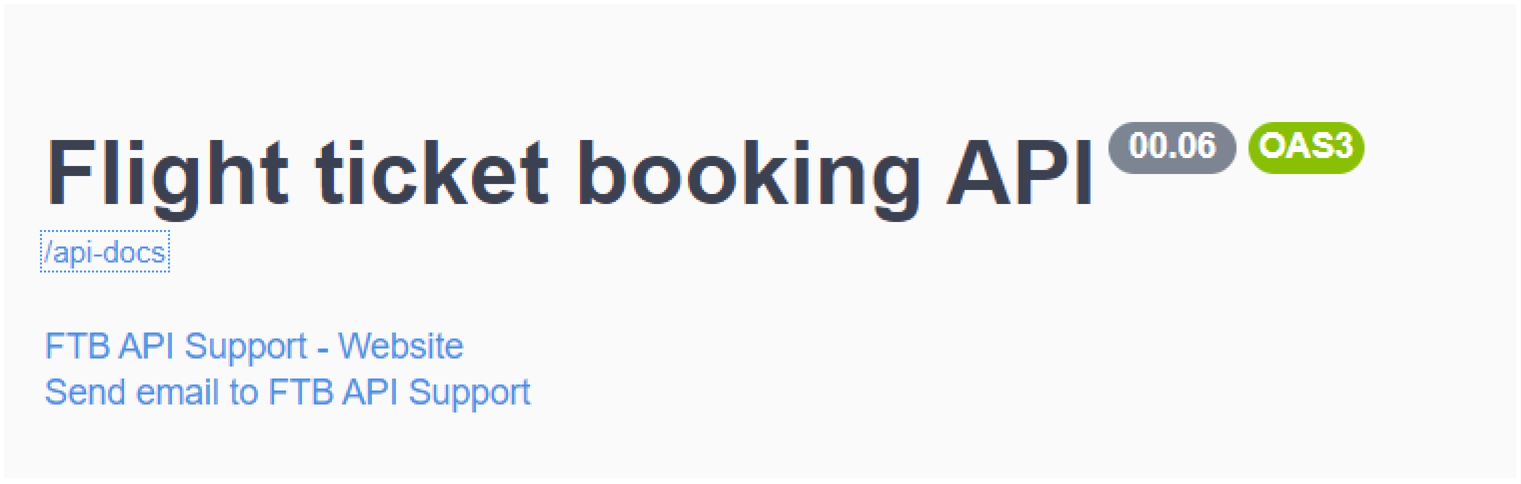
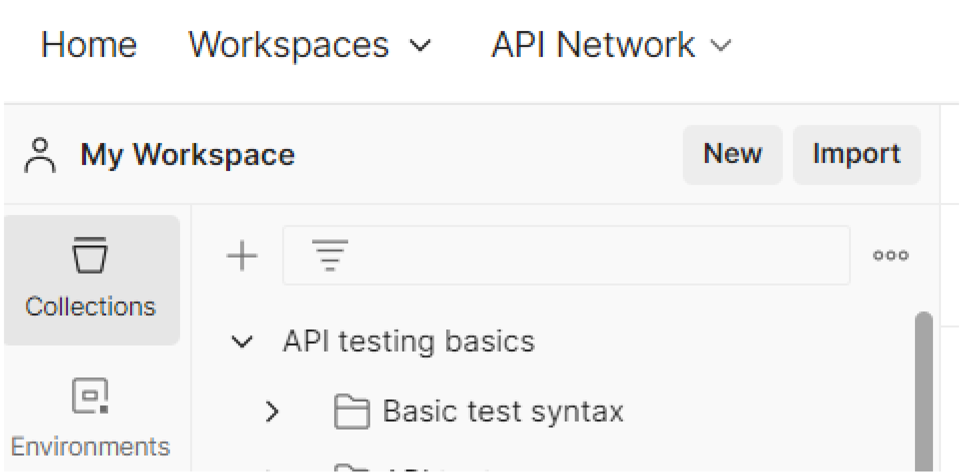
Papara 10.

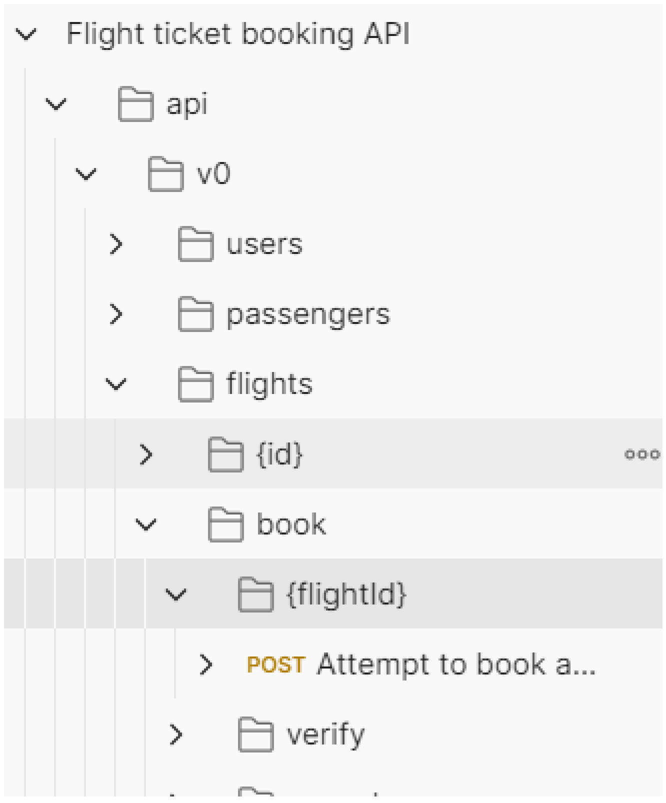
In order to import all FTB documentation from Swagger into Postman we should click to the link /api-docs at the very top of the page.



Flight ticket booking API
00.06
OAS3
/api-docs
FTB API Support - Website
Send email to FTB API Support

The page /server_ip/api-docs will open.  After that, copy all information from the page in notepad or other text reader and save as json. And after that in Postman click the bottom import, select the file that you saved before and all methods will be added to the new collection.





**Task**

- **Fully cover POST /api/v0/flights/book/{flightId} with tests, according to the textbook recommendations**
- **Run all the tests in Postman**
- **In the HW, report the results of each test, in the table:**

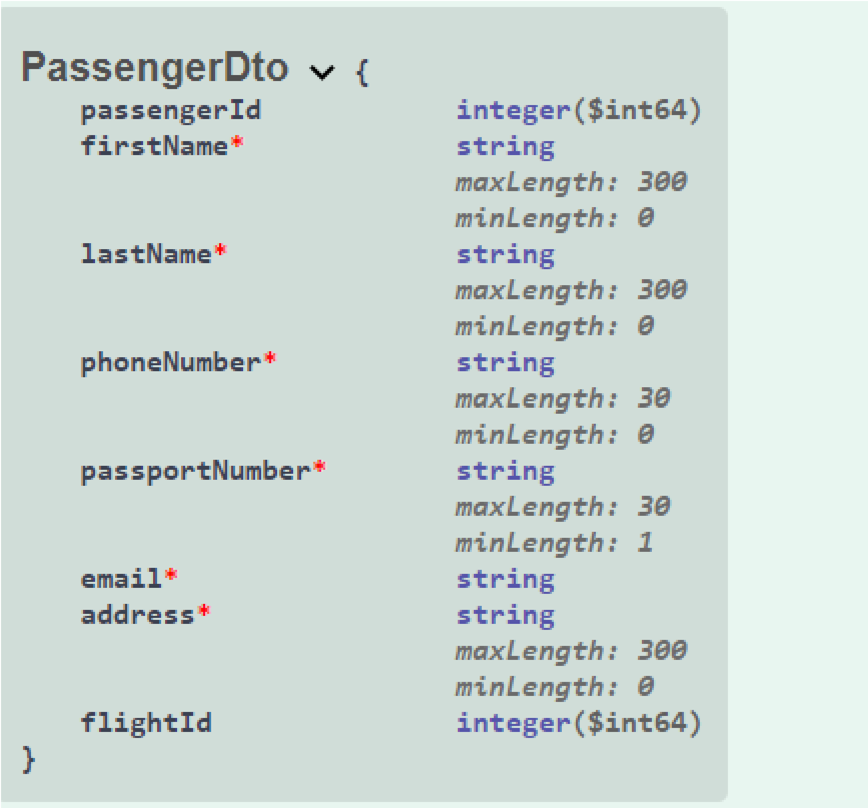
| | What was checked | What was the result | Was this expected and why |
|---|---|---|---|
| 1 | we can take real flightid and passengerid from real DB | 200 OK | all is work . It is expected result . Passengerid is autoincrement value. We can change only flightid value. We can see new strings in DB |
| 2 | we need to check wich fields must be not blank. We can do it with empty json too | We can see that all fields must not be blank 400 BAD_REQUEST ","message":"Validation Failed","details":["firstName \| must not be blank","email \| must not be blank","passportNumber \| must not be blank","lastName \| must not be blank","phoneNumber \| must not be blank","address \| must not be blank","passportNumber \| size must be between 1 and 30"]} | It is the expected result. We can see that all fields are important and must not be blank .It is right that all fields must be not empty because all this info is important. We also need to see real requirements |
| 3 | whether it is possible to post data with empty body | "status": "400 BAD REQUEST". It is not possible | It is expected result. We can not post empty body without data. We need to have some checks for this |
| 4 | whether it is possible to post data without any flightid | 405 Method not allowed. It is not possible | It is expected result, because flightid is a important key |
| 5 | whether it is possible to post the same data with the same flightid | "status": "400 BAD REQUEST". Unfortunately human clones are not allowed on flight | It was expected because if we would have the same person more than 1 time for the same flight it would be a bug |
| 6 | we can check what will be if we post with empty parameters in each field | We found a bug, because we see in API doc that it is allowed to print data from 0 to 300 symbols in firstname, last name, address fields. It is a mistake, because the app does not do it. In DB it is varchar variable with 255 symbols. we need to see the requirements | It is expected result. We can see that all fields are important and must be not blank. We can see in details which parameters can have which info. For example,interesting info about passport number |
| 7 | we can check passport number with 1 digit | 200 Ok. All works | It is the expected result according our API docs. All works with 1 digit. But we can check passport number in requirements, because it strange that the size can have 1 digit It is not right restriction |
| 8 | we can check passport number with 30 digits | 200 Ok. All works | It is expected result. All works with 30 digit in passport number |

|  |  |  |  |
|---|---|---|---|
| 9 | we can check passport number with 31 digits | with 31 digits in passport number it does not work "400 BAD, BAD REQUEST",<br>"message": "Validation Failed",<br>"details": [<br>"passportNumber \| size must be between 1 and 30" | It is the expected result, because we can see from the message about 30 digits and API docs shows us the same result. With 31 digits in passport number it does not work. But we need to check requirements about the right size |
| 10 | What will be if we take letters instead of digits in phone numbers | 200 Ok | It is unexpected result. The application should show an error message because there should be numbers here |
| 11 | wether we can take email address without @ | "400 BAD_REQUEST - BAD, BAD REQUEST",<br>"message": "Validation Failed",<br>"details": [<br>"email \| must be a well-formed email address"<br>] | It is expected result, because email should have @ sign |
| 12 | put sign @ in the beginning or in the end of email address | 400 BAD_REQUEST - BAD, BAD REQUEST","message":"Validation Failed","details":["email \| must be a well-formed email address"] | It is expected result. Sign @ can not be in the beginning or in the end of email address.<br><br>the email field must have some restrictions about the length too. In API docs we do not see any. It has some but it is not the same like it is written in DB |
| 13 | if we take only digits with @ in the middle of email | 200 ok | It is a bag, because real email must have not only digits and @ |
| 14 | email minlength can be r@r, or 1@1 | 200 ok | we can not check min lenght of email correctly, because we do not see it in API docs. From test we see that this is the min length that app let us post for email in API docs |
| 15 | What will be if we post with 255 symbols for email | 400 BAD_REQUEST - BAD,details":["email \| must be a well-formed email address"]} | It is a bug. We tried to check what maxlength has email. We need to see some message about max length , too many symbols in the email address. We do not have any min and max length in API docs for email addresses. We only see from DB varchar(255) for this field |
| 16 | we can check address from only digits | 200 Ok | It should be a bug, because address can not consist from only digits |
| 17 | We can check address from only letters | 200 Ok | It should be a bug, because address can not consist from only letters, it should have numbers too |
| 18 | We can check whether passport number can be from only letters | 200 Ok | It should be a bug, because passport number can not consist from only letters, it should have numbers too or only numbers it depends on country |

| | | | |
|---|---|---|---|
| | | | and requirements |
| 19 | minimum value for phone number 0 | 200 ok | It is expected result |
| 20 | maximum value for phone number 30 | 200 ok | It is expected result |
| 21 | if we print more than maximum value for phone number , for example 31 digits | 400 BAD_REQUEST - BAD, BAD REQUEST","message":"Validation Failed","details":["phoneNumber \| size must be between 0 and 30"]} | It is expected result, because the app says that the size of phone number must be between 0 and 30.But we need to check the right size in requirements |
| 22 | try to print 300 symbols for firstname , lastname and address | "400 BAD_REQUEST","message":"could not execute statement; SQL [n/a]; nested exception is org.hibernate.exception.DataException: could not execute statement","details":["could not execute statement; SQL [n/a]; nested exception is org.hibernate.exception.DataException: could not execute statement","could not execute statement","Data truncation: Data too long for column | we see in API doc that it is allowed to print data from 0 to 300 symbols in firstname, last name, address fields. It is a mistake, because the app does not do it. In DB it is varchar variable with 255 symbols |
| 23 | try to print 255 symbols for firstname , lastname and address | 400 BAD REQUEST the same result Data too long for column | we see some mistakes, because the number of characters does not match the database and the documentation in the API |
| 24 | If we take 0 for minlenghth for firstname, lastname, phonenumber, address | 400 BAD_REQUEST | It is not right according to API docs, because in API docs we can see 0 as minlength for firstname, last name, phone number, address. It is a documentation inconsistency. We need to see requirements. Because the condition for min length must be in these fields, but not 0 |
| | We need to check all info in requirements | | |

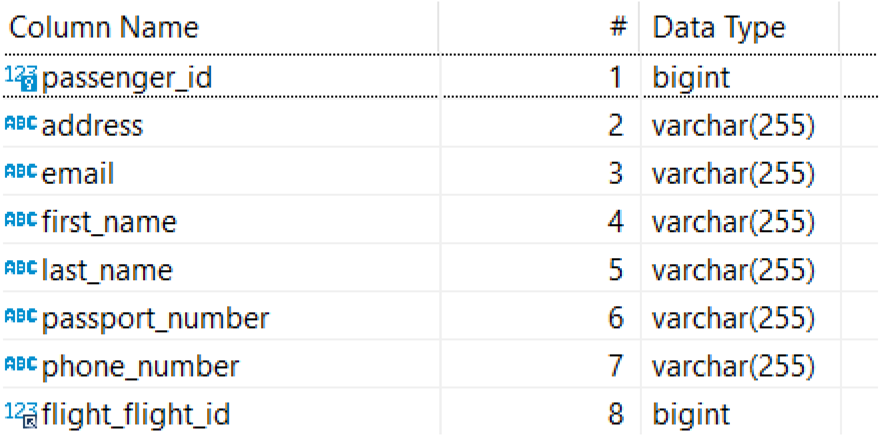
We can see API docs in swagger. It shows us min and max size and variable type for all fields. It looks strange that we could have so many symbols in names and addresses. We can see that in these fields all sizes must be not null, it means that the min length at least must be more than 0

In API docs there is no info about email min and max size



```
PassengerDto ⌄ {
    passengerId          integer($int64)
    firstName*           string
                         maxLength: 300
                         minLength: 0
    lastName*            string
                         maxLength: 300
                         minLength: 0
    phoneNumber*         string
                         maxLength: 30
                         minLength: 0
    passportNumber*      string
                         maxLength: 30
                         minLength: 1
    email*               string
    address*             string
                         maxLength: 300
                         minLength: 0
    flightId             integer($int64)
}
```

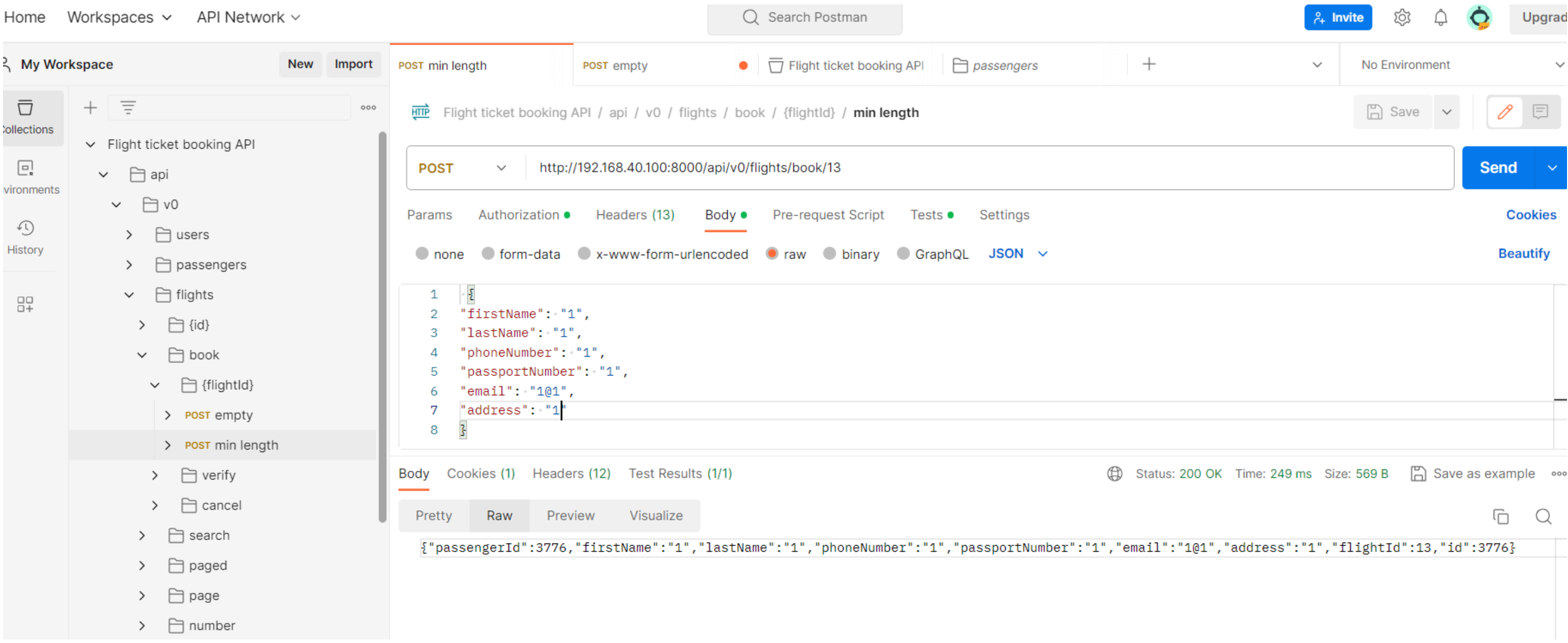
We also could see info from DB that max size of varchar is 255 symbols



| Column Name | # | Data Type |
|---|---|---|
| 123 passenger_id | 1 | bigint |
| ABC address | 2 | varchar(255) |
| ABC email | 3 | varchar(255) |
| ABC first_name | 4 | varchar(255) |
| ABC last_name | 5 | varchar(255) |
| ABC passport_number | 6 | varchar(255) |
| ABC phone_number | 7 | varchar(255) |
| 123 flight_flight_id | 8 | bigint |

Snippets in Postman

We can check min length with snippets. All is ok 200 status



Flight ticket booking API / api / v0 / flights / book / {flightId} / min length

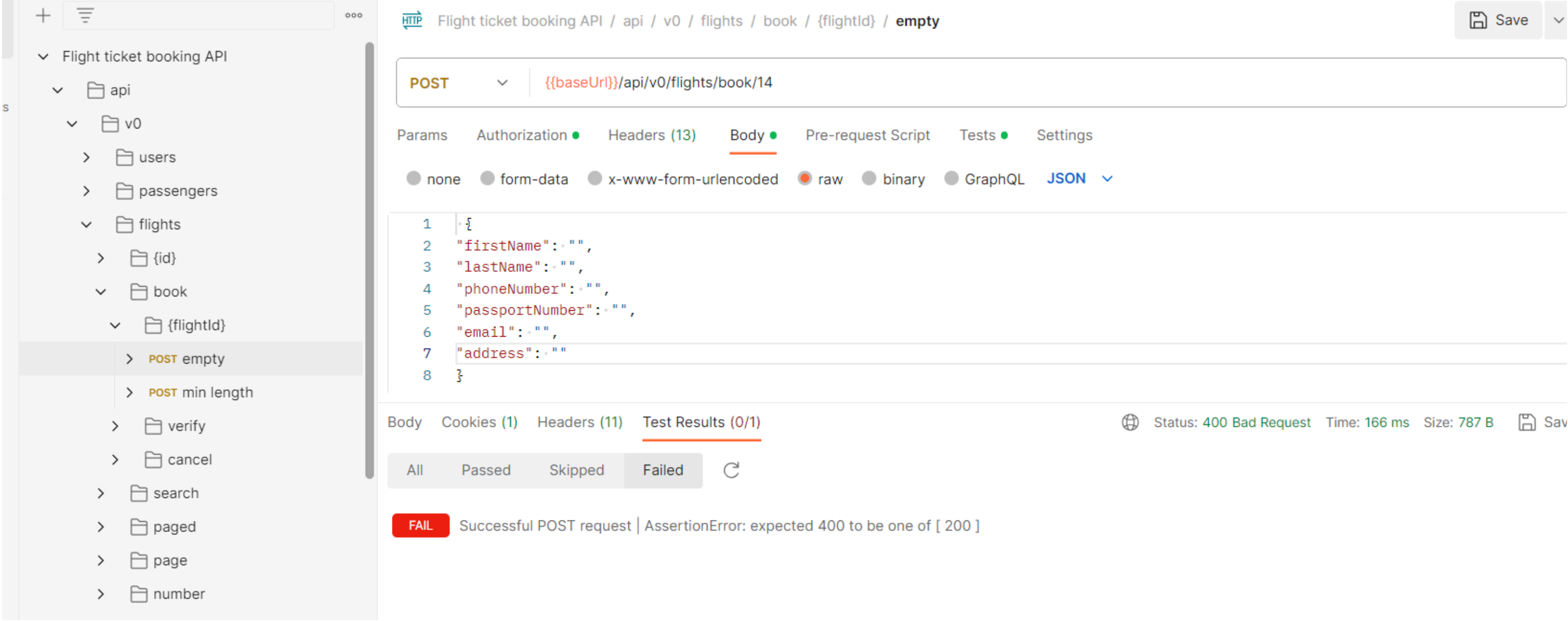
POST http://192.168.40.100:8000/api/v0/flights/book/13

```
{
"firstName": "1",
"lastName": "1",
"phoneNumber": "1",
"passportNumber": "1",
"email": "1@1",
"address": "1"
}
```

Status: 200 OK  Time: 249 ms  Size: 569 B

```
{"passengerId":3776,"firstName":"1","lastName":"1","phoneNumber":"1","passportNumber":"1","email":"1@1","address":"1","flightId":13,"id":3776}
```

We can check with snippets empty fields in post with snippets. We have 400 status , it is right, because we can not post empty fields



Flight ticket booking API / api / v0 / flights / book / {flightId} / empty

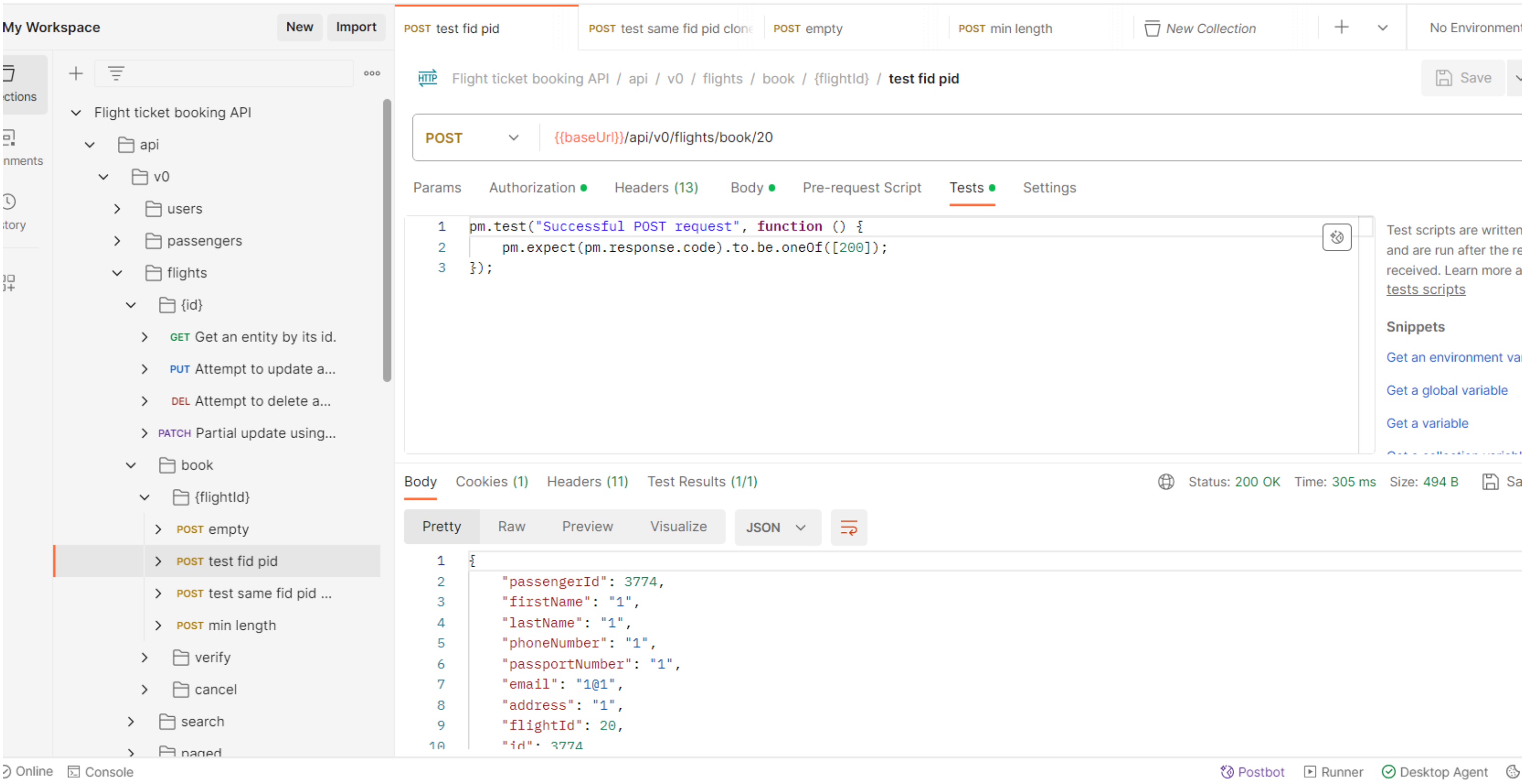
POST {{baseUrl}}/api/v0/flights/book/14

```
{
"firstName": "",
"lastName": "",
"phoneNumber": "",
"passportNumber": "",
"email": "",
"address": ""
}
```

Status: 400 Bad Request Time: 166 ms Size: 787 B

FAIL Successful POST request | AssertionError: expected 400 to be one of [ 200 ]

We can check with snippets adding info with right data, in DB all data was added successfully



Flight ticket booking API / api / v0 / flights / book / {flightId} / test fid pid

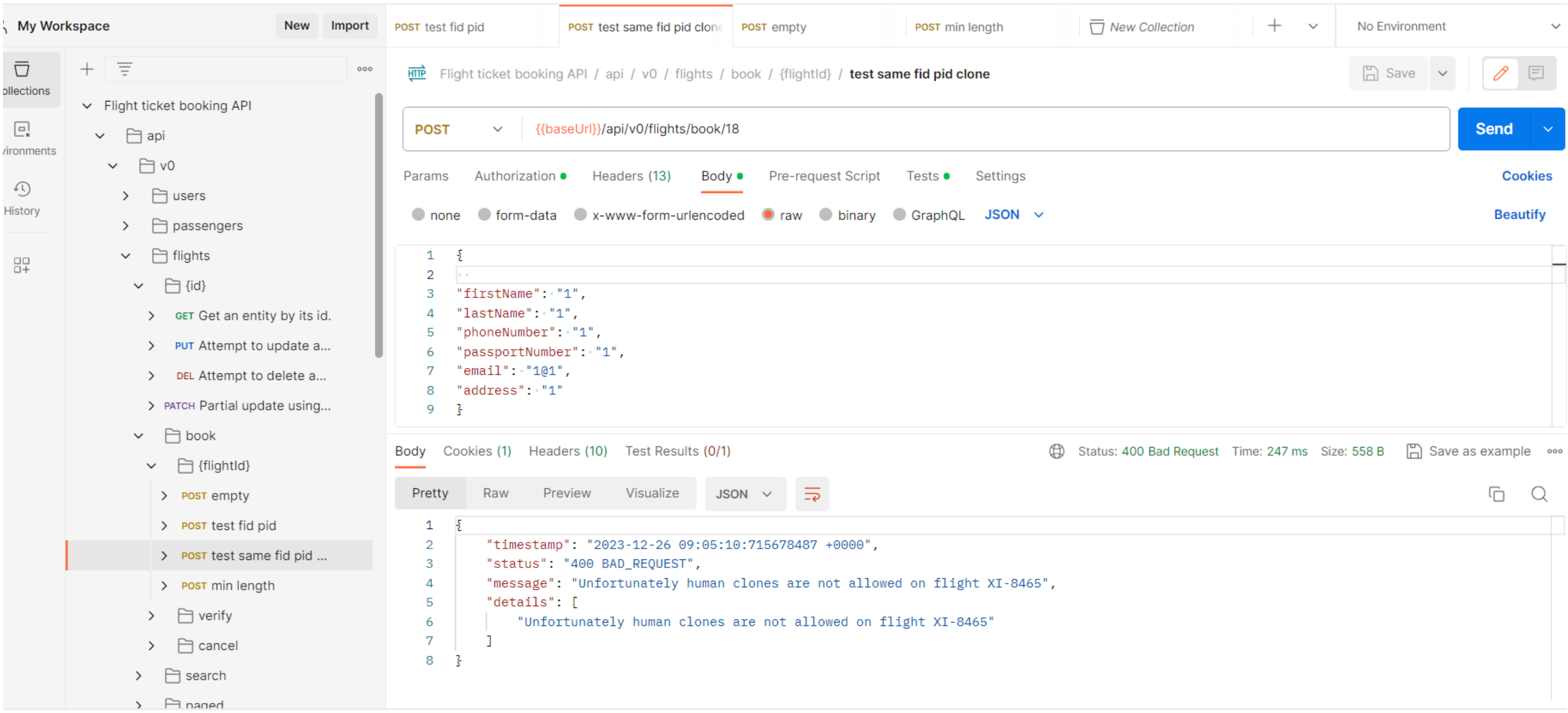
POST {{baseUrl}}/api/v0/flights/book/20

```
pm.test("Successful POST request", function () {
    pm.expect(pm.response.code).to.be.oneOf([200]);
});
```

Status: 200 OK Time: 305 ms Size: 494 B

```
{
    "passengerId": 3774,
    "firstName": "1",
    "lastName": "1",
    "phoneNumber": "1",
    "passportNumber": "1",
    "email": "1@1",
    "address": "1",
    "flightId": 20,
    "id": 3774
```

We can check with snippets what will be If we try to do a clone



POST {{baseUrl}}/api/v0/flights/book/18

```
{

"firstName": "1",
"lastName": "1",
"phoneNumber": "1",
"passportNumber": "1",
"email": "1@1",
"address": "1"
}
```

```
{
    "timestamp": "2023-12-26 09:05:10:715678487 +0000",
    "status": "400 BAD_REQUEST",
    "message": "Unfortunately human clones are not allowed on flight XI-8465",
    "details": [
        "Unfortunately human clones are not allowed on flight XI-8465"
    ]
}
```