

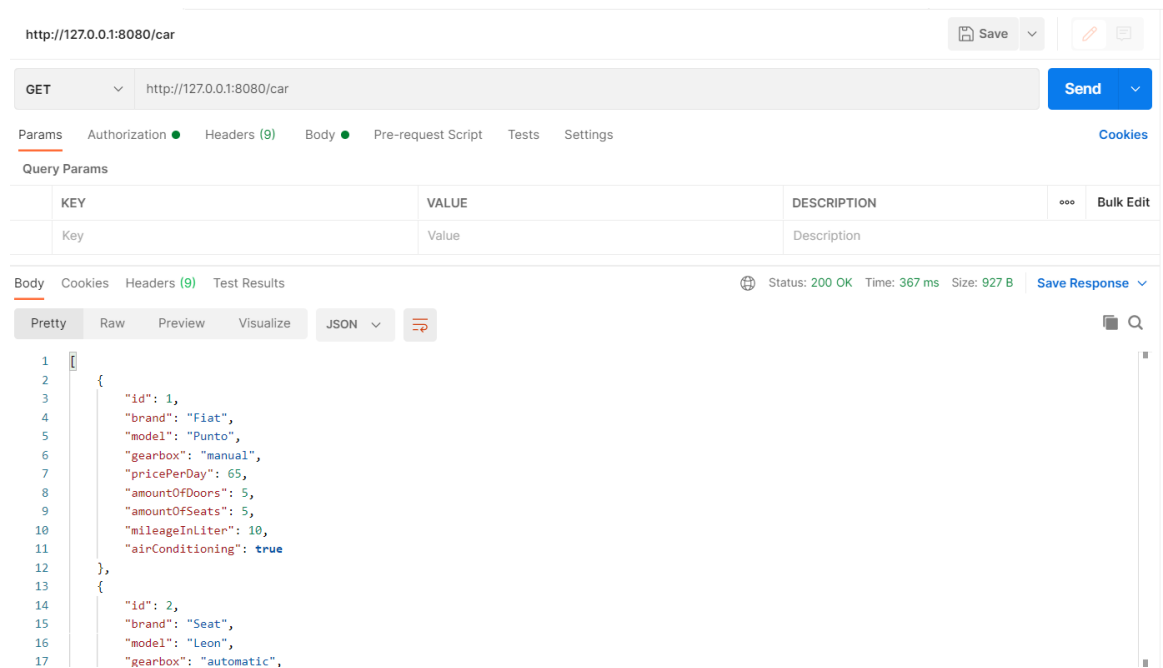
Raport z testów „CarRent” części backend.

Testy ręczne:

CONTROLLER CAR

GET/car

Po wysłaniu requesta /car API zwraca wszystkie samochody z bazy danych, co jest żądanym zachowaniem.



http://127.0.0.1:8080/car

GET http://127.0.0.1:8080/car

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

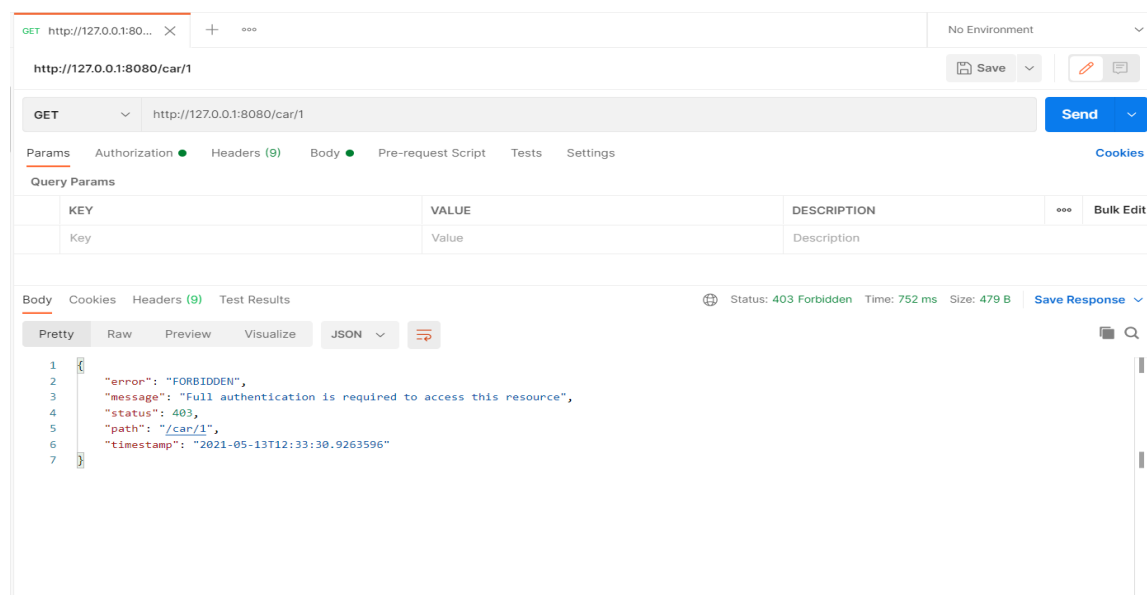
Body Cookies Headers (9) Test Results Status: 200 OK Time: 367 ms Size: 927 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "brand": "Fiat",
5     "model": "Punto",
6     "gearbox": "manual",
7     "pricePerDay": 65,
8     "amountOfDoors": 5,
9     "amountOfSeats": 5,
10    "mileageInLiter": 10,
11    "airConditioning": true
12  },
13  {
14    "id": 2,
15    "brand": "Seat",
16    "model": "Leon",
17    "gearbox": "automatic",
```

GET/car/{id}

W przypadku braku autentykacji (zalogowania) otrzymujemy status „403”. Co jest zachowaniem pożądanym.



GET http://127.0.0.1:8080/car/1

GET http://127.0.0.1:8080/car/1

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 403 Forbidden Time: 752 ms Size: 479 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": "/car/1",
6   "timestamp": "2021-05-13T12:33:30.9263596"
7 }
```

W przypadku zalogowania (header) API zwraca zadany samochód.

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8080/car/1`. The request is saved and ready to be sent. The response status is 200 OK, with a time of 1108 ms and a size of 466 B. The response body is displayed in JSON format, showing details for a car with id 1, brand Fiat, model Punto, manual gearbox, price per day 65, amount of doors 5, amount of seats 5, mileage in liter 10, and air conditioning true.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "id": 1,
3   "brand": "Fiat",
4   "model": "Punto",
5   "gearbox": "manual",
6   "pricePerDay": 65,
7   "amountOfDoors": 5,
8   "amountOfSeats": 5,
9   "mileageInLiter": 10,
10  "airConditioning": true
11 }
```

W przypadku podania nieistniejącego id serwer zwraca stosowny kod.

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8080/car/0`. The request is saved and ready to be sent. The response status is 500 Internal Server Error, with a time of 591 ms and a size of 488 B. The response body is displayed in JSON format, showing an error message: "No message available".

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "timestamp": "2021-05-13T15:50:08.164+0000",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "No message available",
6   "path": "/car/0"
7 }
```

POST /car

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

przy poprawnych danych

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8080/car`. The request body is a JSON object with the following fields: `brand` (FS0), `model` (Polonez), `gearbox` (manual), `pricePerDay` (30), `amountOfDoors` (5), `amountOfSeats` (5), `mileageInLiter` (12), and `airConditioning` (true). The response status is 200 OK, with a time of 1148 ms and a size of 467 B. The response body is a JSON object with the following fields: `id` (5), `brand` (FS0), `model` (Polonez), `gearbox` (manual), `pricePerDay` (30), `amountOfDoors` (5), `amountOfSeats` (5), `mileageInLiter` (12), and `airConditioning` (true).

```
POST http://127.0.0.1:8080/car

{
  "brand": "FS0",
  "model": "Polonez",
  "gearbox": "manual",
  "pricePerDay": 30,
  "amountOfDoors": 5,
  "amountOfSeats": 5,
  "mileageInLiter": 12,
  "airConditioning": true
}
```

Status: 200 OK Time: 1148 ms Size: 467 B Save Response

```
{
  "id": 5,
  "brand": "FS0",
  "model": "Polonez",
  "gearbox": "manual",
  "pricePerDay": 30,
  "amountOfDoors": 5,
  "amountOfSeats": 5,
  "mileageInLiter": 12,
  "airConditioning": true
}
```

przy niepoprawnych danych

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8080/car`. The request body is a JSON object with the following fields: `brand` (FS0), `model` (Polonez), `amountOfSeats` (5), `mileageInLiter` (12), `airConditioning` (true), and `asdsad` (false). The response status is 400 Bad Request, with a time of 23 ms and a size of 285 B. The response body is empty.

```
POST http://127.0.0.1:8080/car

{
  "brand": "FS0",
  "model": "Polonez",
  "amountOfSeats": 5,
  "mileageInLiter": 12,
  "airConditioning": true,
  "asdsad": false
}
```

Status: 400 Bad Request Time: 23 ms Size: 285 B Save Response

PUT /car{id}

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

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://127.0.0.1:8080/car/5
- Body:** A JSON object with the following fields:

```
{  "brand": "FSO",  "model": "Polonez OHV",  "gearbox": "manual",  "pricePerDay": 30,  "amountOfDoors": 5,  "amountOfSeats": 5,  "mileageInLiter": 12,  "airConditioning": true}
```
- Status:** 200 OK
- Time:** 267 ms
- Size:** 257 B
- Response:** (Empty body)

Błędne dane lub złe id

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://127.0.0.1:8080/car/233
- Body:** A JSON object with the following fields:

```
{  "brand": "FSO",  "model": "Polonez OHV",  "gearbox": "manual",  "pricePerDay": null,  "sgf": 5,  "fdg": 12,  "dfdfg": true}
```
- Status:** 400 Bad Request
- Time:** 343 ms
- Size:** 514 B
- Response:** A JSON object with the following fields:

```
{  "message": "Invalid method arguments",  "errors": {    "amountOfSeats": "must not be null",    "amountOfDoors": "must not be null",    "pricePerDay": "must not be null"  },  "instance": "/car/233"}
```

DELETE car/{id}

DELETE

http://127.0.0.1:8080/car/5

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 (7)

Test Results

Status: 204 No Content

Time: 691 ms

Size: 246 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

Nie istniejące lub złe id

GET http://127.0.0.1:80...

PUT http://127.0.0.1:80...

DEL http://127.0.0.1:80...

+

...

No Environment

http://127.0.0.1:8080/car

Save

DELETE

http://127.0.0.1:8080/car/23553

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: 2.66 s

Size: 519 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

{

"timestamp": "2021-05-13T19:06:22.503+0000",

"status": 500,

"error": "Internal Server Error",

"message": "attempt to create delete event with null entity",

"path": "/car/23553"

}

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

Autorzy:
Olga Żalińska
Patrik Żur