SOT

**ALI HAMZA,3056325**

**All Passengers**

GET - flights/passengers

Content-type: application/json

A GET request has been sent to fetch all the passengers.

The request that was sent had status 200. If we have status 200, that means it was successful and we are going to get the results, we have asked.

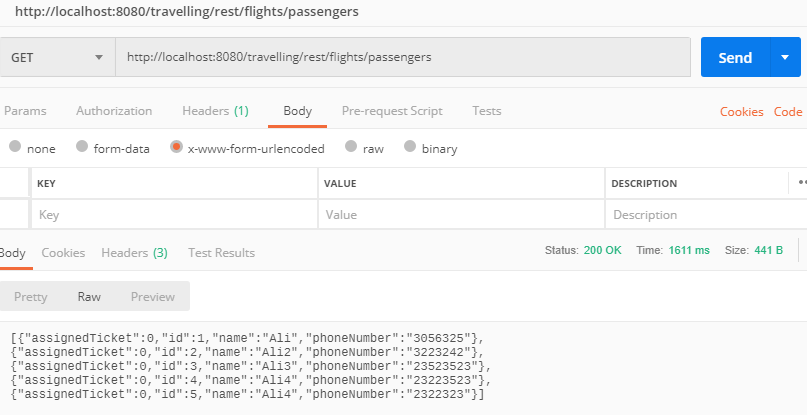
HTTP/1.1 200

**Content-Type →**application/json

**Content-Length →**332

**Date →**Wed, 25 Sep 2019 12:51:25 GMT

**Results**



**All tickets**

GET - flights/tickets

Content-type: application/json

A GET request has been sent to fetch all tickets.

The request that was sent had status 200. If we have status 200, that means it was successful and we are going to get the results, we have asked.

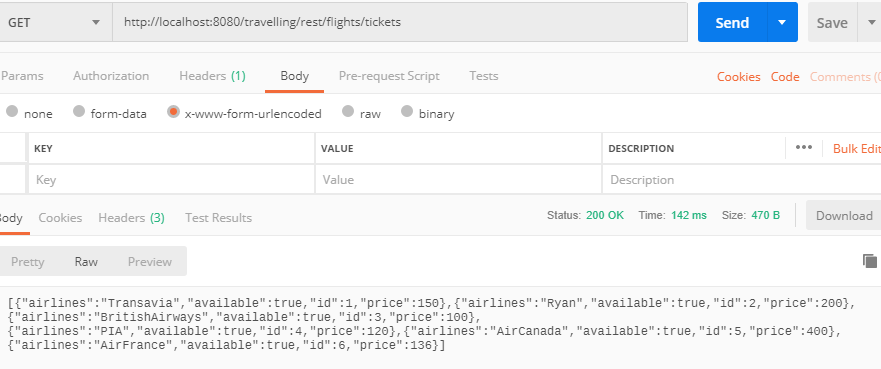
HTTP/1.1 200

**Content-Type →**application/json

**Content-Length →**361

**Date →**Wed, 25 Sep 2019 13:12:25 GMT

**Results**



**Get tickets by airline**

GET /tickets?airline=Transavia HTTP/1.1

Content-type: application/x-www-form-urlencoded

A GET request with a parameter **airline** where the name of the requested airline is passed in the URL, and the server handles it by binding it to the parameter of the endpoint login

HTTP/1.1 200

**POST Ticket**

POST /travelling /rest/flights /create HTTP/1.1

Host: localhost:8080

Content-Type: application/x-www-form-urlencoded

A POST method that sends data through the requests body as name and value pairs (JSON). The POST is used to create a new ticket, by taking a id and airline as body parameters.

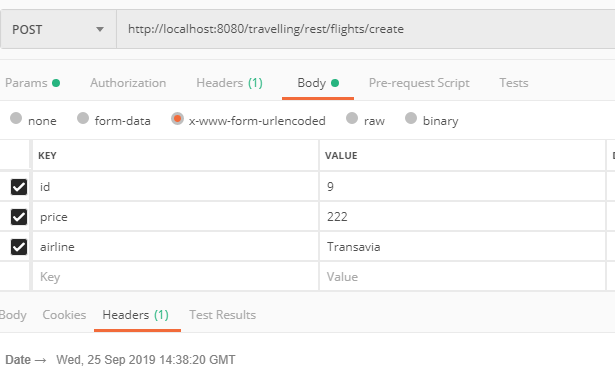
The entity of http request are three form parameters. The http response message is 204 No-Content because the entity of the response is empty for POST.

HTTP/1.1 204

Content-type: application/x-www-form-urlencoded

**Date →**Wed, 25 Sep 2019 14:38:20 GMT

**Result**



**Delete Passenger**

DELETE /school/rest/flights /passenger /2 HTTP/1.1

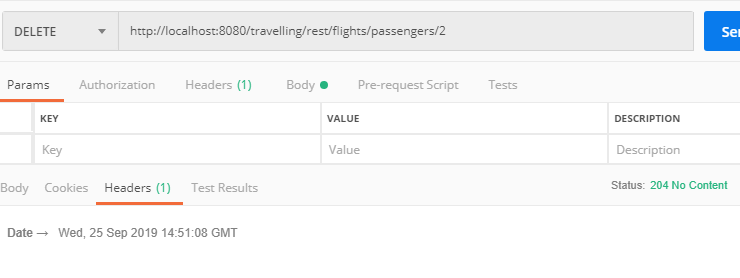
Host: localhost:8080

A DELETE request that takes an ID as a parameter. The logic of the endpoints takes that parameter, looks up the passenger that is requested and deletes it from memory. In this case the request will be sent to /passenger /2.

HTTP/1.1 204

Content-Type: No-Content

**Date →**Wed, 25 Sep 2019 14:51:08 GMT



**Delete Ticket**

DELETE /school/rest/flights /passenger /2 HTTP/1.1

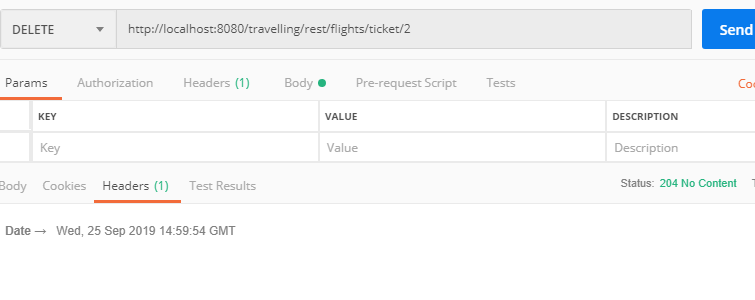
Host: localhost:8080

A DELETE request that takes an ID as a parameter. The logic of the endpoints takes that parameter, looks up the ticket that is requested and deletes it from memory. In this case the request will be sent to /ticket/2 .

The http response message for DELETE operation is always 204 NO-CONTENT.

HTTP/1.1 204

Content-Type: No-Content



**Reserve a ticket for passenger**

PUT /travelling /rest/flights/reserve /2/tickets/2 HTTP/1.1

Host: localhost:8080

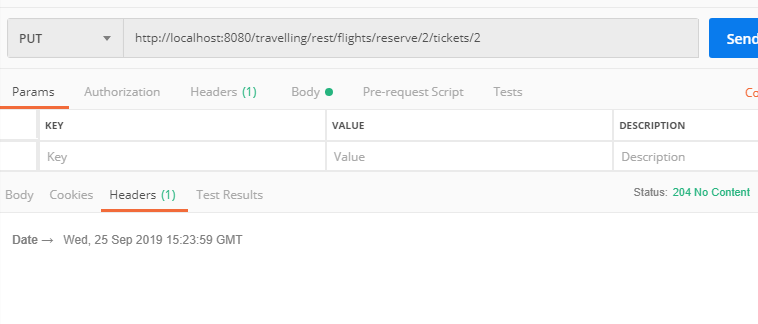
The PUT is used to create update ticket status, by taking a ticket id and passenger id. The entity of http request are three form parameters. The http response message is 204 No-Content because the entity of the response is empty for PUT.

**Result**

HTTP/1.1 204

Content-Type: No-Content

**Date →**Wed, 25 Sep 2019 15:23:59 GMT



we can also the console messages of the service for the different CRUD operations. For example, when ticket is created it shows message “ticket created” or if the PUT operation is perform it shows console message “ticket reserved”.

