

1) Get Flight as Json:

Method: GET

URL: <http://localhost:8000/flight/pegasus>

The screenshot shows a REST client interface with a GET request to <http://localhost:8000/flight/pegasus>. The response status is 200 OK, with a time of 23 ms and a size of 656 B. The response body is displayed in JSON format:

```
1 {
2   "flightNumber": "pegasus",
3   "price": 145,
4   "from": "Istanbul",
5   "to": "Izmir",
6   "departureTime": "2025-03-22-19",
7   "arrivalTime": "2025-03-22-19",
8   "description": "EE",
9   "plane": {
10    "model": "HH",
11    "manufacturer": "II",
12    "yearOfManufacture": 199
13  },
14  "seatsLeft": 11,
15  "capacity": 30,
16  "passengers": [
17    {
18      "id": "4fe0ddb5-f23a-4857-9aa2-e48f93016c83",
19      "firstname": "Michael",
20      "lastname": "Jordan",
21      "age": 33,
22      "gender": "male"
23    }
24  ]
25 }
```

2) Get Flight as Xml:

Method: GET

URL: <http://localhost:8000/flight/pegasus?xml=true>

The screenshot shows a REST client interface with a GET request to <http://localhost:8000/flight/pegasus?xml=true>. The response status is 200 OK, with a time of 26 ms and a size of 911 B. The response body is displayed in XML format:

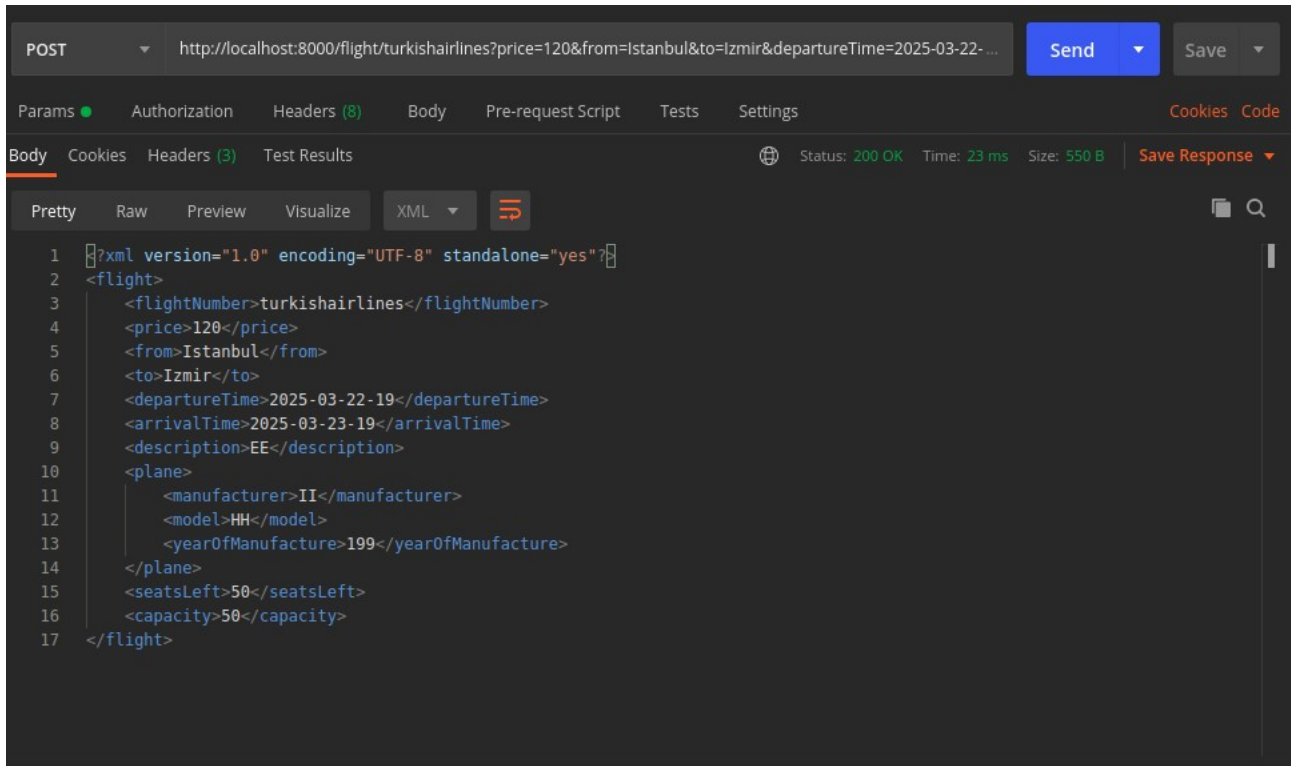
```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <flight>
3   <flightNumber>pegasus</flightNumber>
4   <price>145</price>
5   <from>Istanbul</from>
6   <to>Izmir</to>
7   <departureTime>2025-03-22-19</departureTime>
8   <arrivalTime>2025-03-22-19</arrivalTime>
9   <description>EE</description>
10  <plane>
11    <manufacturer>II</manufacturer>
12    <model>HH</model>
13    <yearOfManufacture>199</yearOfManufacture>
14  </plane>
15  <seatsLeft>11</seatsLeft>
16  <capacity>30</capacity>
17  <passengers>
18    <age>33</age>
19    <firstname>Michael</firstname>
20    <gender>male</gender>
21    <id>4fe0ddb5-f23a-4857-9aa2-e48f93016c83</id>
22  </passengers>
23 </flight>
```

3) Create or Update Flight:

Json Method: POST

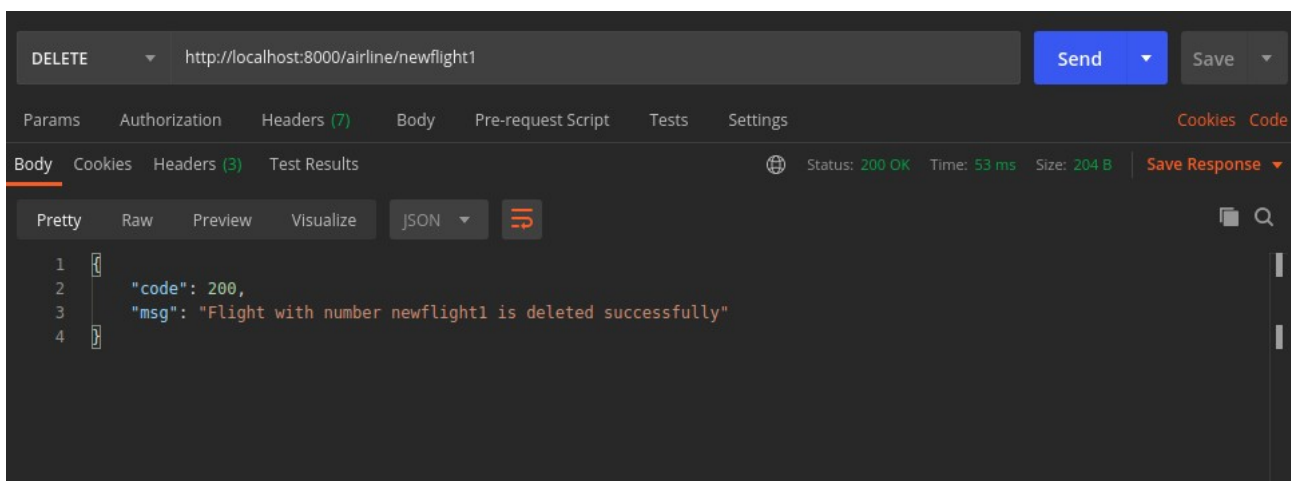
URL: http://localhost:8000/flight/turkishairlines?

price=120&from=Istanbul&to=Izmir&departureTime=2025-03-22-19&arrivalTime=2025-03-23-19&description=EE&capacity=50&model=HH&manufacturer=II&yearOfManufacture=199



4) Delete Flight:

URL: http://localhost:8000/airline/newflight1



5) Get Passenger:

Json Method: GET

URL: <http://localhost:8000/passenger/4fe0ddb5-f23a-4857-9aa2-e48f93016c83>

The screenshot shows a REST client interface with a GET request to `http://localhost:8000/passenger/4fe0ddb5-f23a-4857-9aa2-e48f93016c83`. The response is a JSON object representing a passenger and their reservations.

```
1 {
2   "id": "4fe0ddb5-f23a-4857-9aa2-e48f93016c83",
3   "firstname": "Michael",
4   "lastname": "Jordan",
5   "age": 33,
6   "gender": "male",
7   "phone": "1234567890",
8   "reservations": [
9     {
10      "orderNumber": "75adee72-7cbb-4065-a958-a42f9688264e",
11      "price": 240,
12      "flights": [
13        {
14          "flightNumber": "pegasus",
15          "price": 145,
16          "from": "Istanbul",
17          "to": "Izmir",
18          "departureTime": "2025-03-22-19",
19          "arrivalTime": "2025-03-22-19",
20          "description": "EE",
21          "plane": {
22            "type": "A320neo",
23            "manufacturer": "Airbus",
24            "year": 2022
25          }
26        }
27      ]
28    }
29  ]
30 }
```

6) Get Passenger:

Xml Method: GET

URL: <http://localhost:8000/passenger/4fe0ddb5-f23a-4857-9aa2-e48f93016c83?xml=true>

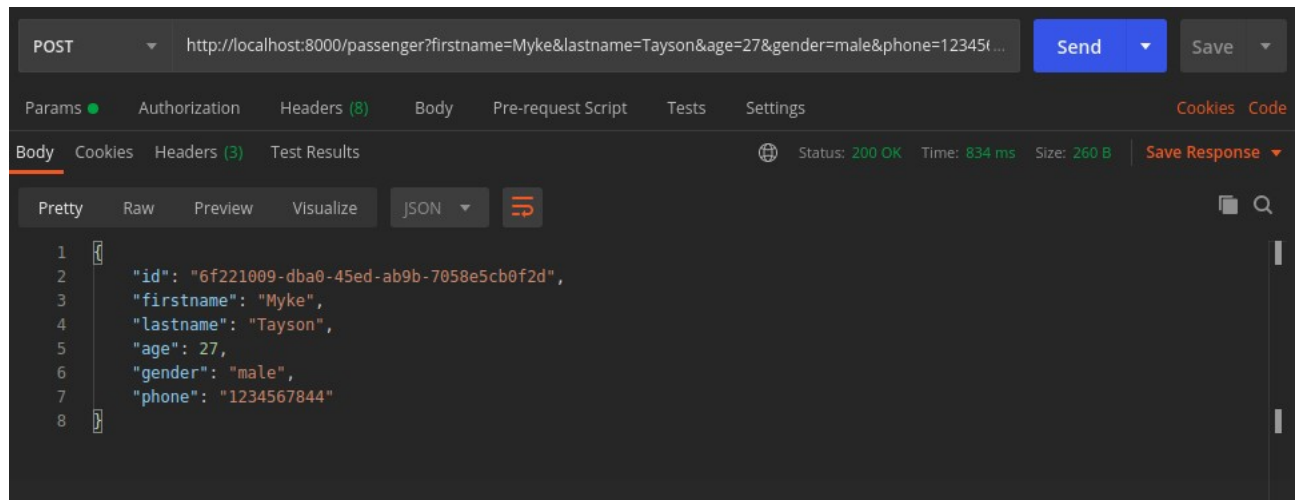
The screenshot shows a REST client interface with a GET request to `http://localhost:8000/passenger/4fe0ddb5-f23a-4857-9aa2-e48f93016c83?xml=true`. The response is an XML document representing the same passenger and reservation data as the JSON response.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <passenger>
3   <age>33</age>
4   <firstname>Michael</firstname>
5   <gender>male</gender>
6   <id>4fe0ddb5-f23a-4857-9aa2-e48f93016c83</id>
7   <lastname>Jordan</lastname>
8   <phone>1234567890</phone>
9   <reservations>
10     <orderNumber>75adee72-7cbb-4065-a958-a42f9688264e</orderNumber>
11     <price>240</price>
12     <flights>
13       <flightNumber>pegasus</flightNumber>
14       <price>145</price>
15       <from>Istanbul</from>
16       <to>Izmir</to>
17       <departureTime>2025-03-22-19</departureTime>
18       <arrivalTime>2025-03-22-19</arrivalTime>
19       <description>EE</description>
20       <plane>
21         <manufacturer>II</manufacturer>
22         <type>A320neo</type>
23         <year>2022</year>
24       </plane>
25     </flights>
26   </reservations>
27 </passenger>
```

URL: <http://localhost:8000/passenger?>

URL: <http://localhost:8000/passenger?>

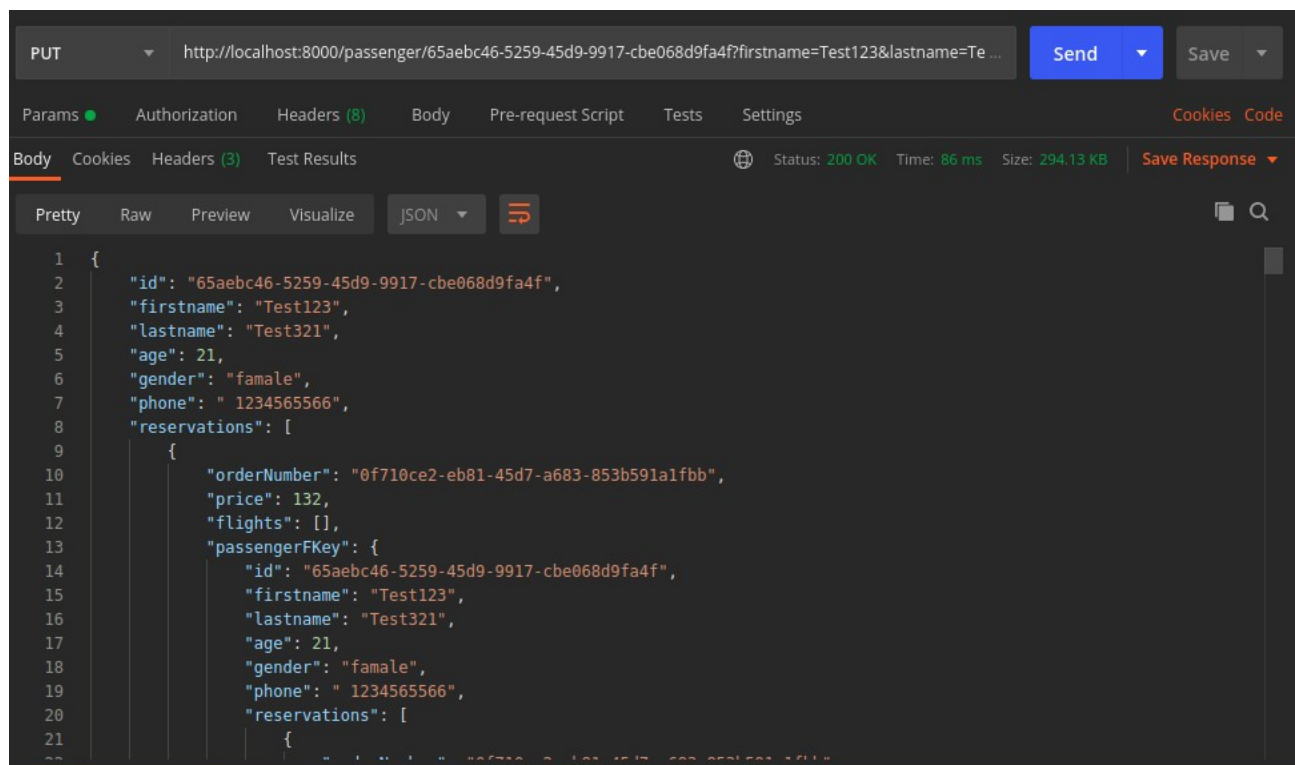
```
firstname=Myke&lastname=Tayson&age=27&gender=male&phone=1234567844
```



8) Update Passenger:

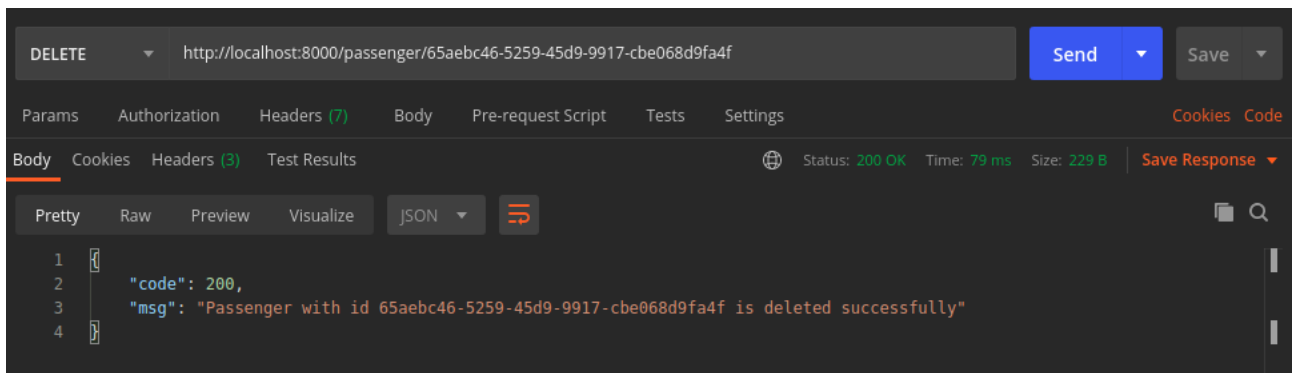
URL: <http://localhost:8000/passenger/65aebc46-5259-45d9-9917-cbe068d9fa4f?>

```
firstname=Test123&lastname=Test321&age=21&gender=famale&phone= 1234565566
```



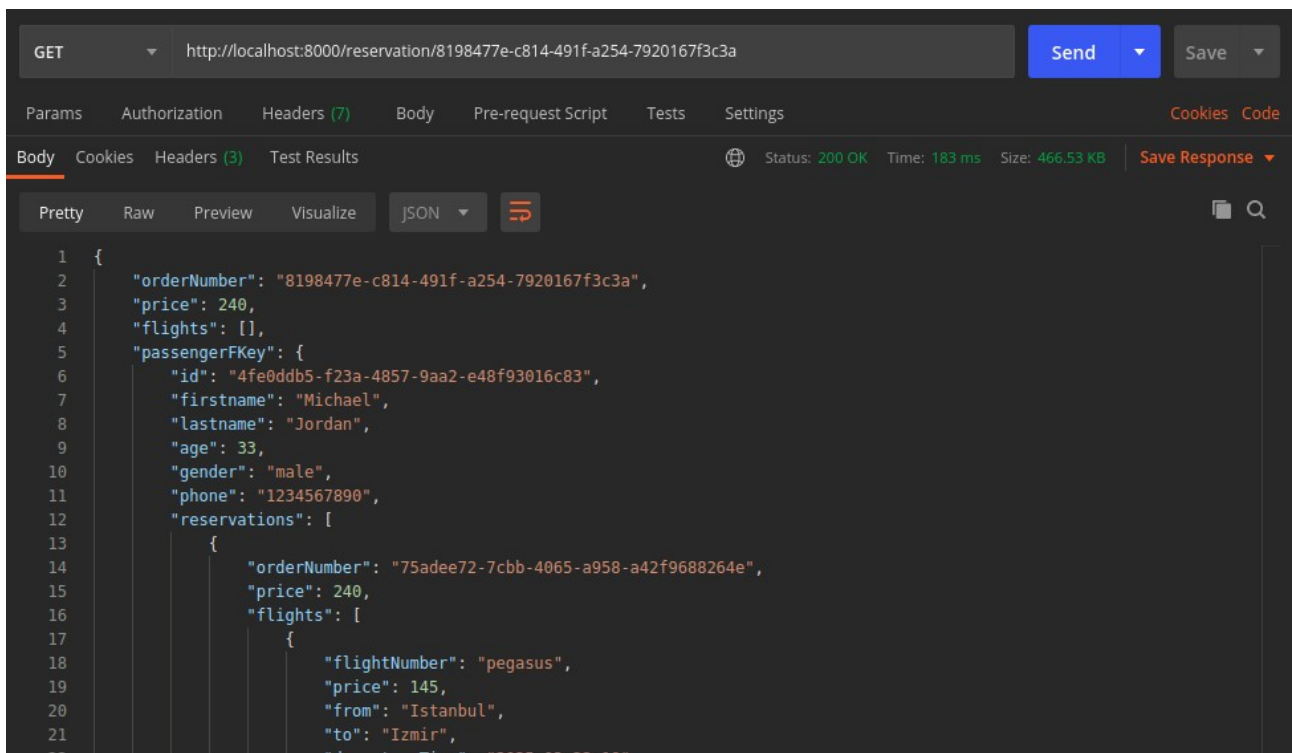
9) Delete Passenger:

URL: <http://localhost:8000/passenger/65aebc46-5259-45d9-9917-cbe068d9fa4f>



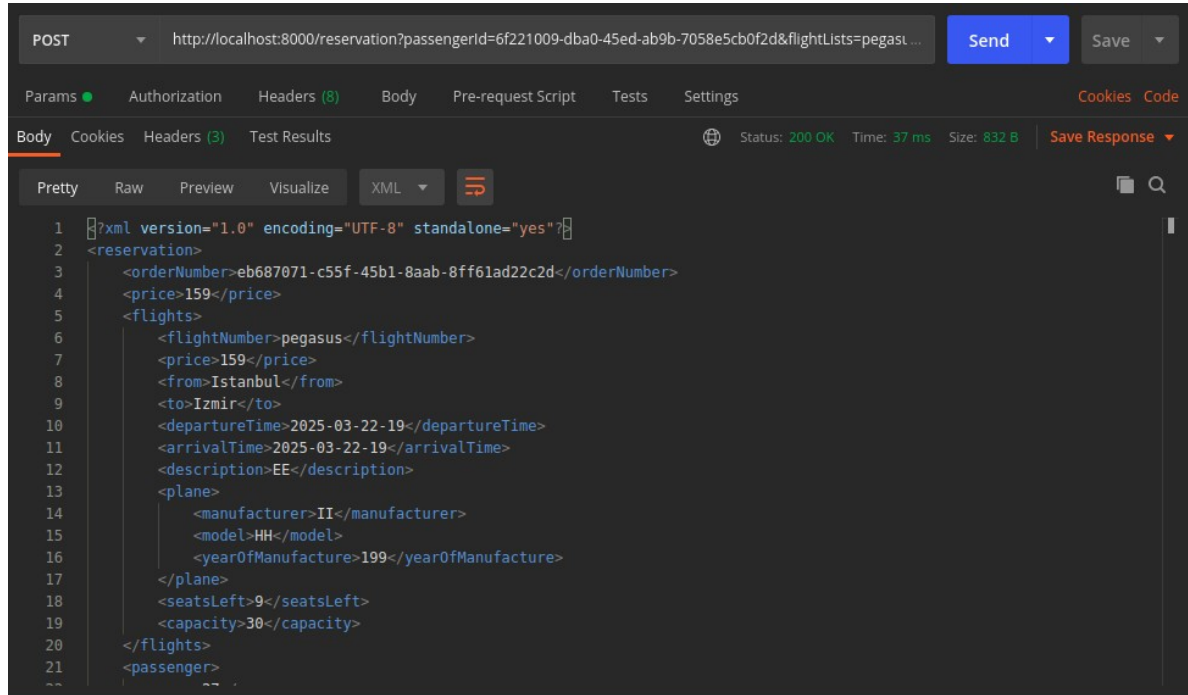
10) Get Reservation:

URL: <http://localhost:8000/reservation/8198477e-c814-491f-a254-7920167f3c3a>



11) Make Reservation:

URL: <http://localhost:8000/reservation?passengerId=6f221009-dba0-45ed-ab9b-7058e5cb0f2d&flightLists=pegasus>



```
POST http://localhost:8000/reservation?passengerId=6f221009-dba0-45ed-ab9b-7058e5cb0f2d&flightLists=pegasu... Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

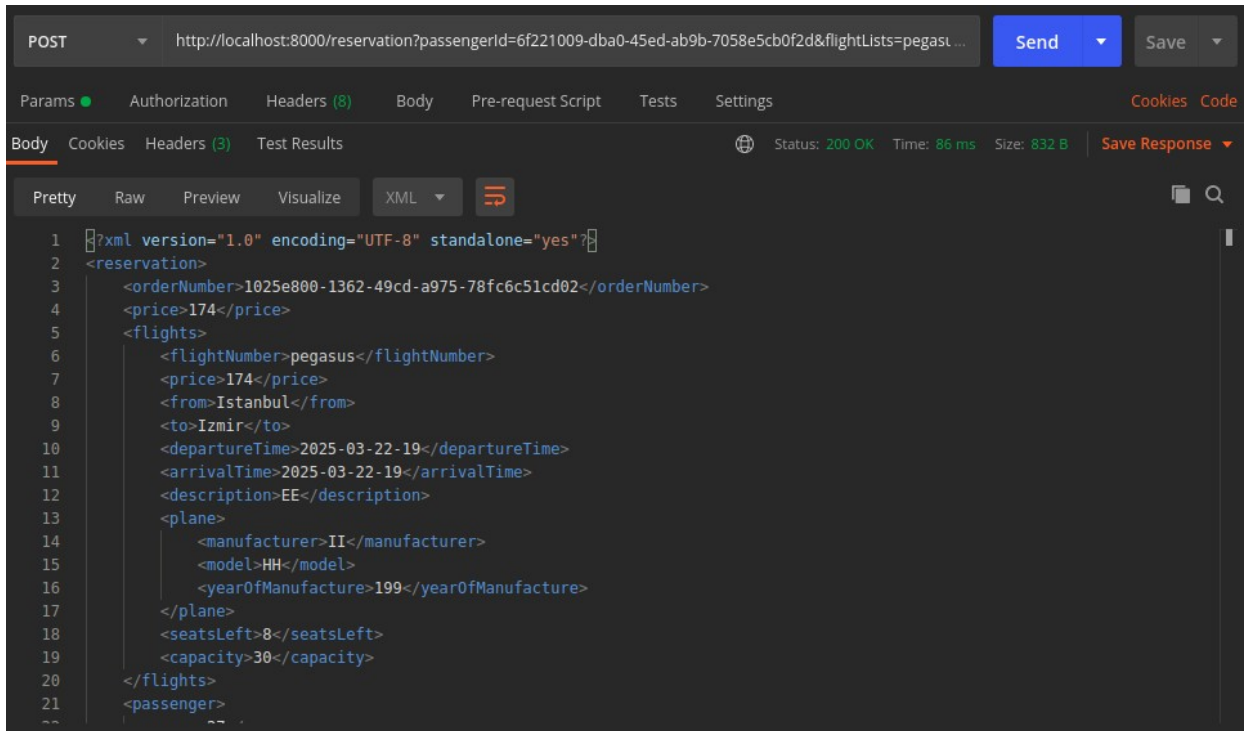
Body Cookies Headers (3) Test Results Status: 200 OK Time: 37 ms Size: 832 B Save Response

Pretty Raw Preview Visualize XML

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <reservation>
3   <orderNumber>eb687071-c55f-45b1-8aab-8ff61ad22c2d</orderNumber>
4   <price>159</price>
5   <flights>
6     <flightNumber>pegasus</flightNumber>
7     <price>159</price>
8     <from>Istanbul</from>
9     <to>Izmir</to>
10    <departureTime>2025-03-22-19</departureTime>
11    <arrivalTime>2025-03-22-19</arrivalTime>
12    <description>EE</description>
13    <plane>
14      <manufacturer>II</manufacturer>
15      <model>HH</model>
16      <yearOfManufacture>199</yearOfManufacture>
17    </plane>
18    <seatsLeft>9</seatsLeft>
19    <capacity>30</capacity>
20  </flights>
21  <passenger>
22  </passenger>
```

Yukarıdaki ve aşağıdaki iki örneği karşılaştıracak olursak;

Burada **capacity** ile **seat_left** değerleri kontrol edilip capacity her %10 azaldığında bilet fiyatı(**price**) da %10 artacak şekilde düzenlendi.



```
POST http://localhost:8000/reservation?passengerId=6f221009-dba0-45ed-ab9b-7058e5cb0f2d&flightLists=pegasu... Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

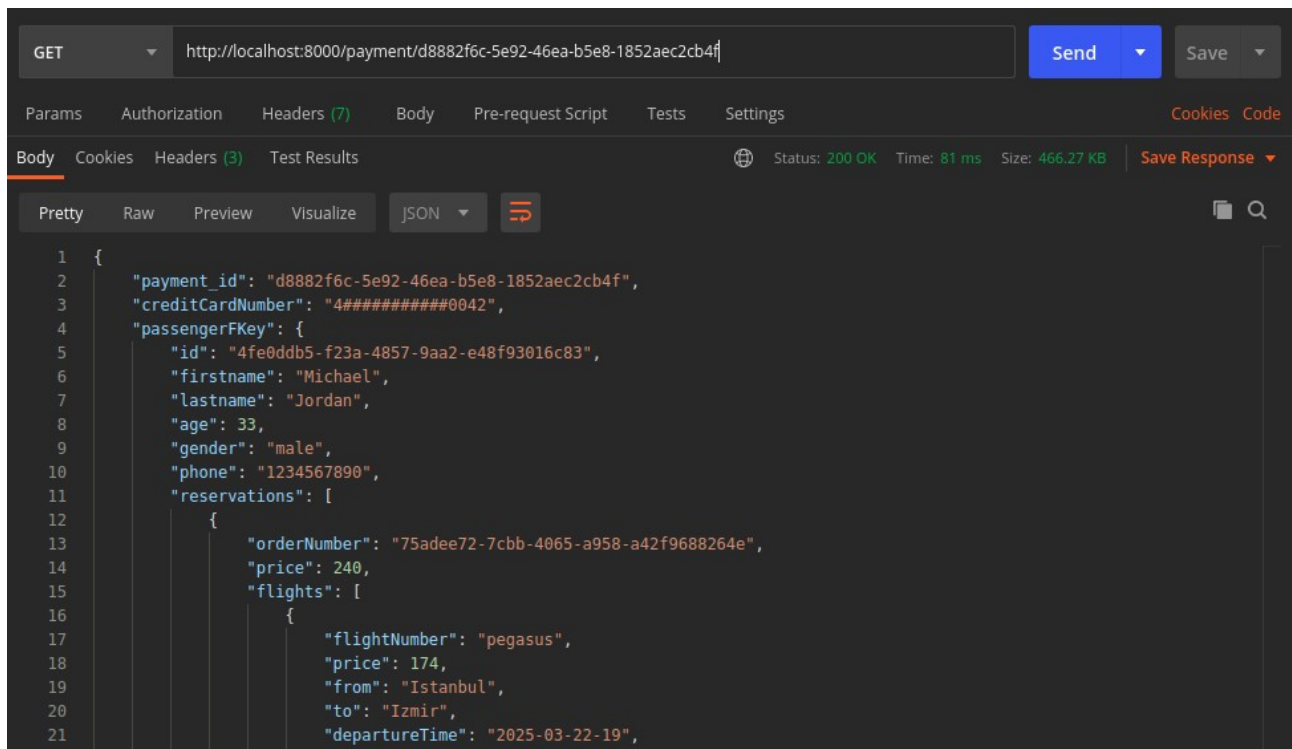
Body Cookies Headers (3) Test Results Status: 200 OK Time: 86 ms Size: 832 B Save Response

Pretty Raw Preview Visualize XML

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <reservation>
3   <orderNumber>1025e800-1362-49cd-a975-78fc6c51cd02</orderNumber>
4   <price>174</price>
5   <flights>
6     <flightNumber>pegasus</flightNumber>
7     <price>174</price>
8     <from>Istanbul</from>
9     <to>Izmir</to>
10    <departureTime>2025-03-22-19</departureTime>
11    <arrivalTime>2025-03-22-19</arrivalTime>
12    <description>EE</description>
13    <plane>
14      <manufacturer>II</manufacturer>
15      <model>HH</model>
16      <yearOfManufacture>199</yearOfManufacture>
17    </plane>
18    <seatsLeft>8</seatsLeft>
19    <capacity>30</capacity>
20  </flights>
21  <passenger>
22  </passenger>
```


12) Get Payment:

URL: <http://localhost:8000/payment/d8882f6c-5e92-46ea-b5e8-1852aec2cb4f>



GET <http://localhost:8000/payment/d8882f6c-5e92-46ea-b5e8-1852aec2cb4f> Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Body Cookies Headers (3) Test Results Status: 200 OK Time: 81 ms Size: 466.27 KB Save Response

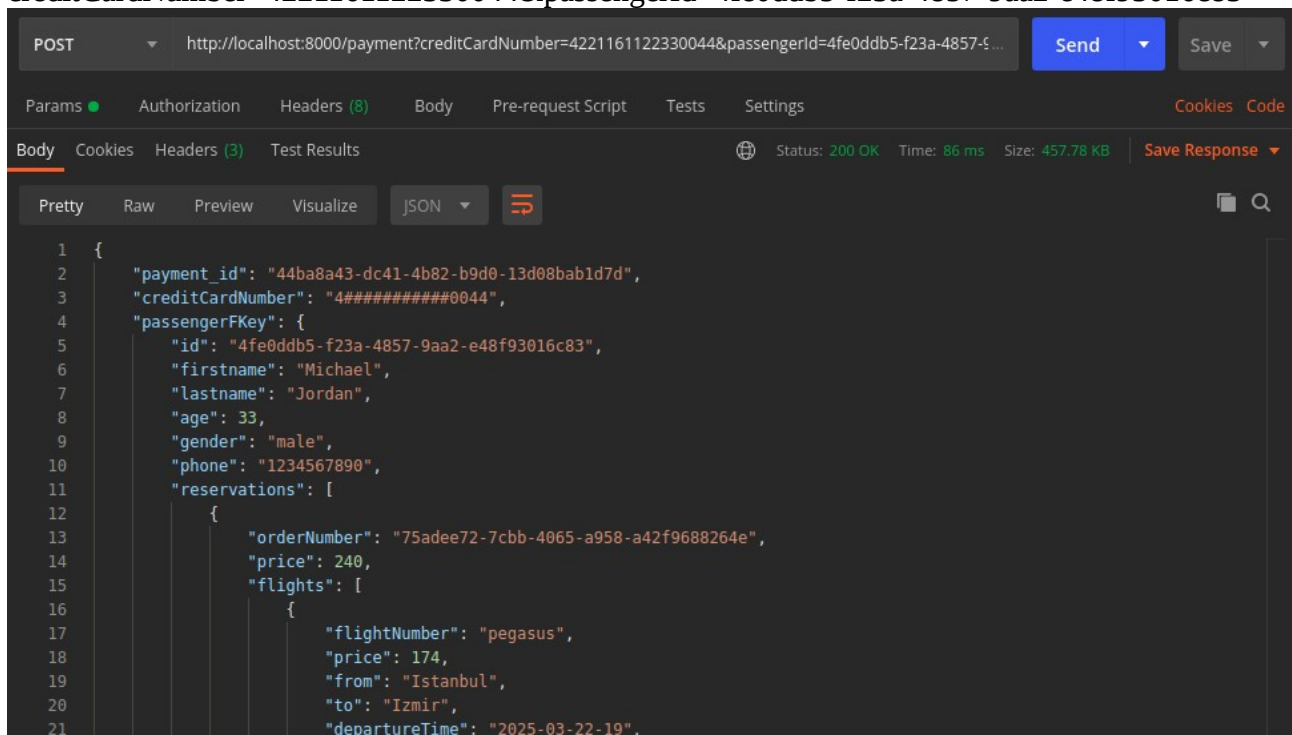
Pretty Raw Preview Visualize JSON

```
1 {
2   "payment_id": "d8882f6c-5e92-46ea-b5e8-1852aec2cb4f",
3   "creditCardNumber": "4#####0042",
4   "passengerFKey": {
5     "id": "4fe0ddb5-f23a-4857-9aa2-e48f93016c83",
6     "firstname": "Michael",
7     "lastname": "Jordan",
8     "age": 33,
9     "gender": "male",
10    "phone": "1234567890",
11    "reservations": [
12      {
13        "orderNumber": "75adee72-7cbb-4065-a958-a42f9688264e",
14        "price": 240,
15        "flights": [
16          {
17            "flightNumber": "pegasus",
18            "price": 174,
19            "from": "Istanbul",
20            "to": "Izmir",
21            "departureTime": "2025-03-22-19",
```

13) Create Payment Method:

URL: <http://localhost:8000/payment?>

[creditCardNumber=4221161122330044&passengerId=4fe0ddb5-f23a-4857-9aa2-e48f93016c83](http://localhost:8000/payment?creditCardNumber=4221161122330044&passengerId=4fe0ddb5-f23a-4857-9aa2-e48f93016c83)



POST <http://localhost:8000/payment?creditCardNumber=4221161122330044&passengerId=4fe0ddb5-f23a-4857-9aa2-e48f93016c83> Send Save

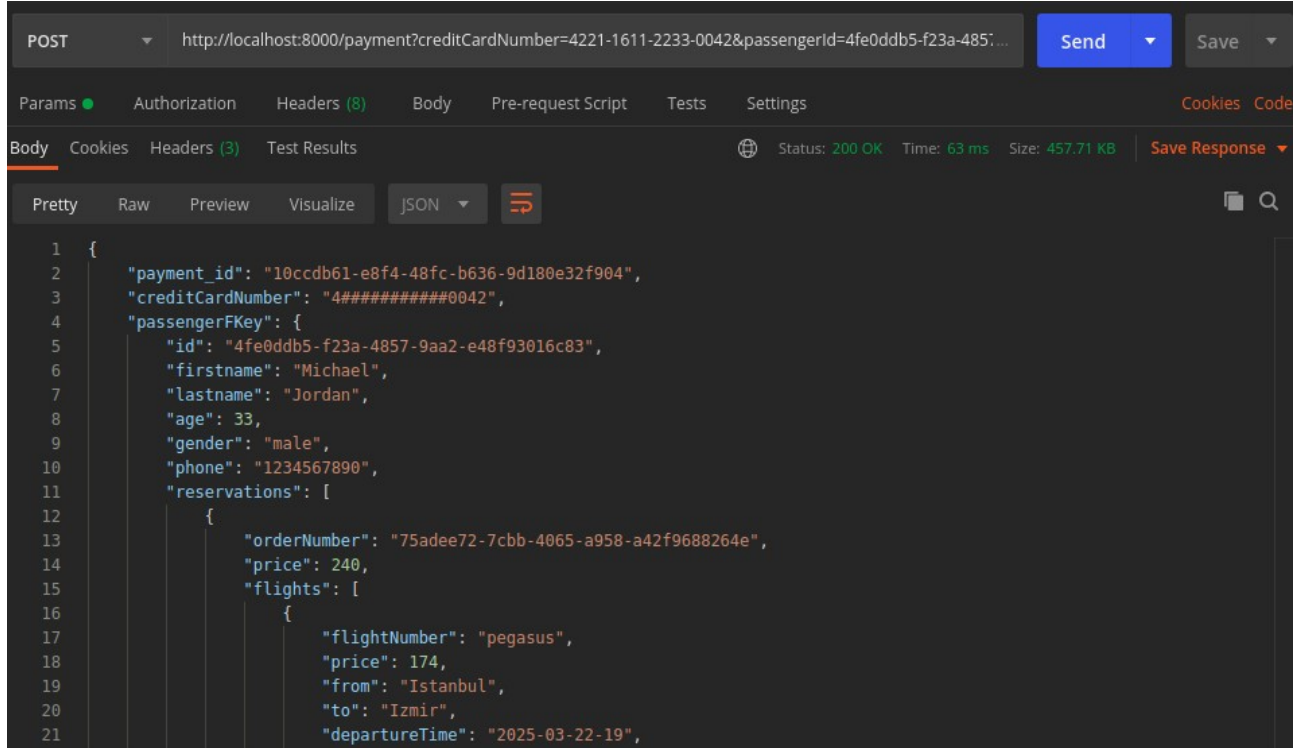
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Body Cookies Headers (3) Test Results Status: 200 OK Time: 86 ms Size: 457.78 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "payment_id": "44ba8a43-dc41-4b82-b9d0-13d08bab1d7d",
3   "creditCardNumber": "4#####0044",
4   "passengerFKey": {
5     "id": "4fe0ddb5-f23a-4857-9aa2-e48f93016c83",
6     "firstname": "Michael",
7     "lastname": "Jordan",
8     "age": 33,
9     "gender": "male",
10    "phone": "1234567890",
11    "reservations": [
12      {
13        "orderNumber": "75adee72-7cbb-4065-a958-a42f9688264e",
14        "price": 240,
15        "flights": [
16          {
17            "flightNumber": "pegasus",
18            "price": 174,
19            "from": "Istanbul",
20            "to": "Izmir",
21            "departureTime": "2025-03-22-19",
```

Aşağıdaki örnekte farklı bir formatta gelen credit card bilgisi beklenen şekilde maskelenmiştir.



The screenshot shows a REST client interface with a POST request to `http://localhost:8000/payment?creditCardNumber=4221-1611-2233-0042&passengerId=4fe0ddb5-f23a-4851-...`. The response is a JSON object with the following structure:

```
1 {
2   "payment_id": "10ccdb61-e8f4-48fc-b636-9d180e32f904",
3   "creditCardNumber": "4#####0042",
4   "passengerFKey": {
5     "id": "4fe0ddb5-f23a-4857-9aa2-e48f93016c83",
6     "firstname": "Michael",
7     "lastname": "Jordan",
8     "age": 33,
9     "gender": "male",
10    "phone": "1234567890",
11    "reservations": [
12      {
13        "orderNumber": "75adee72-7cbb-4065-a958-a42f9688264e",
14        "price": 240,
15        "flights": [
16          {
17            "flightNumber": "pegasus",
18            "price": 174,
19            "from": "Istanbul",
20            "to": "Izmir",
21            "departureTime": "2025-03-22-19",
```