

Dokumentacja LOT_App

Spis treści

1. Krótki opis:.....	2
2. Instrukcja uruchomienia:.....	2
A) Uruchomienie poprzez plik jar przesłany w mailu:.....	2
B) Uruchomienie poprzez pobranie kodu źródłowego.....	3
2. Linia komend	3
Komendy podstawowe:	4
- help	4
- help commandName	4
Komendy pasażera:	5
- addPassenger firstName lastName phoneNumber.....	5
- getAllPassengers	5
- getPassengerById id	6
- updateFirstName id newFirstName	6
- updateLastName id newLastName	6
- updatePhoneNumber id newPhoneNumber.....	7
- deletePassengerById id	7
- addPassengerToFlight id flightNumber	8
Komendy lotu:	9
- addFlight flightNumber departureAirport destinationAirport departureDate numberOfSeats.....	9
- getAllFlights	9
- getFlightByFlightNumber flightNumber	9
- findFlights --freeSeats --from date --to date --departureAirport airport.....	10
--destinationAirport airport.....	10
- updateDepartureAirport flightNumber newDepartureAirport.....	10
- updateDestinationAirport flightNumber newDestinationAirport.....	11
- updateDepartureDate flightNumber newDepartureDate	11
3. Protokół HTTP	12
Zapytania pasażera:.....	13
http://localhost:8080/passenger/add (POST).....	13

http://localhost:8080/passenger/all (GET)	14
http://localhost:8080/passenger/{id} (GET)	14
http://localhost:8080/passenger/{passengerId}/toFlight/{flightNumber} (PUT)	15
http://localhost:8080/passenger/{id} (DELETE)	16
http://localhost:8080/passenger/{id}/updatePhoneNumber/{newPhoneNumber} (PUT)	17
http://localhost:8080/passenger/{id}/updateFirstName/{newFirsName} (PUT)	17
http://localhost:8080/passenger/{id}/updateLastName/{newLasName} (PUT)	17
Komendy lotu:	18
http://localhost:8080/flight/add (POST)	18
http://localhost:8080/flight/all (GET)	19
http://localhost:8080/flight/{flightNumber} (GET)	20
http://localhost:8080/flight/{flightNumber} (DELETE)	20
http://localhost:8080/flight/find (GET)	20
http://localhost:8080/flight/{flightNumber}/updateDepartureAirport/{newDepartureAirport} (PUT)	22
http://localhost:8080/flight/{flightNumber}/updateDestinationAirport/{newDestinationAirport} (PUT)	22
http://localhost:8080/flight/LO3334/updateDepartureDate/{newDepartureDate} (PUT)	22

1. Krótki opis:

System pozwala na zarządzanie bazą lotów oraz pasażerów zarówno poprzez linię komend jak i protokół HTTP.

2. Instrukcja uruchomienia:

Do maila powinien zostać załączony plik jar który wystarczy uruchomić aby móc korzystać z aplikacji w mailu udostępniam też kod źródłowy aplikacji.

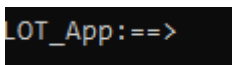
A) Uruchomienie poprzez plik jar przesłany w mailu:

Należy wejść w folder do którego pobraliśmy przesłany plik jar następnie uruchomić linię komend poleceniem cmd. Po uruchomieniu linii komend należy wpisać komendę: ***java -jar LotApp-0.0.1-SNAPSHOT.jar***

Jeśli wszystko zadziałało poprawnie to powinien się pojawić banner startowy



Gdy pojawi się znacznik



Oznacza to że program działa poprawnie i można zacząć użytkowanie.

B) Uruchomienie poprzez pobranie kodu źródłowego

W folderze do którego został pobrany kod źródłowy należy uruchomić linie komend a następnie wpisać komendę **mvn package** aby zbudować projekt.

Jeśli wszystko poszło w porządku to powinna się pokazać informacja o poprawnym zbudowaniu aplikacji

```
jar.original
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.395 s
[INFO] Finished at: 2024-04-18T23:15:58+02:00
[INFO] -----
```

Teraz w folderze target powinien się znajdować plik jar za którego pomocą można uruchomić aplikację (uruchomienie zostało pokazane w podpunkcie A)

2.Linia komend

Zarządzać lotami i pasażerami możemy zarządzać poprzez linie komend.
Wpisując odpowiednie komendy w wyznaczonym miejscu:

```
LOT_App:==>
```

Komendy podstawowe:

- *help*

Pozwala na wyświetlenie wszystkich komend oraz krótkiej informacji o nich

przykład:

```
LOT_App:==>help
AVAILABLE COMMANDS

Built-In Commands
  help: Display help about available commands
  stacktrace: Display the full stacktrace of the last error.
  quit, exit: Exit the shell.
  history: Display or save the history of previously run commands

Flight Commands
  addFlight: Add a new flight
  deleteFlight: Delete a flight
  getFlight: Get a flight by ID
  listFlights: List all flights
  updateFlight: Update a flight
```

- *help commandName*

Pozwala na uzyskanie krótkiej informacji o danej komendzie

```
LOT_App:==>help addPassenger
NAME
  addPassenger - Add passenger
example: addPassenger firstName telephoneNumber

SYNOPSIS
  addPassenger [--firstName String] [--lastName String] [--phoneNumber String] --help

OPTIONS
  --firstName String
    [Mandatory]

  --lastName String
    [Mandatory]

  --phoneNumber String
    [Mandatory]

  --help or -h
    help for addPassenger
    [Optional]
```

Komendy pasażera:

- *addPassenger firstName lastName phoneNumber*

Komenda ta dodaje pasażera na podstawie imienia, nazwiska oraz numeru telefonu (który powinien być podany w formacie 123456789) oraz przypisuje mu unikatowe id.

Komenda ta powinna zwrócić informacje o stworzonym pasażerze:

```
LOT_App==>addPassenger Adam Nowak 123456789
(
id = 1
firstName = Adam
lastName = Nowak
phoneNumber = 123456789
  flights = []
)
```

Jeśli wpisujemy niepoprawne dane albo nie podamy ich wcale powinny zostać zwrócone błędy:

```
LOT_App==>addPassenger
←[31mMissing mandatory option '--firstName'←[0m
←[31mMissing mandatory option '--phoneNumber'←[0m
←[31mMissing mandatory option '--lastName'←[0m
```

```
LOT_App==>addPassenger Adam Nowak 12345678
Phone Number must have nine digits example 122137321
```

- *getAllPassengers*

Pozwala na wyświetlenie informacji o wszystkich zapisanych pasażerach

```
LOT_App==>getAllPassengers
[(
id = 1
firstName = Adam
lastName = Nowak
phoneNumber = 123456789
  flights = []
), (
id = 2
firstName = Kamil
lastName = Bodnar
phoneNumber = 111111111
  flights = [(
flightNumber = L03334
departureAirport = Krakow
destinationAirport = Warszawa
departureDate = 2024-12-01T10:00
numberFreeSeats = 119
)]
)]
```

(każdy pasażer posiada pole flights które pokazuje do których lotów jest aktualnie przypisany)

- *getPassengerById id*

Pozwala na wypisanie informacji o pasażerze o danym id podanym przez użytkownika

```
LOT_App==>getPassengerById 2
(
  id = 2
  firstName = Kamil
  lastName = Bodnar
  phoneNumber = 111111111
  flights = [(
    flightNumber = L03334
    departureAirport = Krakow
    destinationAirport = Warszawa
    departureDate = 2024-12-01T10:00
    numberFreeSeats = 119
  )]
)
```

- *updateFirstName id newFirstName*

Pozwala na zmianę imienia pasażera o podanym id

```
LOT_App==>getPassengerById 1
(
  id = 1
  firstName = Adam
  lastName = Nowak
  phoneNumber = 123456789
  flights = []
)
LOT_App==>updateFirstName 1 Emil
Done
LOT_App==>getPassengerById 1
(
  id = 1
  firstName = Emil
  lastName = Nowak
  phoneNumber = 123456789
  flights = []
)
```

- *updateLastName id newLastName*

Pozwala na zmianę nazwiska pasażera o podanym id

```

LOT_App==>getPassengerById 1
(
id = 1
firstName = Emil
lastName = Nowak
phoneNumber = 123456789
  flights = []
)
LOT_App==>updateLastName 1 Kruk
Done
LOT_App==>getPassengerById 1
(
id = 1
firstName = Emil
lastName = Kruk
phoneNumber = 123456789
  flights = []
)

```

- *updatePhoneNumber id newPhoneNumber*

Pozwala na zmianę numeru telefonu pasażera o podanym id

```

LOT_App==>getPassengerById 1
(
id = 1
firstName = Emil
lastName = Kruk
phoneNumber = 123456789
  flights = []
)
LOT_App==>updatePhoneNumber 1 212137777
Done
LOT_App==>getPassengerById 1
(
id = 1
firstName = Emil
lastName = Kruk
phoneNumber = 212137777
  flights = []
)

```

- *deletePassengerById id*

Usuwa pasażera o podanym id

```

LOT_App==>addPassenger Kamil Nowak 123456789
(
id = 1
firstName = Kamil
lastName = Nowak
phoneNumber = 123456789
  flights = []
)
LOT_App==>deletePassengerById 1
Done
LOT_App==>getPassengerById 1
Passenger not found

```

- *addPassengerToFlight id flightNumber*

Dodaje pasażera o podanym id do lotu o zadanym numerze.

Zwraca zaktualizowane informacje o pasażerze.

Po dodaniu pasażera do lotu liczba miejsc zostaje zmniejszona. Gdy pasażer zostanie usunięty liczba wolnych miejsc powiększy się.

Nie jest możliwe dodanie pasażera do lotu bez wolnych miejsc.

```

LOT_App==>getFlightByFlightNumber L0334
(
flightNumber = L0334
departureAirport = Krakow
destinationAirport = Warszawa
departureDate = 2023-11-11T12:10
numberFreeSeats = 120
)
LOT_App==>getPassengerById 2
(
id = 2
firstName = Kamil
lastName = Nowak
phoneNumber = 123456789
  flights = []
)
LOT_App==>addPassengerToFlight 2 L0334
(
id = 2
firstName = Kamil
lastName = Nowak
phoneNumber = 123456789
  flights = [(
flightNumber = L0334
departureAirport = Krakow
destinationAirport = Warszawa
departureDate = 2023-11-11T12:10
numberFreeSeats = 119
)]
)

```


Komendy lotu:

- *addFlight flightNumber departureAirport destinationAirport departureDate numberOfSeats*

Zapisuje lot na podstawie powyższych parametrów i wypisuje zapisany lot, wszystkie są obowiązkowe, każdy lot musi mieć unikatowy numer oraz data ma być w formacie: yyyy-MM-dd_HH:mm:ss

```
LOT_App==>addFlight L0335 Krakow Warszawa 2024-03-11_11:11:00 120
(
flightNumber = L0335
departureAirport = Krakow
destinationAirport = Warszawa
departureDate = 2024-03-11T11:11
numberFreeSeats = 120
)
```

- *getAllFlights*

Wypisuje wszystkie zapisane loty

```
LOT_App==>getAllFlights
[(
flightNumber = L0335
departureAirport = Krakow
destinationAirport = Warszawa
departureDate = 2024-03-11T11:11
numberFreeSeats = 120
), (
flightNumber = L0331
departureAirport = Lizbona
destinationAirport = Praga
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)]
```

- *getFlightByFlightNumber flightNumber*

Wypisuje lot o zadanym numerze

```
LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Lizbona
destinationAirport = Praga
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)
```

*- findFlights --freeSeats --from date --to date --departureAirport airport
--destinationAirport airport*

Znajduje lot na podstawie podanych parametrów ale żaden parametr nie jest obowiązkowy jeśli użyjemy samej komendy **findFlights** zadziała ona w ten sam sposób co **getAllFlights**

Parametry:

--freeSeats jeśli zostanie podany ten parametr to wyszukuje tylko loty z wolnymi siedzeniami

--from date jeśli zostanie podany ten parametr to wyszukuje tylko loty z datą wylotu po podanej dacie

--to date jeśli zostanie podany ten parametr to wyszukuje tylko loty z datą wylotu przed podanej dacie

(jeśli zadamy oba parametry **from** i **to** to wyszukuje lot z data wylotu pomiędzy podanymi datami)

--departureAirport airport jeśli zostanie podany ten parametr to wyszukuje tylko loty z podanym lotniskiem wylotu

--destinationAirport airport jeśli zostanie podany ten parametr to wyszukuje tylko loty z podanym lotniskiem docelowym

```
LOT_App==>findFlights --freeSeats --from 2024-01-13_11:11:00 --to 2024-08-13_11:11:00 --departureAirport Lizbona --destinationAirport Praga
[
  {
    FlightNumber = L0331
    departureAirport = Lizbona
    destinationAirport = Praga
    departureDate = 2024-06-13T11:11
    numberFreeSeats = 120
  }
]
```

- updateDepartureAirport flightNumber newDepartureAirport
Aktualizuje lotnisko wylotu dla lotu o podanym numerze

```

LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Lizbona
destinationAirport = Praga
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)
LOT_App==>updateDepartureAirport L0331 Krakow
Done
LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Krakow
destinationAirport = Praga
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)

```

- *updateDestinationAirport flightNumber newDestinationAirport*
Aktualizuje lotnisko docelowe dla lotu o podanym numerze

```

LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Krakow
destinationAirport = Praga
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)
LOT_App==>updateDestinationAirport L0331 Rzym
Done
LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Krakow
destinationAirport = Rzym
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)

```

- *updateDepartureDate flightNumber newDepartureDate*
Aktualizuje datę wylotu dla lotu o podanym numerze, data ma być w formacie:
yyyy-MM-dd_HH:mm:ss

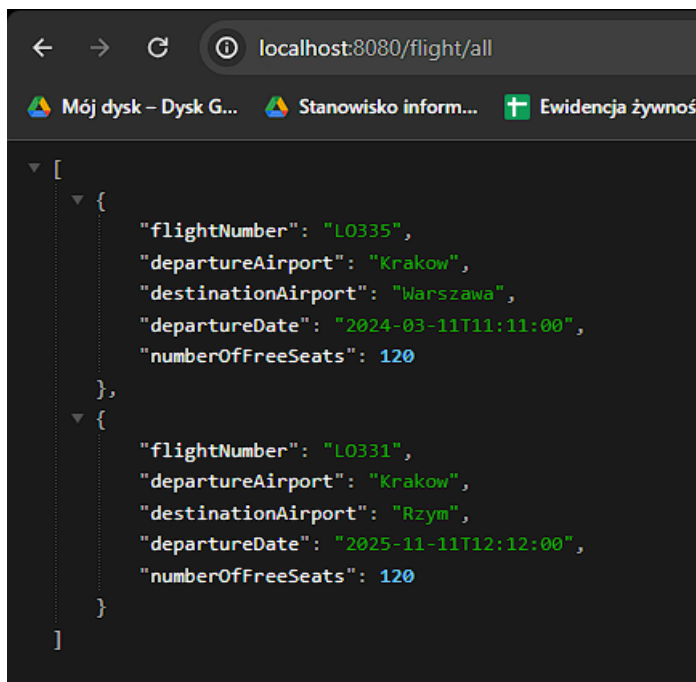
```

LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Krakow
destinationAirport = Rzym
departureDate = 2024-06-13T11:11
numberFreeSeats = 120
)
LOT_App==>updateDepartureDate L0331 2025-11-11_12:12:00
Done
LOT_App==>getFlightByFlightNumber L0331
(
flightNumber = L0331
departureAirport = Krakow
destinationAirport = Rzym
departureDate = 2025-11-11T12:12
numberFreeSeats = 120
)

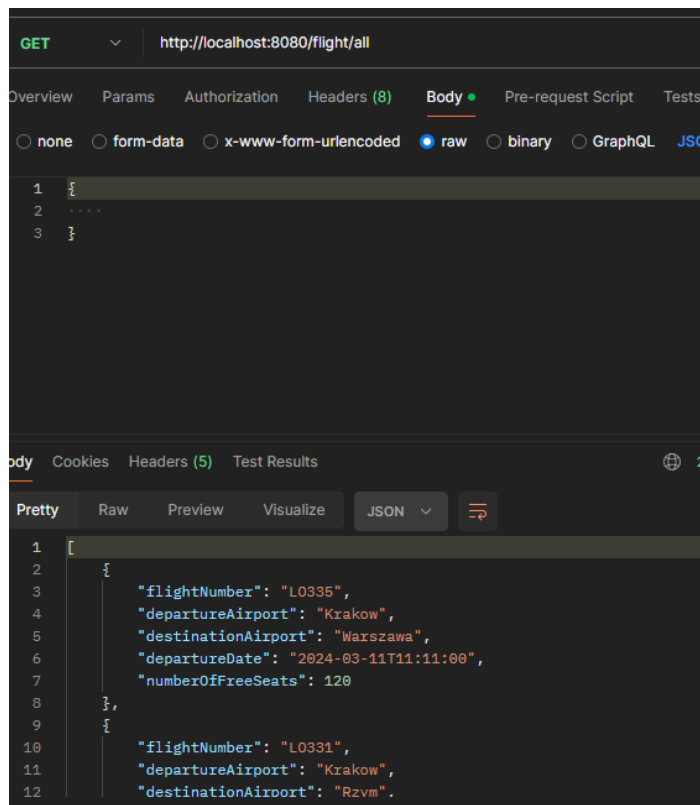
```

3.Protokół HTTP

Gdy uruchomimy aplikacje możemy z nią pracować poprzez okno przeglądarki:



albo używając narzędzia Postman:



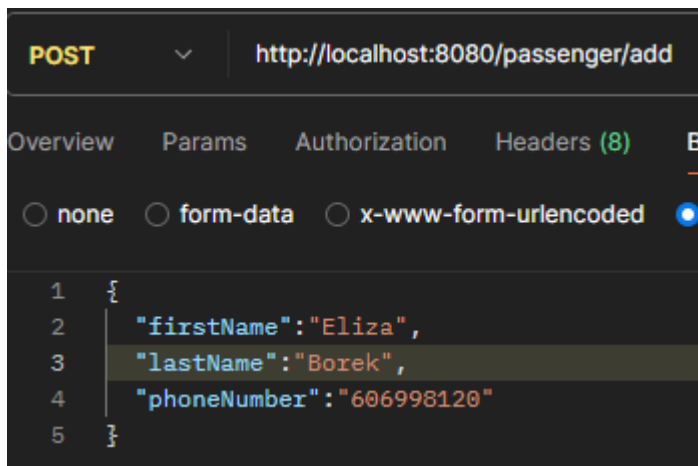
Tutaj link do kolekcji na postmanie zawierającej wszystkie możliwe zapytania:

<https://www.postman.com/seq4/workspace/lot/collection/28941026-4b4a9eca-d77e-44d8-8542-4f1c91c3fab0?action=share&creator=28941026>

Zapytania pasażera:

`http://localhost:8080/passenger/add (POST)`

Zapytanie to dodaje pasażera na podstawie JSON z wartościami imienia, nazwiska oraz numeru telefonu (który powinien być podany w formacie 123456789) oraz przypisuje mu unikatowe id.

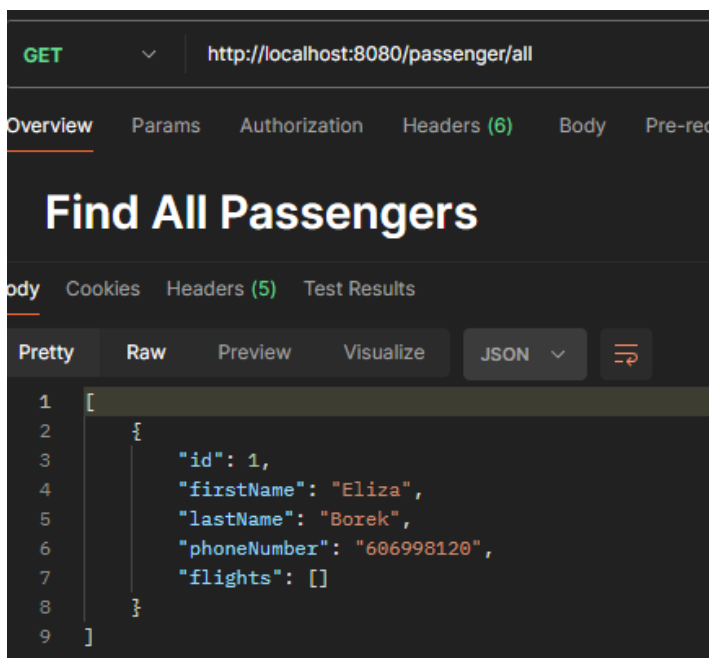


Zapytanie to powinna zwrócić JSON z informacją o stworzonym pasażerze:



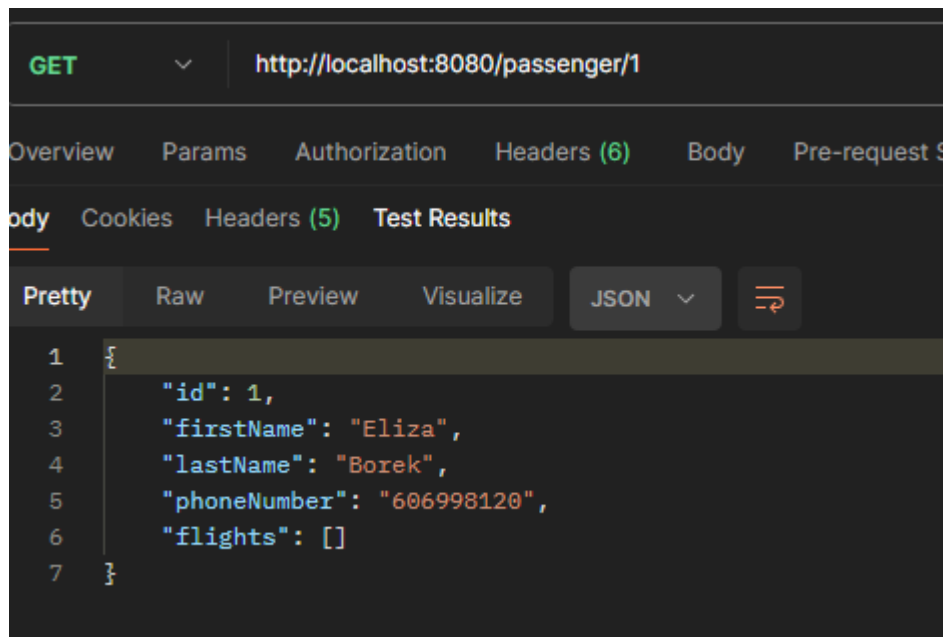
`http://localhost:8080/passenger/all` (GET)

Zapytanie to pokazuje wszystkich zapisanych pasażerów



`http://localhost:8080/passenger/{id}` (GET)

Zapytanie to pokazuje pasażera o podanym id



The screenshot shows a REST client interface with a GET request to `http://localhost:8080/passenger/1`. The response is displayed in the 'Test Results' tab, formatted as JSON. The JSON object contains the following fields: `id` (1), `firstName` (Eliza), `lastName` (Borek), `phoneNumber` (606998120), and `flights` (an empty array).

```
1 {  
2   "id": 1,  
3   "firstName": "Eliza",  
4   "lastName": "Borek",  
5   "phoneNumber": "606998120",  
6   "flights": []  
7 }
```

`http://localhost:8080/passenger/{passengerId}/toFlight/{flightNumber}` (PUT)

Dodaje pasażera o podanym id do lotu o zadany numerze.

Zwraca zaktualizowane informacje o pasażerze.

Po dodaniu pasażera do lotu liczba miejsc zostaje zmniejszona. Gdy pasażer zostanie usunięty liczba wolnych miejsc powiększy się.

Nie jest możliwe dodanie pasażera do lotu bez wolnych miejsc.

```
PUT http://localhost:8080/passenger/1/toFlight/LO221

Overview Params Authorization Headers (7) Body Pre-request Sc

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

1 {
2   "id": 1,
3   "firstName": "Eliza",
4   "lastName": "Borek",
5   "phoneNumber": "660999540",
6   "flights": [
7     {
8       "flightNumber": "LO221",
9       "departureAirport": "Krakow",
10      "destinationAirport": "Warszawa",
11      "departureDate": "2022-12-01T10:00:00",
12      "numberOfFreeSeats": 119
13    }
14  ]
15 }
```

http://localhost:8080/passenger/{id} (DELETE)

usuwa pasażera o podanym id

```
DELETE http://localhost:8080/passenger/1

Overview Params Authorization Headers (6) Body Pre-req

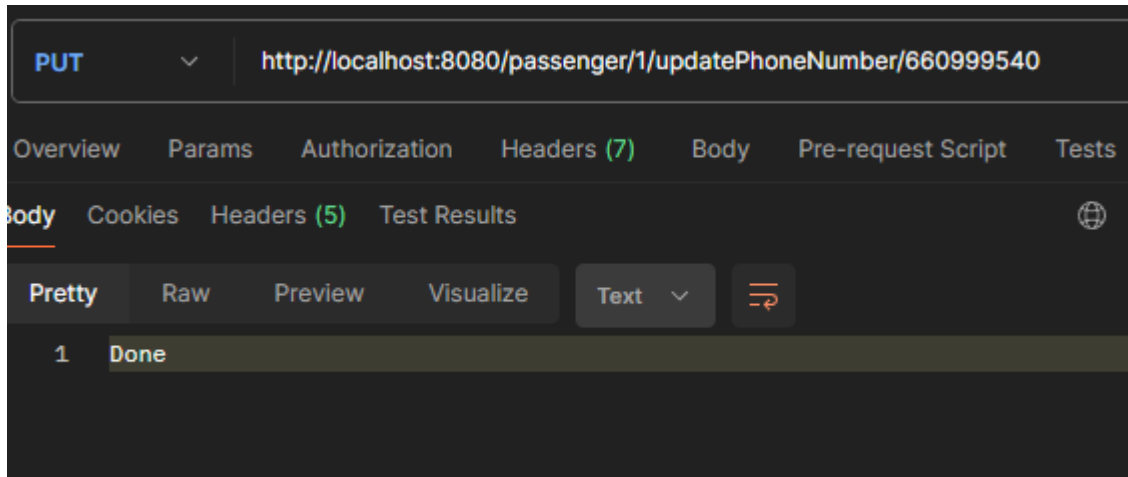
Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

1 Done
```

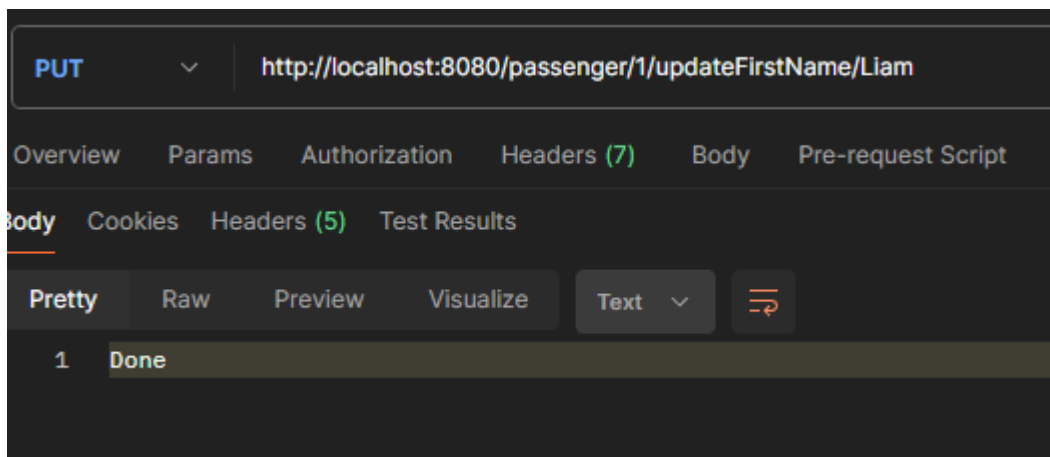

<http://localhost:8080/passenger/{id}/updatePhoneNumber/{newPhoneNumber}> (PUT)

Pozwala na zmianę numeru telefonu pasażera o podanym id



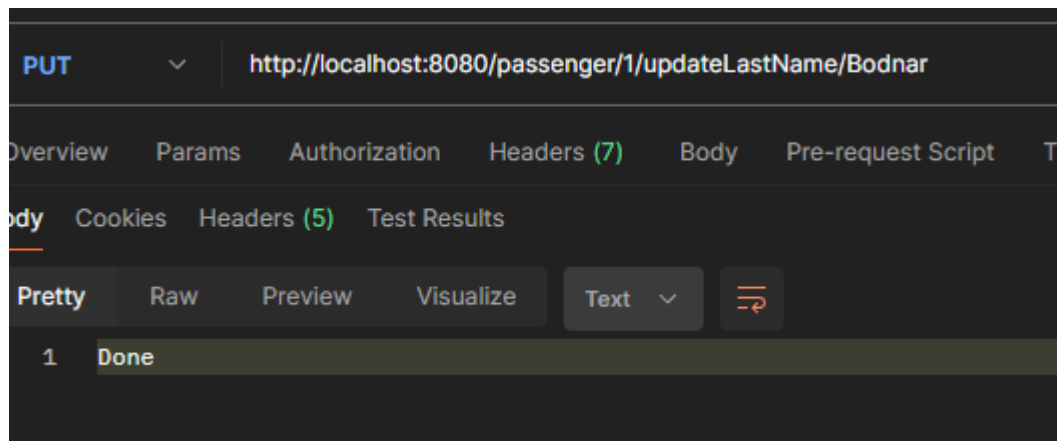
<http://localhost:8080/passenger/{id}/updateFirstName/{newFirstName}> (PUT)

Pozwala na zmianę imienia pasażera o podanym id



<http://localhost:8080/passenger/{id}/updateLastName/{newLastName}> (PUT)

Pozwala na zmianę nazwiska pasażera o podanym id



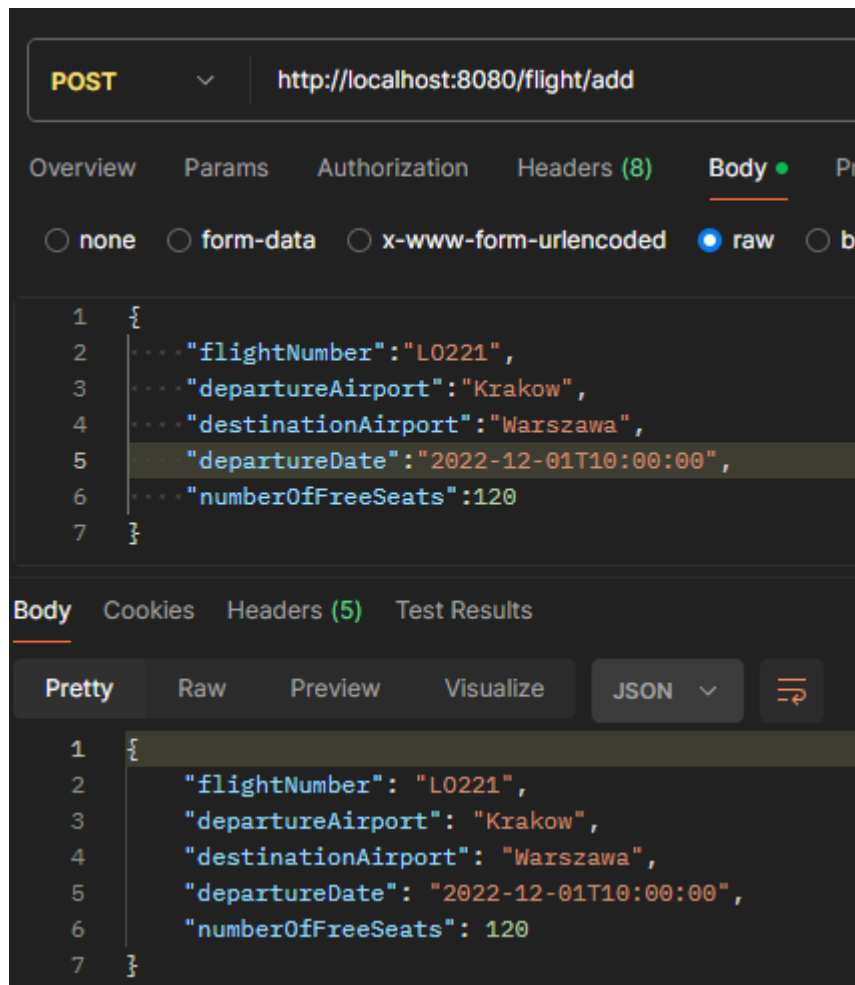
Komendy lotu:

http://localhost:8080/flight/add (POST)

Zapisuje lot na podstawie JSON

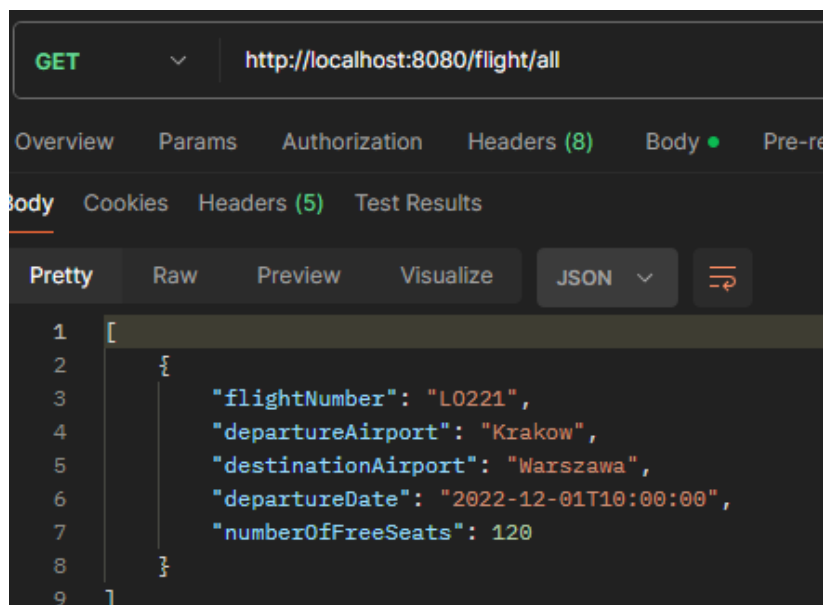
```
{
  "flightNumber": "L0221",
  "departureAirport": "Krakow",
  "destinationAirport": "Warszawa",
  "departureDate": "2022-12-01_10:00:00",
  "numberOfFreeSeats": 120
}
```

i wypisuje zapisany lot:



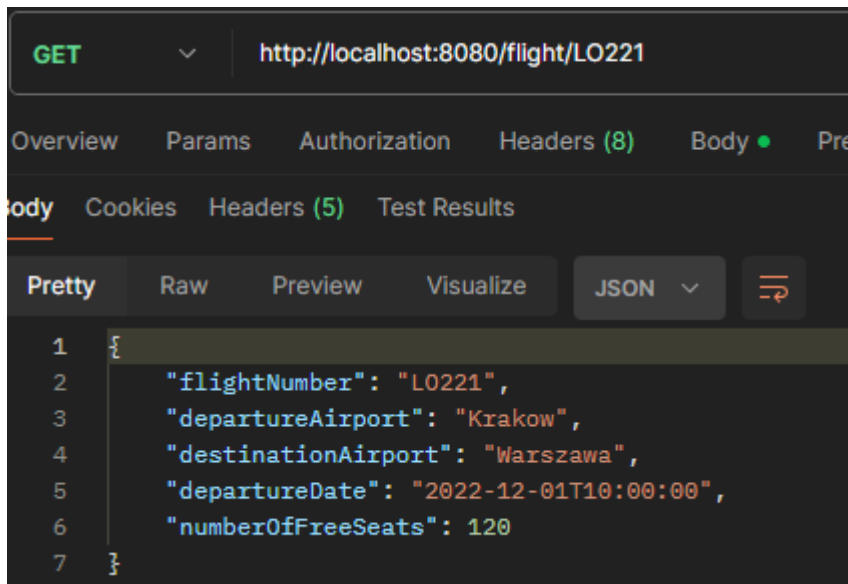
`http://localhost:8080/flight/all` (GET)

pokazuje wszystkie zapisane loty



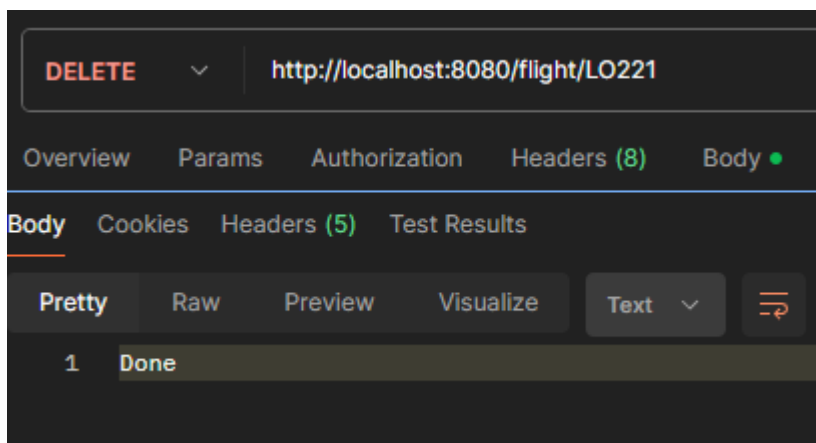
http://localhost:8080/flight/{flightNumber} (GET)

pokazuje wszystkie loty o podanym numerze



http://localhost:8080/flight/{flightNumber} (DELETE)

usuwa lot o podanym numerze



http://localhost:8080/flight/find (GET)

Znajduje lot na podstawie podanych parametrów ale żaden parametr nie jest obowiązkowy

Parametry:

freeSeats:true jeśli zostanie podany ten parametr to wyszukuje tylko loty z wolnymi siedzeniami

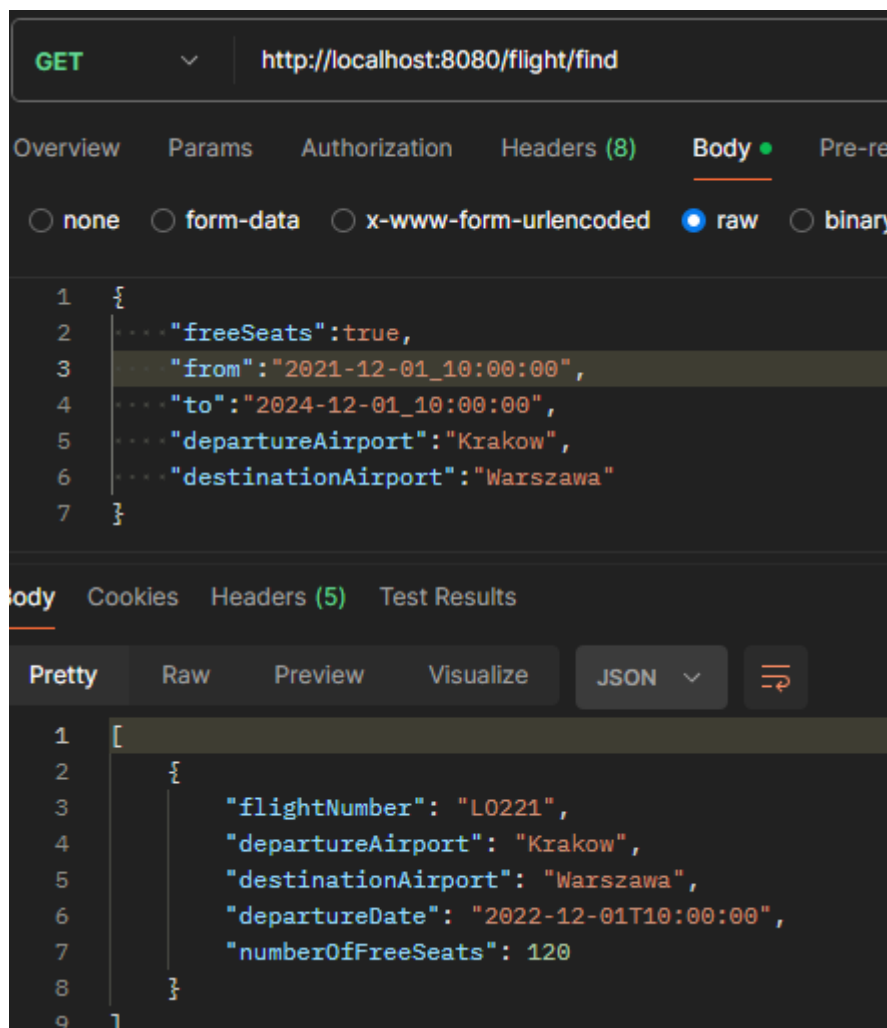
from:date jeśli zostanie podany ten parametr to wyszukuje tylko loty z datą wylotu po podanej dacie

to: date jeśli zostanie podany ten parametr to wyszukuje tylko loty z datą wylotu przed podanej dacie

(jeśli zadamy oba parametry **from** i **to** to wyszukuje lot z data wylotu pomiędzy podanymi datami)

departureAirport:airport jeśli zostanie podany ten parametr to wyszukuje tylko loty z podanym lotniskiem wylotu

destinationAirport:airport jeśli zostanie podany ten parametr to wyszukuje tylko loty z podanym lotniskiem docelowym



The screenshot shows a REST client interface with a GET request to `http://localhost:8080/flight/find`. The request body is a JSON object with the following fields: `freeSeats: true`, `from: "2021-12-01_10:00:00"`, `to: "2024-12-01_10:00:00"`, `departureAirport: "Krakow"`, and `destinationAirport: "Warszawa"`. The response body is a JSON array containing one object with the following fields: `flightNumber: "LO221"`, `departureAirport: "Krakow"`, `destinationAirport: "Warszawa"`, `departureDate: "2022-12-01T10:00:00"`, and `numberOfFreeSeats: 120`.

```
GET http://localhost:8080/flight/find

Overview Params Authorization Headers (8) Body • Pre-re

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary

1 {
2   ... "freeSeats": true,
3   ... "from": "2021-12-01_10:00:00",
4   ... "to": "2024-12-01_10:00:00",
5   ... "departureAirport": "Krakow",
6   ... "destinationAirport": "Warszawa"
7 }

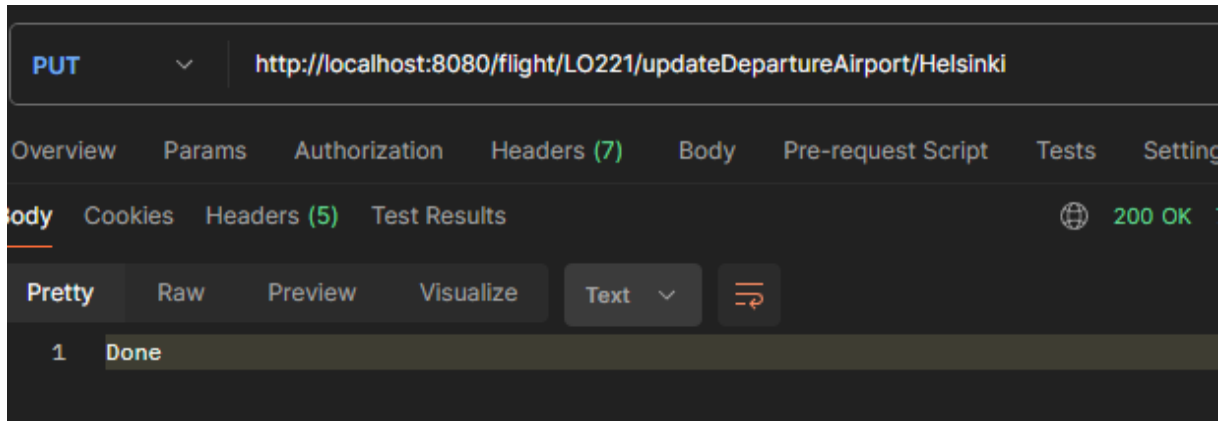
Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ↕

1 [
2   {
3     "flightNumber": "LO221",
4     "departureAirport": "Krakow",
5     "destinationAirport": "Warszawa",
6     "departureDate": "2022-12-01T10:00:00",
7     "numberOfFreeSeats": 120
8   }
9 ]
```

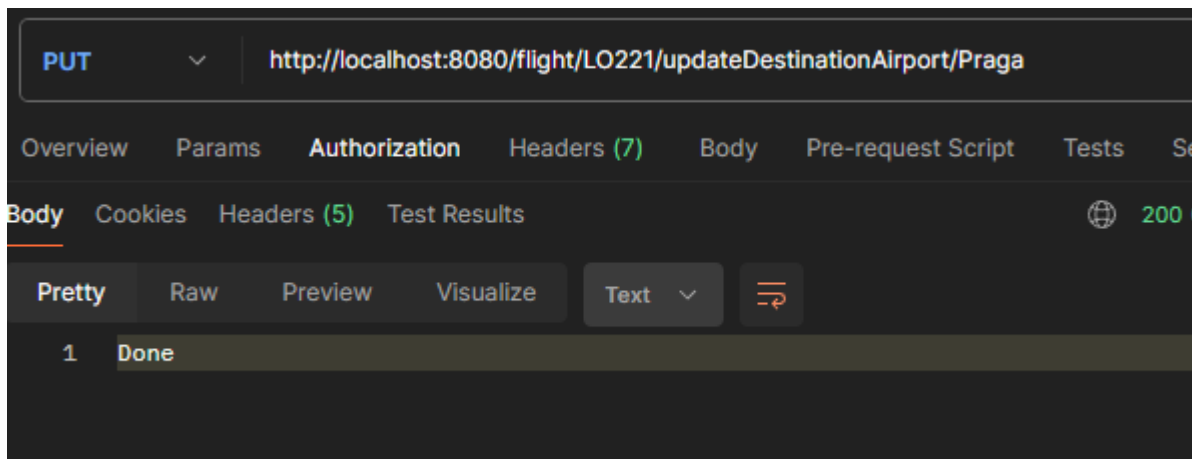
http://localhost:8080/flight/{flightNumber}/updateDepartureAirport/{newDepartureAirport} (PUT)

aktualizuje lotnisko wylotu



http://localhost:8080/flight/{flightNumber}/updateDestinationAirport/{newDestinationAirport} (PUT)

aktualizuje lotnisko docelowe



http://localhost:8080/flight/LO3334/updateDepartureDate/{newDepartureDate} (PUT)

aktualizuje date wylotu

