

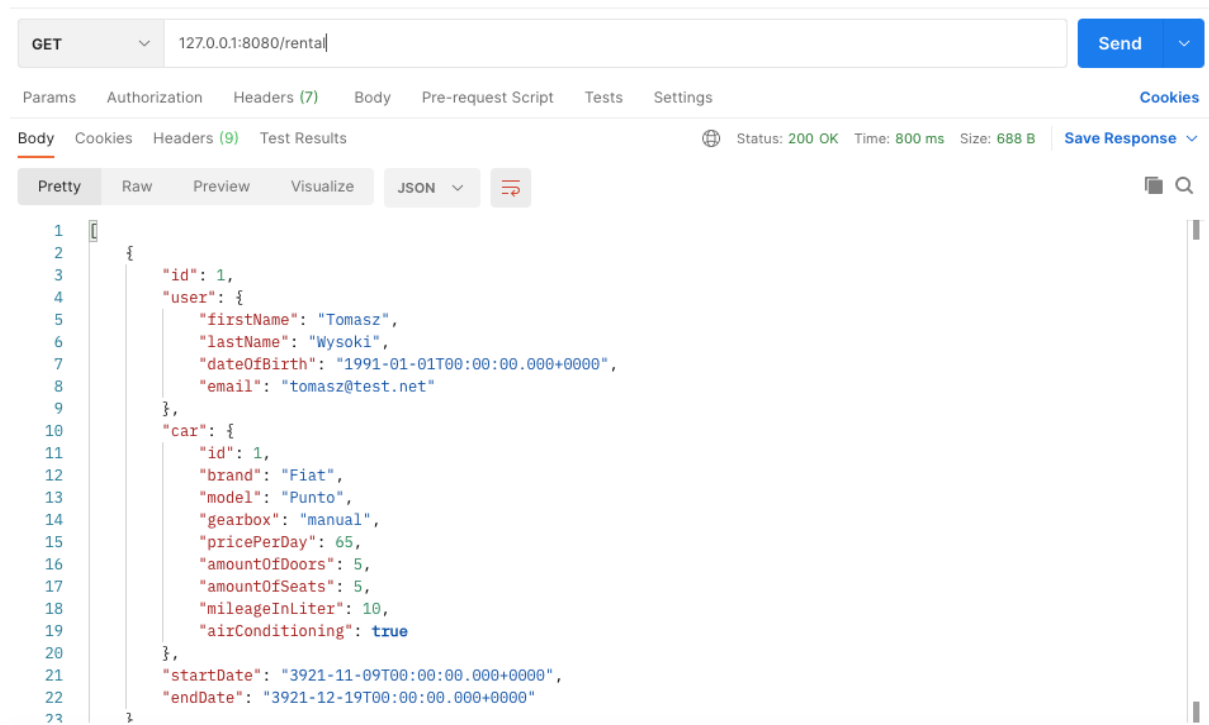
Raport z testów „CarRent” części backend.

Testy ręczne:

CONTROLLER RENTAL

GET /rental

Autentykacja jest wymagana. Request zwraca listę wszystkich rezerwacji, co jest żądanym zachowaniem.



GET 127.0.0.1:8080/rental Send

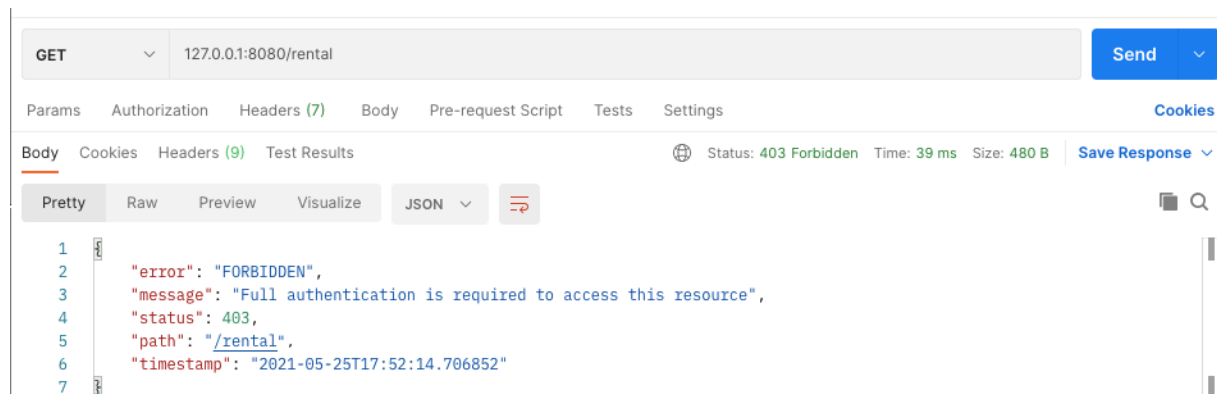
Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (9) Test Results Status: 200 OK Time: 800 ms Size: 688 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "user": {
5       "firstName": "Tomasz",
6       "lastName": "Wysoki",
7       "dateOfBirth": "1991-01-01T00:00:00.000+0000",
8       "email": "tomasz@test.net"
9     },
10    "car": {
11      "id": 1,
12      "brand": "Fiat",
13      "model": "Punto",
14      "gearbox": "manual",
15      "pricePerDay": 65,
16      "amountOfDoors": 5,
17      "amountOfSeats": 5,
18      "mileageInLiter": 10,
19      "airConditioning": true
20    },
21    "startDate": "3921-11-09T00:00:00.000+0000",
22    "endDate": "3921-12-19T00:00:00.000+0000"
23  }
```

W przypadku braku autentykacji otrzymujemy stosowny kod odpowiedzi.



GET 127.0.0.1:8080/rental Send

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

Body Cookies Headers (9) Test Results Status: 403 Forbidden Time: 39 ms Size: 480 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "error": "FORBIDDEN",
3   "message": "Full authentication is required to access this resource",
4   "status": 403,
5   "path": "/rental",
6   "timestamp": "2021-05-25T17:52:14.706852"
7 }
```

GET /rental/{id}

Wymaga autentykacji, w przypadku braku, zwraca stosowny status „403”. Co jest zachowaniem pożądanym.

Po zalogowaniu otrzymujemy dane dotyczące rezerwacji o id=1:

GET 127.0.0.1:8080/rental/1 Send

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

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1327 ms Size: 686 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "user": {
4     "firstName": "Tomasz",
5     "lastName": "Wysoki",
6     "dateOfBirth": "1991-01-01T00:00:00.000+0000",
7     "email": "tomasz@test.net"
8   },
9   "car": {
10    "id": 1,
11    "brand": "Fiat",
12    "model": "Punto",
13    "gearbox": "manual",
14    "pricePerDay": 65,
15    "amountOfDoors": 5,
16    "amountOfSeats": 5,
17    "mileageInLiter": 10,
18    "airConditioning": true
19  },
20   "startDate": "3921-11-09T00:00:00.000+0000",
21   "endDate": "3921-12-19T00:00:00.000+0000"
22 }
```

W przypadku podania złego id serwer zwraca stosowny kod:

GET http://127.0.0.1:8080/rental/0 Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (10) Test Results Status: 500 Internal Server Error Time: 262 ms Size: 491 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2021-05-13T16:44:02.494+0000",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "No message available",
6   "path": "/rental/0"
7 }
```

POST/rental

Również wymaga autentykacji, w przypadku braku, zwraca stosowny status „403”. Co jest zachowaniem pożądanym.

Przy poprawnych danych.

POST 127.0.0.1:8080/rental

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": 17,
3   "userId": 2,
4   "carId": 3,
5   "startDate": "2021-02-21T00:00:00.000+0000",
6   "endDate": "2021-03-12T00:00:00.000+0000"
7 }
```

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 262 ms Size: 686 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 29,
3   "user": {
4     "firstName": "Kasia",
5     "lastName": "Dlugonosa",
6     "dateOfBirth": "2000-02-02T00:00:00.000+0000",
7     "email": "kasia@test.net"
8   },
9   "car": {
10    "id": 3,
11    "brand": "audi",
12    "model": "a5",
13    "color": "black"
14  }
15 }
```

przy niepoprawnych danych (end zamiast endDate)

POST 127.0.0.1:8080/rental

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": 17,
3   "userId": 2,
4   "carId": 3,
5   "startDate": "2021-02-21T00:00:00.000+0000",
6   "end": "2021-03-12T00:00:00.000+0000"
7 }
```

Body Cookies Headers (10) Test Results

Status: 400 Bad Request Time: 22 ms Size: 439 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid method arguments",
3   "errors": {
4     "endDate": "must not be null"
5   },
6   "instance": "/rental"
7 }
```

PUT/rental/{id}

Wymaga autentykacji, w przypadku braku, zwraca stosowny status „403”. Co jest zachowaniem pożądanym.

Przy poprawnych danych

PUT 127.0.0.1:8080/rental/17

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": 17,
3   "userId": 1,
4   "carId": 4,
5   "startDate": "2021-05-27T00:00:00.000+0000",
6   "endDate": "2021-05-28T00:00:00.000+0000"
7 }
```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 264 ms Size: 682 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 17,
3   "user": {
4     "firstName": "Tomasz",
5     "lastName": "Wysoki",
6     "dateOfBirth": "1991-01-01T00:00:00.000+0000",
7     "email": "tomasz@test.net"
8   },
9   "car": {
10    "id": 4,
11    "brand": "BMW",
12    "model": "x5",
13    "year": 2018
14  }
15 }
```

Przy błędnych danych

PUT 127.0.0.1:8080/rental/2

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "userId": 1,
3   "carId": 10,
4   "startDate": "2021-05-27T00:00:00.000+0000",
5   "endDate": "2021-05-28T00:00:00.000+0000"
6 }
```

Body Cookies Headers (10) Test Results Status: 500 Internal Server Error Time: 200 ms Size: 491 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2021-05-31T15:46:54.430+0000",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "No message available",
6   "path": "/rental/2"
7 }
```

DELETE rental/{id}

Przy poprawnych danych

DELETE

127.0.0.1:8080/rental/26

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 204 No ContentTime: 238 msSize: 246 BSave Response

PrettyRawPreviewVisualizeText

1

Przy użyciu błędnego id (przy próbie usunięcia nieistniejącej rezerwacji)

DELETE

127.0.0.1:8080/rental/1

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (10)Test Results

Status: 500 Internal Server ErrorTime: 221 msSize: 518 BSave Response

PrettyRawPreviewVisualizeJSON

```
1
2  "timestamp": "2021-05-31T15:54:50.441+0000",
3  "status": 500,
4  "error": "Internal Server Error",
5  "message": "attempt to create delete event with null entity",
6  "path": "/rental/1"
7
```

Powyższy raport jest ostatecznym raportem gotowej aplikacji, testy były wykonywane na bieżąco, a występujące błędy zgłaszane na bieżąco programistom.

Autorzy:
Olga Żalińska
Patryk Żur