:

- *Baza lotów (Miasto od , Miasto do, dzień, godzina)*
- *Wyszukiwanie lotów*
- *Kupno biletu,*
- *Odbiór potwierdzenia kupna w formacie PDF*
- *Sprawdzenie rezerwacji na podstawie podanego numeru*

Spełniono także dodatkowe wymagania нефункционалне takie jak:

- *Użycie Handlers*
- *Użycie przesyłania załączników binarnych (@MTOM)*
- *Szyfrowanie SSL/TLS dla jednego z serwisów*
- *Klient napisany w innym języku niż Java (Python)*

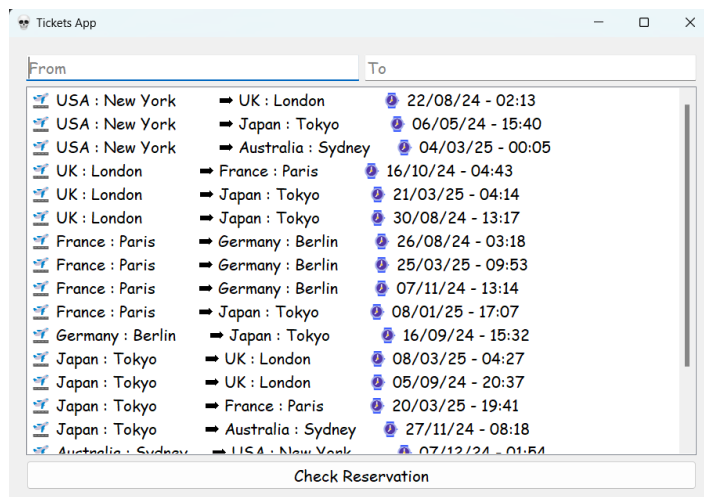
## 1.3 HANDLERS

Handlerzy wykorzystano jako zabezpieczenie w postaci klucza API. Serwer ma zapisane klucze, które porównuje z kluczem podanym w soap headerze. W przypadku braku podania klucza lub braku odpowiadającego mu klucza w serwerze zostanie wyrzucony wyjątek.

Przykładowy klucz API: **bc759280-b4b5-43bb-8b79-5cf089dffb8**

## 1.4 KLIENT

Klient został napisany w języku python (wersja 3.12) z wykorzystaniem biblioteki *PyQt5* do implementacji interfejsu graficznego oraz biblioteki *ZEEP* do połączenia z serwisem.



## 2 SPECYFIKACJA WEB SERVICE

---

### 2.1 TECHNOLOGIE

System rezerwacji biletów lotniczych został zaimplementowany jako usługa sieciowa (Web Service) z użyciem protokołu SOAP w wersji 1.1.

System dzieli się na dwa serwisy:

- FlightsService – rezerwacja i pdf (SSL)
- FlightsInfo – pozyskiwanie informacji

Serwer wykorzystywany do hosting do GlassFish 4.1.2, a kod jest napisany w Java 8

### 2.2 ADRES USŁUGI

- [https://\"SERWER-IP\":8181/ticketsWebService-1.0/FlightsServiceService](https://\)
- [https://\"SERWER-IP\":8080/ticketsWebService-1.0/FlightsInfoService](https://\)

### 2.3 SPECYFIKACJA WSDL

#### 2.3.1 FlightsService

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://webservices.ticketswebservice.rsi/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://webservices.ticketswebservice.rsi/" name="FlightsService">
  <wsp:Policy wsu:Id="FlightsServicePortBinding_MTOM_Policy-FlightsServicePortBinding_MTOM_Policy">
    <ns1:OptimizedMimeSerialization
      xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
      wsp:Optional="true" />
  </wsp:Policy>
  <types>
    <xsd:schema>
      <xsd:import namespace="http://webservices.ticketswebservice.rsi/"
        schemaLocation="https://localhost:8181/ticketsWebService-1.0/FlightsServiceService?xsd=1" />
    </xsd:schema>
  </types>
  <message name="bookFlight">
    <part name="parameters" element="tns:bookFlight" />
  </message>
  <message name="bookFlightResponse">
    <part name="parameters" element="tns:bookFlightResponse" />
  </message>
  <message name="getBookingConfirmation">
    <part name="parameters" element="tns:getBookingConfirmation" />
  </message>
  <message name="getBookingConfirmationResponse">
    <part name="parameters" element="tns:getBookingConfirmationResponse" />
  </message>
  <message name="IOException">
    <part name="fault" element="tns:IOException" />
  </message>
  <portType name="FlightsService">
    <operation name="bookFlight">
      <input>
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsService/bookFlightRequest"
        message="tns:bookFlight" />
      <output>
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsService/bookFlightResponse"
        message="tns:bookFlightResponse" />
      </operation>
    <operation name="getBookingConfirmation">
      <input>
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsService/getBookingConfirmationRequest"
```

```

        message="tns:getBookingConfirmation" />
    </output>
    wsam:Action="http://webservices.ticketswebservice.rsi/FlightsService/getBookingConfirmationResponse"
    message="tns:getBookingConfirmationResponse" />
    <fault message="tns:IOException" name="IOException"
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsService/getBookingConfirmation/Fault/IOE
xception" />
    </operation>
</portType>
<binding name="FlightsServicePortBinding" type="tns:FlightsService">
    <wsp:PolicyReference
        URI="#FlightsServicePortBinding_MTOM_Policy-FlightsServicePortBinding_MTOM_Policy" />
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="bookFlight">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getBookingConfirmation">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="IOException">
            <soap:fault name="IOException" use="literal" />
        </fault>
    </operation>
</binding>
<service name="FlightsServiceService">
    <port name="FlightsServicePort" binding="tns:FlightsServicePortBinding">
        <soap:address
            Location="https://localhost:8181/ticketsWebService-1.0/FlightsServiceService" />
    </port>
</service>
</definitions>

```

## 2.3.2 FlightsInfo

```

<?xml version='1.0' encoding='UTF-8'?>
<definitions
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://webservices.ticketswebservice.rsi/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://webservices.ticketswebservice.rsi/" name="FlightsInfoService">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://webservices.ticketswebservice.rsi/"
                schemaLocation="http://localhost:8080/ticketsWebService-1.0/FlightsInfoService?xsd=1" />
        </xsd:schema>
    </types>
    <message name="getFlightsFrom">
        <part name="parameters" element="tns:getFlightsFrom" />
    </message>
    <message name="getFlightsFromResponse">
        <part name="parameters" element="tns:getFlightsFromResponse" />
    </message>
    <message name="getFlights">
        <part name="parameters" element="tns:getFlights" />
    </message>
    <message name="getFlightsResponse">
        <part name="parameters" element="tns:getFlightsResponse" />
    </message>
    <message name="getFlightsTo">
        <part name="parameters" element="tns:getFlightsTo" />
    </message>
    <message name="getFlightsToResponse">
        <part name="parameters" element="tns:getFlightsToResponse" />
    </message>
    <message name="getCities">
        <part name="parameters" element="tns:getCities" />
    </message>
    <message name="getCitiesResponse">
        <part name="parameters" element="tns:getCitiesResponse" />
    </message>

```

```

<message name="getTickets">
  <part name="parameters" element="tns:getTickets" />
</message>
<message name="getTicketsResponse">
  <part name="parameters" element="tns:getTicketsResponse" />
</message>
<message name="getTicket">
  <part name="parameters" element="tns:getTicket" />
</message>
<message name="getTicketResponse">
  <part name="parameters" element="tns:getTicketResponse" />
</message>
<message name="getFlightsBetween">
  <part name="parameters" element="tns:getFlightsBetween" />
</message>
<message name="getFlightsBetweenResponse">
  <part name="parameters" element="tns:getFlightsBetweenResponse" />
</message>
<portType name="FlightsInfo">
  <operation name="getFlightsFrom">
    <input
      wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsFromRequest"
      message="tns:getFlightsFrom" />
    <output
      wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsFromResponse"
      message="tns:getFlightsFromResponse" />
    </operation>
    <operation name="getFlights">
      <input
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsRequest"
        message="tns:getFlights" />
      <output
        wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsResponse"
        message="tns:getFlightsResponse" />
      </operation>
      <operation name="getFlightsTo">
        <input
          wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsToRequest"
          message="tns:getFlightsTo" />
        <output
          wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsToResponse"
          message="tns:getFlightsToResponse" />
        </operation>
        <operation name="getCities">
          <input
            wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getCitiesRequest"
            message="tns:getCities" />
          <output
            wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getCitiesResponse"
            message="tns:getCitiesResponse" />
          </operation>
          <operation name="getTickets">
            <input
              wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getTicketsRequest"
              message="tns:getTickets" />
            <output
              wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getTicketsResponse"
              message="tns:getTicketsResponse" />
            </operation>
            <operation name="getTicket">
              <input
                wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getTicketRequest"
                message="tns:getTicket" />
              <output
                wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getTicketResponse"
                message="tns:getTicketResponse" />
              </operation>
              <operation name="getFlightsBetween">
                <input
                  wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsBetweenRequest"
                  message="tns:getFlightsBetween" />
                <output
                  wsam:Action="http://webservices.ticketswebservice.rsi/FlightsInfo/getFlightsBetweenResponse"
                  message="tns:getFlightsBetweenResponse" />
                </operation>
              </portType>
            <binding name="FlightsInfoPortBinding" type="tns:FlightsInfo">
              <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
              <operation name="getFlightsFrom">
                <soap:operation soapAction="" />
                <input>
                  <soap:body use="literal" />
                </input>
                <output>
                  <soap:body use="literal" />
                </output>
              </binding>
            </portType>
          </binding>
        </portType>
      </binding>
    </portType>
  </binding>
</portType>

```

```

        </output>
    </operation>
    <operation name="getFlights">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getFlightsTo">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getCities">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getTickets">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getTicket">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="getFlightsBetween">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
</binding>
<service name="FlightsInfoService">
    <port name="FlightsInfoPort" binding="tns:FlightsInfoPortBinding">
        <soap:address location="http://localhost:8080/ticketsWebService-1.0/FlightsInfoService" />
    </port>
</service>
</definitions>

```

## 2.4 OPERACJE USŁUGI FLIGHTS SERVICE

### 2.4.1 Opracje serwisu FlightsService

Możliwe operacje w serwisie *FlightsService*

- *BookFlight*
- *GetBookingConfirmation*

### 2.4.2 Operacja BookFlight

#### 2.4.2.1 Opis ogólny

Opis ogólny Operacja *BookFlight* służy do rezerwacji lotu dla pasażera o podanym imieniu i nazwisku. Operacja korzysta z aktualnego stanu danych, tj. dostępności lotu w momencie wystania zapytania.

#### 2.4.2.2 Opis szczegółowy

Dane wejściowe:

- *flightId* - Numer identyfikacyjny lotu - składający się z alfanumerycznych znaków, bez białych znaków i znaków specjalnych.
- *passengerFirstName* - Imię pasażera - ciąg znaków.
- *passengerSurname* - Nazwisko pasażera - ciąg znaków.

Dane wyjściowe:

- Wynik operacji:
  - *TicketDto* - obiekt przekazujący dane reprezentujący szczegóły biletu. Zawiera ID biletu, ID lotu, imię pasażera i nazwisko pasażera.

### 2.4.3 Operacja GetBookingConfirmation

#### 2.4.3.1 Opis ogólny

Operacja *GetBookingConfirmation* służy do pobrania potwierdzenia rezerwacji dla biletu o podanym ID biletu. Operacja korzysta z aktualnego stanu danych, tj. statusu biletu w momencie wystania zapytania.

#### 2.4.3.2 Opis szczegółowy

Dane wejściowe:

- *ticketId* - Numer identyfikacyjny biletu - składający się z alfanumerycznych znaków, bez białych znaków i znaków specjalnych.

Dane wyjściowe:

- Wynik operacji:
  - *byte[]* - tablica bajtów reprezentująca dokument potwierdzający rezerwację. Ten dokument można przekształcić na PDF do wyświetlania lub drukowania. Jeśli ID biletu jest nieprawidłowe lub jeśli wystąpi błąd podczas generowania dokumentu, zostanie zgłoszony wyjątek *IOException*.

## 2.5 OPERACJE USŁUGI FLIGHTS INFO

### 2.5.1 Opracje serwisu FlightsInfo

Możliwe operacje w serwisie *FlightsInfo*

- *GetFlights*
- *GetFlightsFrom*
- *GetFlightsTo*
- *GetFlightsBetween*
- *GetCities*
- *GetTickets*
- *GetTicket*

### 2.5.2 Operacja GetFlights

#### 2.5.2.1 Opis ogólny

Operacja *getFlights* służy do pobierania listy dostępnych lotów. Operacja korzysta z aktualnego stanu danych, tj. dostępności lotów w momencie wystania zapytania.

#### 2.5.2.2 Opis szczegółowy

Dane wejściowe:

- Brak

Dane wyjściowe:

- Wynik operacji:
  - Lista obiektów *FlightDto* - lista obiektów reprezentujących dostępne loty.

### 2.5.3 Operacja GetFlightsFrom

#### 2.5.3.1 Opis ogólny

Operacja *getFlightsFrom* służy do pobierania listy lotów z określonego miasta. Operacja korzysta z aktualnego stanu danych, tj. dostępności lotów w momencie wystania zapytania.

#### 2.5.3.2 Opis szczegółowy

Dane wejściowe:

- *query* - ciąg znaków reprezentujący miasto, z którego odlatują loty.

Dane wyjściowe:

- Wynik operacji:
  - Lista obiektów *FlightDto* - lista obiektów reprezentujących dostępne loty z określonego miasta.



## 2.5.4 Operacja GetFlightsTo

### 2.5.4.1 Opis ogólny

Operacja *getFlightsTo* służy do pobierania listy lotów do określonego miasta. Operacja korzysta z aktualnego stanu danych, tj. dostępności lotów w momencie wystania zapytania.

### 2.5.4.2 Opis szczegółowy

Dane wejściowe:

- *query* - ciąg znaków reprezentujący miasto, do którego przylatują loty.

Dane wyjściowe:

- Wynik operacji:
  - *Lista obiektów FlightDto* - lista obiektów reprezentujących dostępne loty do określonego miasta.

## 2.5.5 Operacja GetFlightsBetween

### 2.5.5.1 Opis ogólny

Operacja *getFlightsBetween* służy do pobierania listy lotów między dwoma miastami. Operacja korzysta z aktualnego stanu danych, tj. dostępności lotów w momencie wystania zapytania.

### 2.5.5.2 Opis szczegółowy

Dane wejściowe:

- *queryFrom* - ciąg znaków reprezentujący miasto, z którego odlatują loty.
- *queryTo* - ciąg znaków reprezentujący miasto, do którego przylatują loty.

Dane wyjściowe:

- Wynik operacji:
  - *Lista obiektów FlightDto* - lista obiektów reprezentujących dostępne loty między dwoma miastami.

## 2.5.6 Operacja GetCities

### 2.5.6.1 Opis ogólny

Operacja *getCities* służy do pobierania listy dostępnych miast. Operacja korzysta z aktualnego stanu danych.

### 2.5.6.2 Opis szczegółowy

Dane wejściowe:

- Brak

Dane wyjściowe:

- Wynik operacji:
  - *Lista obiektów CityDto* - lista obiektów reprezentujących dostępne miasta.

## 2.5.7 Operacja GetTickets

### 2.5.7.1 Opis ogólny

Operacja `getTickets` służy do pobierania listy już zarezerwowanych biletów. Operacja korzysta z aktualnego stanu danych.

### 2.5.7.2 Opis szczegółowy

Dane wejściowe:

- Brak

Dane wyjściowe:

- Wynik operacji:
  - *Lista obiektów TicketDto* - lista obiektów reprezentujących dostępne bilety.

## 2.5.8 Operacja GetTicket

### 2.5.8.1 Opis ogólny

Operacja `getTicket` służy do pobierania szczegółów biletu o podanym ID. Operacja korzysta z aktualnego stanu danych, tj. warunku istnienia biletu w momencie wystąpienia zapytania.

### 2.5.8.2 Opis szczegółowy

Dane wejściowe:

- *ticketId* - Numer identyfikacyjny biletu - składający się z alfanumerycznych znaków, bez białych znaków i znaków specjalnych.

Dane wyjściowe:

- Wynik operacji:
  - *TicketDto* - obiekt reprezentujący szczegóły biletu.

## 2.6 PRZYKŁADY UŻYCIA WYBRANYCH OPERACJI

### 2.6.1 Przykład użycia GetFlightsFrom

Komunikat wysłany

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <S:Body>
    <ns2:getFlightsFrom xmlns:ns2="http://webservices.ticketswebsevice.rsi/">
      <arg0>lond</arg0>
    </ns2:getFlightsFrom>
  </S:Body>
</S:Envelope>
```

Komunikat odebrany

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <S:Body>
    <ns2:getFlightsFromResponse xmlns:ns2="http://webservices.ticketswebsevice.rsi/">
      <return>
```

```

    <arrivalCity>
      <country>France</country>
      <id>f3d14711-bd44-45dc-bbf2-6a89513354cd</id>
      <name>Paris</name>
    </arrivalCity>
    <availableSeats>30</availableSeats>
    <departureCity>
      <country>UK</country>
      <id>dbfe42ad-bbe4-42ec-b90b-034fbc049085</id>
      <name>London</name>
    </departureCity>
    <departureTime>2024-08-08T23:39:45.624</departureTime>
    <id>64acf789-956a-414b-a7bc-a352ea0db7f6</id>
  </return>
</return>
  <arrivalCity>
    <country>Japan</country>
    <id>d9b19a4b-a25f-4c2a-b235-342e3a47e187</id>
    <name>Tokyo</name>
  </arrivalCity>
  <availableSeats>20</availableSeats>
  <departureCity>
    <country>UK</country>
    <id>dbfe42ad-bbe4-42ec-b90b-034fbc049085</id>
    <name>London</name>
  </departureCity>
  <departureTime>2024-11-03T22:09:45.624</departureTime>
  <id>7dd87c74-a045-4c8e-a505-0f16fa2ca64b</id>
</return>
</return>
  <arrivalCity>
    <country>Japan</country>
    <id>d9b19a4b-a25f-4c2a-b235-342e3a47e187</id>
    <name>Tokyo</name>
  </arrivalCity>
  <availableSeats>15</availableSeats>
  <departureCity>
    <country>UK</country>
    <id>dbfe42ad-bbe4-42ec-b90b-034fbc049085</id>
    <name>London</name>
  </departureCity>
  <departureTime>2024-05-13T13:37:45.624</departureTime>
  <id>62b75cef-599c-4575-be37-638a0e8d661c</id>
</return>
</ns2:getFlightsFromResponse>
</S:Body>
</S:Envelope>

```

## 2.6.2 Przykład użycia GetTicket

Komunikat wysłany

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <S:Body>
    <ns2:getTicket xmlns:ns2="http://webservices.ticketswebservice.rsi/">
      <arg0>98828f31-8204-4b05-8a81-256189586a22</arg0>
    </ns2:getTicket>
  </S:Body>
</S:Envelope>

```

Komunikat odebrany

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <S:Body>
    <ns2:getTicketResponse xmlns:ns2="http://webservices.ticketswebservice.rsi/">

```

```
<return>
  <flight>
    <arrivalCity>
      <country>UK</country>
      <id>dbfe42ad-bbe4-42ec-b90b-034fbc049085</id>
      <name>London</name>
    </arrivalCity>
    <availableSeats>50</availableSeats>
    <departureCity>
      <country>USA</country>
      <id>1df7ea9c-a883-4692-99f9-3e0e8979c5c6</id>
      <name>New York</name>
    </departureCity>
    <departureTime>2025-03-20T22:50:45.623</departureTime>
    <id>f5e0f8cb-218e-4c92-ba1d-f175edfd44ea</id>
  </flight>
  <id>98828f31-8204-4b05-8a81-256189586a22</id>
  <passengerFirstName>John</passengerFirstName>
  <passengerSurname>Doe</passengerSurname>
</return>
</ns2:getTicketResponse>
</S:Body>
</S:Envelope>
```