Konfiguracja testów w Postmanie

- Endpoint: `http://localhost:8080/flights` (lub odpowiedni adres API)
- Metoda: GET
- Nagłówki: `Content-Type: application/json`

Scenariusze i przypadki testowe

Scenariusz 1: Wyszukiwanie lotów z pełnymi parametrami

- **Wejście**:
  - `departure=New York`
  - `arrival=Warsaw`
  - `date=05-12-2024`
- **Kroki**:
  - `Wyślij żądanie GET z podanymi parametrami.`
- **Oczekiwany wynik**:
  - `Status: 200 OK`

```
Treść odpowiedzi:
[
  {
    "departure": "New York",
    "arrival": "Warsaw",
    "flightNumber": "PL405",
    "date": "05-12-2024"
  }
]
```

  -

Scenariusz 2: Wyszukiwanie lotów na podstawie miejsca wylotu

- **Wejście**:

  `departure=Paris`

- **Kroki**:

Wyślij żądanie GET z parametrem departure=Paris.

- **Oczekiwany wynik**:

  Status: 200 OK

  Treść odpowiedzi:

```
[
  {
    "departure": "Paris",
    "arrival": "Rio de Janeiro",
    "flightNumber": "FR455",
    "date": "06-12-2024"
  },
  {
    "departure": "Paris",
    "arrival": "Warsaw",
    "flightNumber": "FR705",
    "date": "07-12-2024"
  }
]
```

Scenariusz 3: Brak wyników dla nieistniejących parametrów

- **Wejście**:
  - departure=Tokyo
  - arrival=Warsaw
  - date=05-12-2024
- **Kroki**:
  - Wyślij żądanie GET z podanymi parametrami.

- **Oczekiwany wynik**:
    - Status: 200 OK

      Treść odpowiedzi:
      json
      Skopiuj kod
      []

    -

Scenariusz 4: Wyszukiwanie bez parametrów (wszystkie loty)

- **Wejście**:
    - Brak parametrów.
- **Kroki**:
    - Wyślij żądanie GET bez żadnych parametrów.
- **Oczekiwany wynik**:
    - Status: 200 OK

      Treść odpowiedzi zawierająca wszystkie loty:
      [

        {

          "departure": "New York",

          "arrival": "Warsaw",

          "flightNumber": "PL405",

          "date": "05-12-2024"

        },

        {

          "departure": "New York",

          "arrival": "Rome",

          "flightNumber": "US205",

          "date": "06-12-2024"

        },

```
{
  "departure": "Paris",
  "arrival": "Rio de Janeiro",
  "flightNumber": "FR455",
  "date": "06-12-2024"
},
{
  "departure": "Paris",
  "arrival": "Warsaw",
  "flightNumber": "FR705",
  "date": "07-12-2024"
},
{
  "departure": "Rome",
  "arrival": "Madrid",
  "flightNumber": "IT200",
  "date": "08-12-2024"
},
{
  "departure": "London",
  "arrival": "Toronto",
  "flightNumber": "UK755",
  "date": "08-12-2024"
}
```

Scenariusz 5: Wyszukiwanie lotów na podstawie daty

- **Wejście**:
  - date=06-12-2024
- **Kroki**:
  - Wyślij żądanie GET z parametrem date=06-12-2024.
- **Oczekiwany wynik**:
  - Status: 200 OK

    Treść odpowiedzi:

    ```
    [
      {
        "departure": "New York",
        "arrival": "Rome",
        "flightNumber": "US205",
        "date": "06-12-2024"
      },
      {
        "departure": "Paris",
        "arrival": "Rio de Janeiro",
        "flightNumber": "FR455",
        "date": "06-12-2024"
      }
    ]
    ```

  -

Scenariusz 6: Wyszukiwanie lotów, które pasują do kilku kryteriów

- **Wejście**:
  - departure=Paris
  - arrival=Warsaw
  - date=07-12-2024
- **Kroki**:
  - Wyślij żądanie GET z parametrami departure=Paris, arrival=Warsaw, date=07-12-2024.
- **Oczekiwany wynik**:
  - Status: 200 OK

    Treść odpowiedzi:

```
[
  {
    "departure": "Paris",
    "arrival": "Warsaw",
    "flightNumber": "FR705",
    "date": "07-12-2024"
  }
]
```

Scenariusz 7: Wyszukiwanie lotów, które nie pasują do daty

- **Wejście**:
  - departure=London
  - arrival=Toronto
  - date=05-12-2024
- **Kroki**:
  - Wyślij żądanie GET z parametrami departure=London, arrival=Toronto, date=05-12-2024.
- **Oczekiwany wynik**:
  - Status: 200 OK

Treść odpowiedzi:
[]

Scenariusz 8: Wyszukiwanie lotów z niepełnymi parametrami

- **Wejście**:
  - departure=Rome
- **Kroki**:
  - Wyślij żądanie GET z parametrem departure=Rome.
- **Oczekiwany wynik**:
  - Status: 200 OK

    Treść odpowiedzi:
    [

      {

        "departure": "Rome",

        "arrival": "Madrid",

        "flightNumber": "IT200",

        "date": "08-12-2024"

      }

    ]

Scenariusz 9: Wyszukiwanie lotów, które pasują do niepełnych nazw miejsc

- **Wejście**:
  - departure=Par
- **Kroki**:
  - Wyślij żądanie GET z parametrem departure=Par.
- **Oczekiwany wynik**:
  - Status: 200 OK

Treść odpowiedzi:


```
[
  {
    "departure": "Paris",
    "arrival": "Rio de Janeiro",
    "flightNumber": "FR455",
    "date": "06-12-2024"
  },
  {
    "departure": "Paris",
    "arrival": "Warsaw",
    "flightNumber": "FR705",
    "date": "07-12-2024"
  }
]
```


Scenariusz 10: Wyszukiwanie lotów bez podania parametrów

- **Wejście**: Brak parametrów.
- **Kroki**: Wyślij żądanie GET bez parametrów.
- **Oczekiwany wynik**:
  - Status: 200 OK
  - Treść odpowiedzi: Zawiera wszystkie dostępne loty.

Scenariusz 11: Wyszukiwanie lotów z błędnym formatem daty

- **Wejście**:
  - date=2024/12/05
- **Kroki**:
  - Wyślij żądanie GET z parametrem date=2024/12/05.
- **Oczekiwany wynik**:

- Status: 400 Bad Request lub odpowiednia odpowiedź w przypadku nieprawidłowego formatu daty.

Scenariusz 12: Wyszukiwanie lotów, które pasują do tylko części miejsca wylotu

- **Wejście**:
  - departure=Lon
- **Kroki**:
  - Wyślij żądanie GET z parametrem departure=Lon.
- **Oczekiwany wynik**:
  - Status: 200 OK

    Treść odpowiedzi:

```
[
  {
    "departure": "London",
    "arrival": "Toronto",
    "flightNumber": "UK755",
    "date": "08-12-2024"
  }
]
```

Poniżej scenariusze testowe dla metod POST, PUT i DELETE w kontekście aplikacji, która obsługuje loty

Scenariusz 1: Dodawanie nowego lotu (POST)

- **Cel**: Testowanie metody POST, która pozwala na dodanie nowego lotu do bazy danych.

Wejście:

Body żądania w formacie JSON:

```
{
  "departure": "Berlin",
  "arrival": "Paris",
```

```
  "flightNumber": "DE123",
  "date": "09-12-2024"
}
```

Kroki:

1. Wyślij żądanie POST do endpointa /flights z danymi lotu.
2. Upewnij się, że odpowiedź zawiera status 201 (Created).
3. Sprawdź, czy odpowiedź zawiera dane nowego lotu, takie jak flightNumber, departure, arrival, i date.

Oczekiwany wynik:

Status: 201 Created

Treść odpowiedzi:

```
{
  "departure": "Berlin",
  "arrival": "Paris",
  "flightNumber": "DE123",
  "date": "09-12-2024"
}
```

Scenariusz 2: Aktualizowanie danych lotu (PUT)

- **Cel**: Testowanie metody PUT, która umożliwia aktualizowanie danych istniejącego lotu.

Wejście:

- ID lotu, który chcesz zaktualizować (np. flightNumber = PL405).

Body żądania w formacie JSON:

```
{
  "departure": "New York",
```

```
  "arrival": "London",

  "flightNumber": "PL405",

  "date": "10-12-2024"

     }
```

Scenariusz 3: Usuwanie lotu (DELETE)

- **Cel**: Testowanie metody DELETE, która pozwala na usunięcie lotu z bazy danych.

Wejście:

- ID lotu, który chcesz usunąć (np. flightNumber = PL405).

Kroki:

1. Wyślij żądanie DELETE do endpointa /flights/{flightNumber}.
2. Upewnij się, że odpowiedź zawiera status 200 (OK).
3. Sprawdź, czy lot został usunięty z bazy danych (możesz to zrobić, wysyłając zapytanie GET po usunięciu lotu).

Oczekiwany wynik:

- Status: 200 OK
- Treść odpowiedzi

```
{

  "message": "Flight PL405 was deleted successfully."

}
```

Scenariusz 4: Niepoprawne żądanie POST (walidacja danych)

- **Cel**: Testowanie metody POST, gdy dane są niepełne lub niepoprawne.

Wejście:

Body żądania z brakującym polem departure:

```
{

  "arrival": "Paris",
```

"flightNumber": "DE123",

"date": "09-12-2024"

}

Kroki:

1. Wyślij żądanie POST z niepełnymi danymi (brakuje departure).
2. Oczekuj odpowiedzi z błędem walidacji (np. status 400).

Oczekiwany wynik:

- Status: 400 Bad Request

Treść odpowiedzi:
{

"error": "Departure field is required."

}

Scenariusz 5: Niepoprawne żądanie PUT (nieistniejący lot)

- **Cel**: Testowanie metody PUT, gdy próbujesz zaktualizować nieistniejący lot.

Wejście:

Body żądania:
{

"departure": "New York",

"arrival": "London",

"flightNumber": "NON_EXISTING",

"date": "10-12-2024"

}

Kroki:

1. Wyślij żądanie PUT z danymi lotu, którego numer flightNumber nie istnieje.Oczekuj odpowiedzi z błędem (np. status 404).

Oczekiwany wynik:

- Status: 404 Not Found

2. Treść odpowiedzi:
   json
   Skopiuj kod
   {
3. "error": "Flight NON_EXISTING not found."

   }