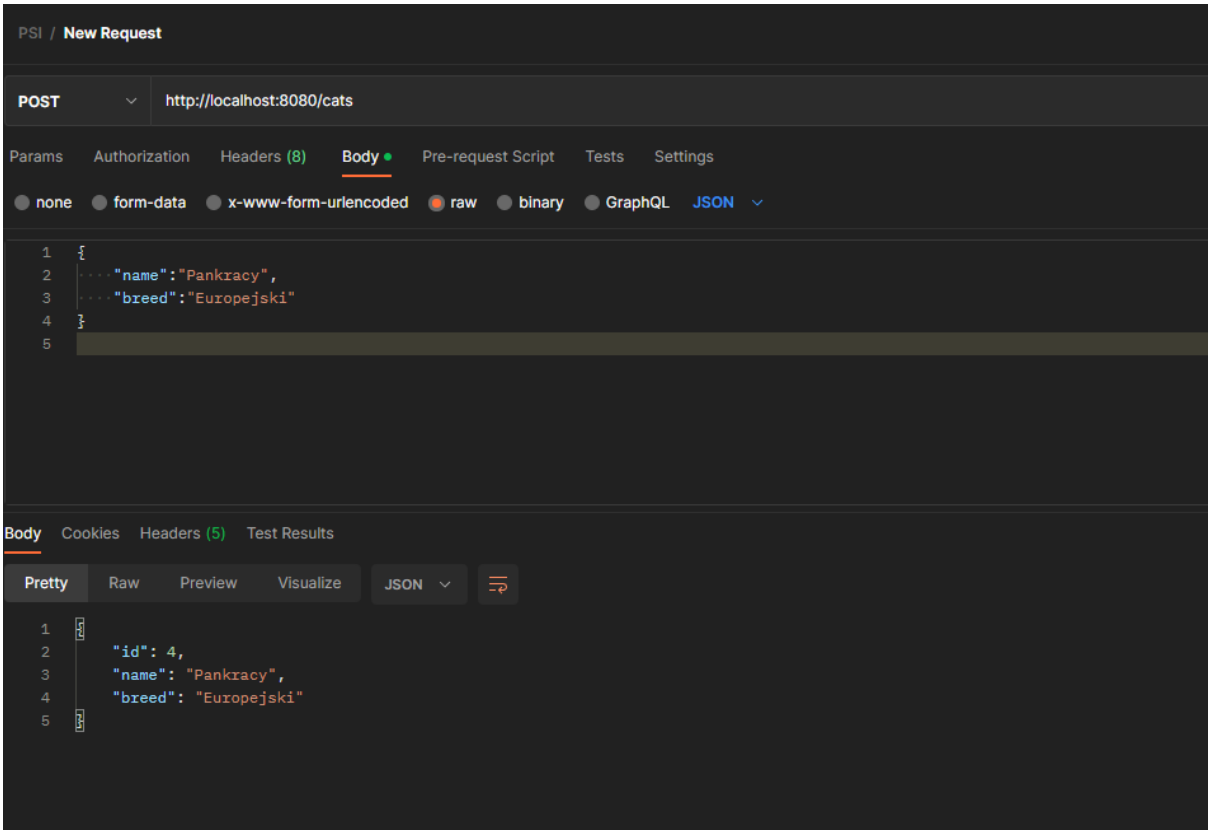
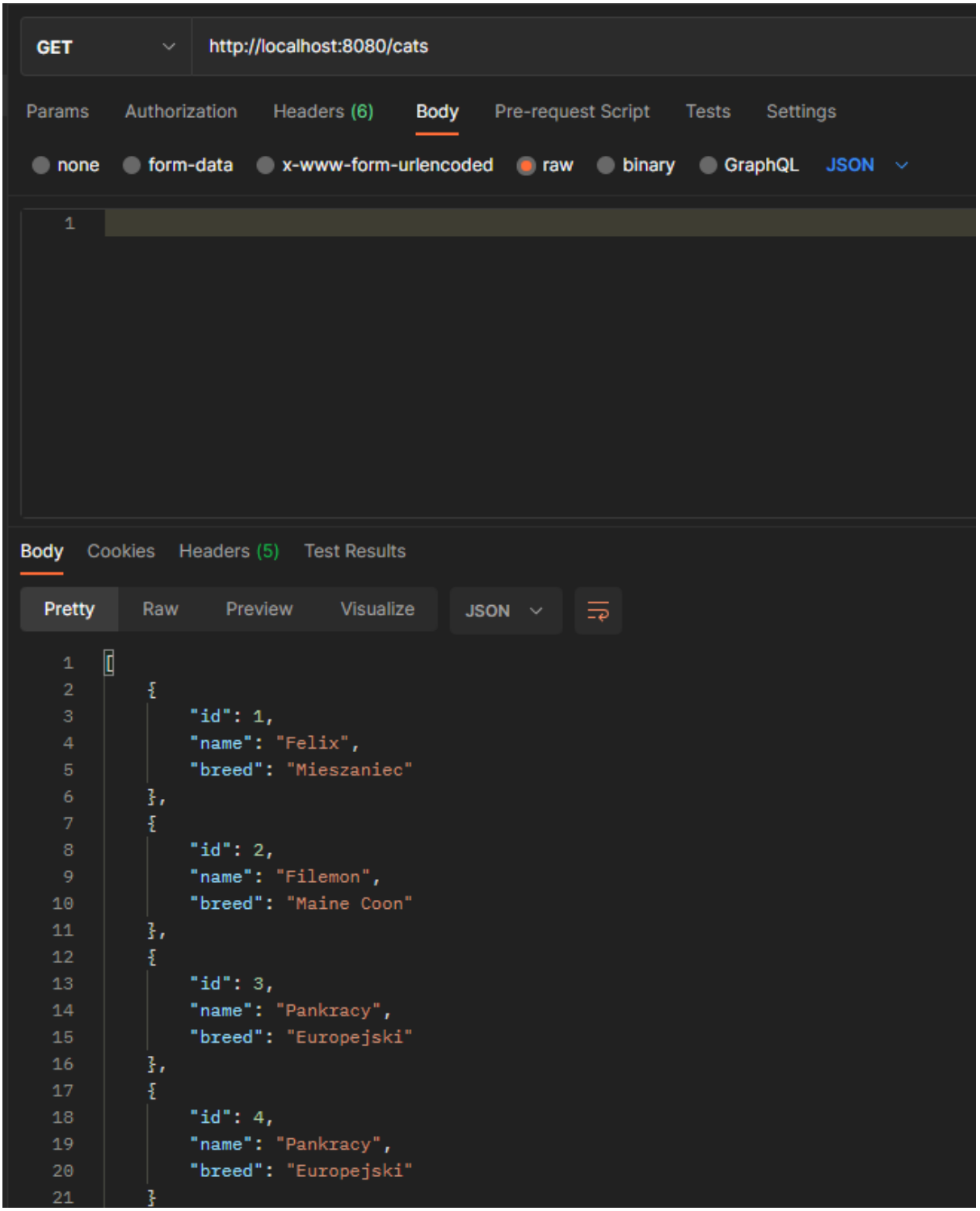
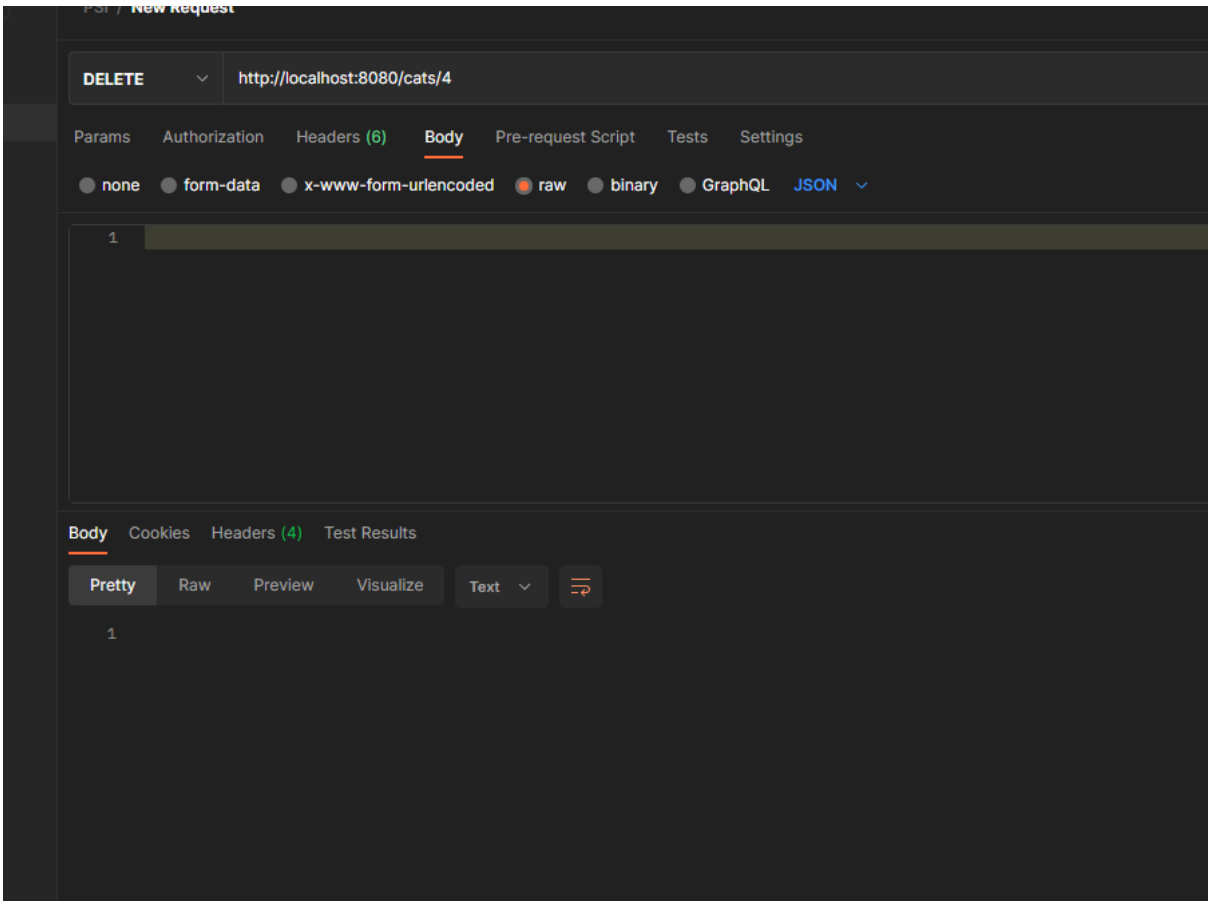
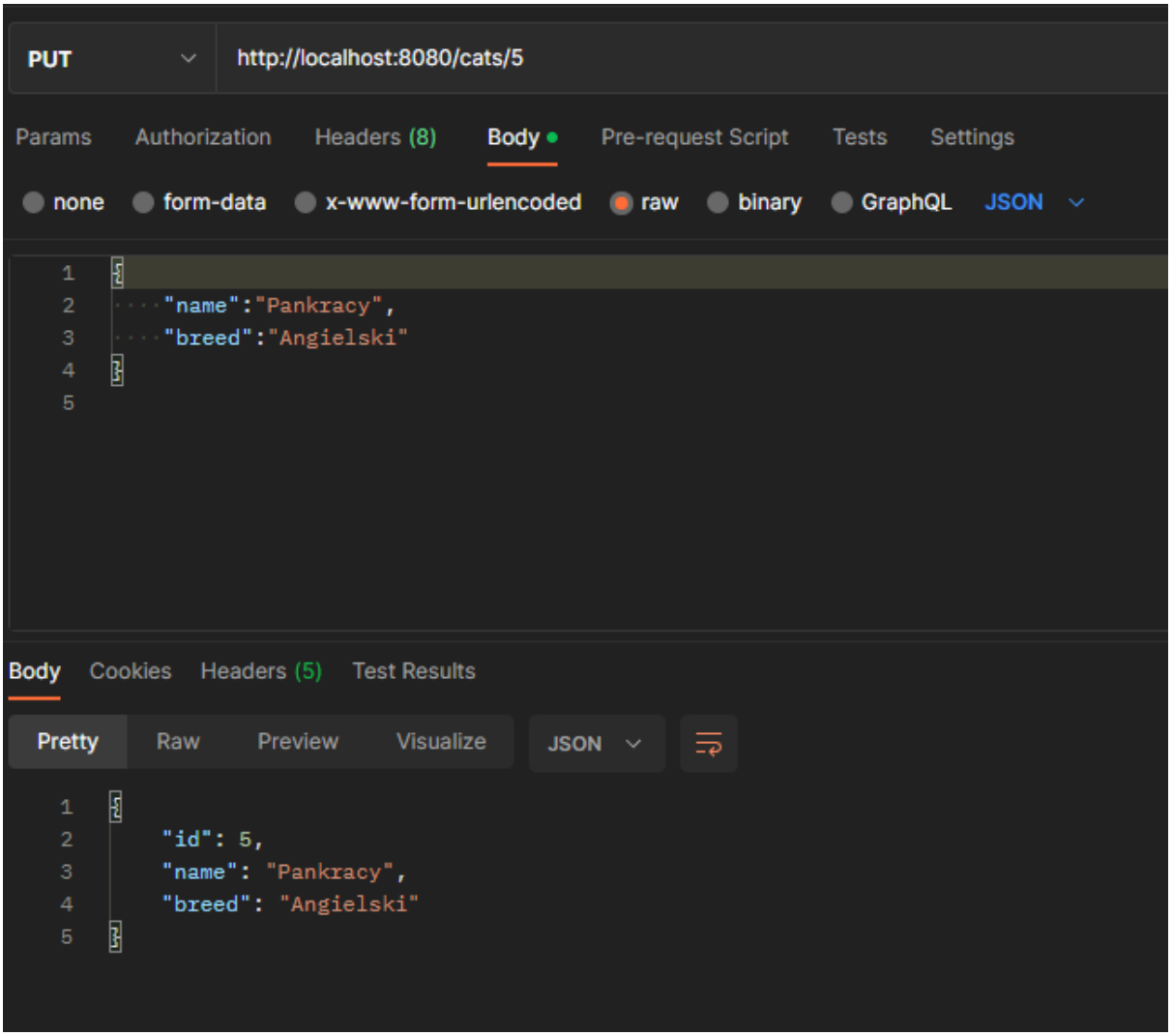
PAW REST API:
Wiktor Fedde, s49016 , informatyka, Ailm, grupa 2

Screeny z metod http:




GET
http://localhost:8080/cats
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"name": "Felix",
"breed": "Mieszaniec"
},
{
"id": 2,
"name": "Filemon",
"breed": "Maine Coon"
},
{
"id": 3,
"name": "Pankracy",
"breed": "Europejski"
},
{
"id": 4,
"name": "Pankracy",
"breed": "Europejski"
}


DELETE
http://localhost:8080/cats/4
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1
Body
Cookies
Headers (4)
Test Results
Pretty
Raw
Preview
Visualize
Text
1

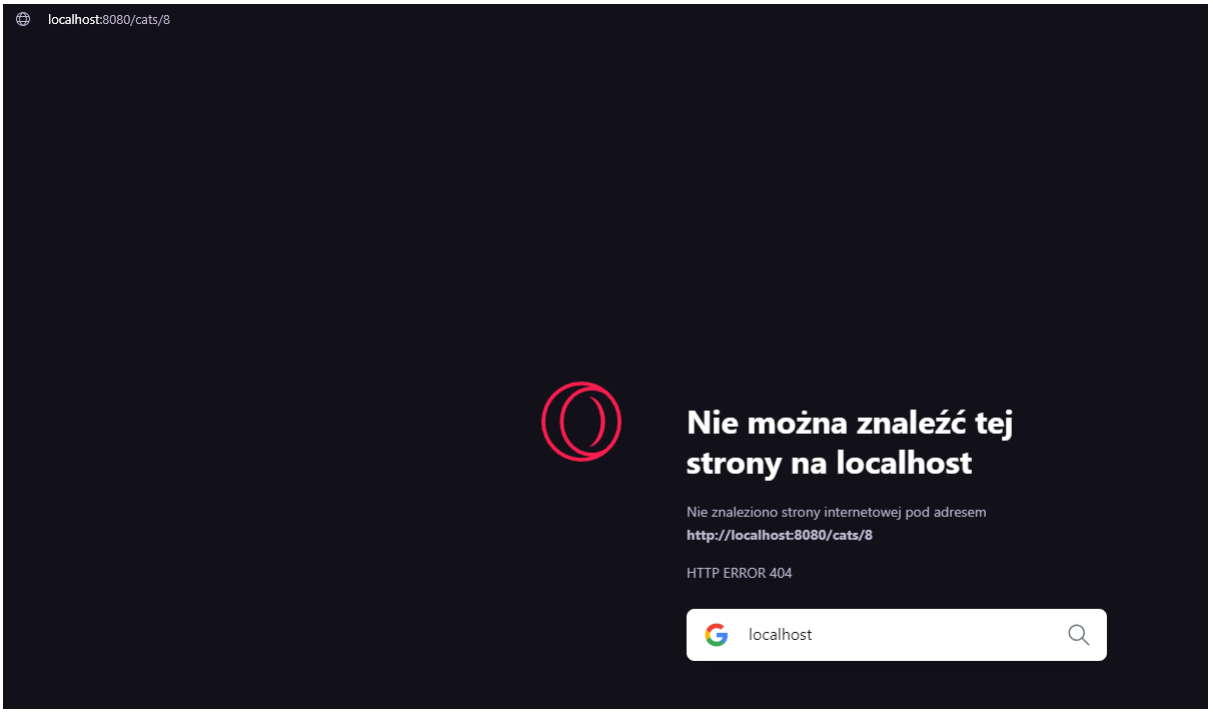

PUT
http://localhost:8080/cats/5
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"name":"Pankracy",
"breed":"Angielski"
}
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"id": 5,
"name": "Pankracy",
"breed": "Angielski"
}


GET
http://localhost:8080/fun-fact
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"name":"Pankracy",
"breed":"Angielski"
}
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 421 ms
Pretty
Raw
Preview
Visualize
Text
{"fact":"A cat uses its whiskers for measuring distances. The whiskers of a cat are capable of registering very small changes in air pressure.","length":134}

7. Obsługa wyjątków:

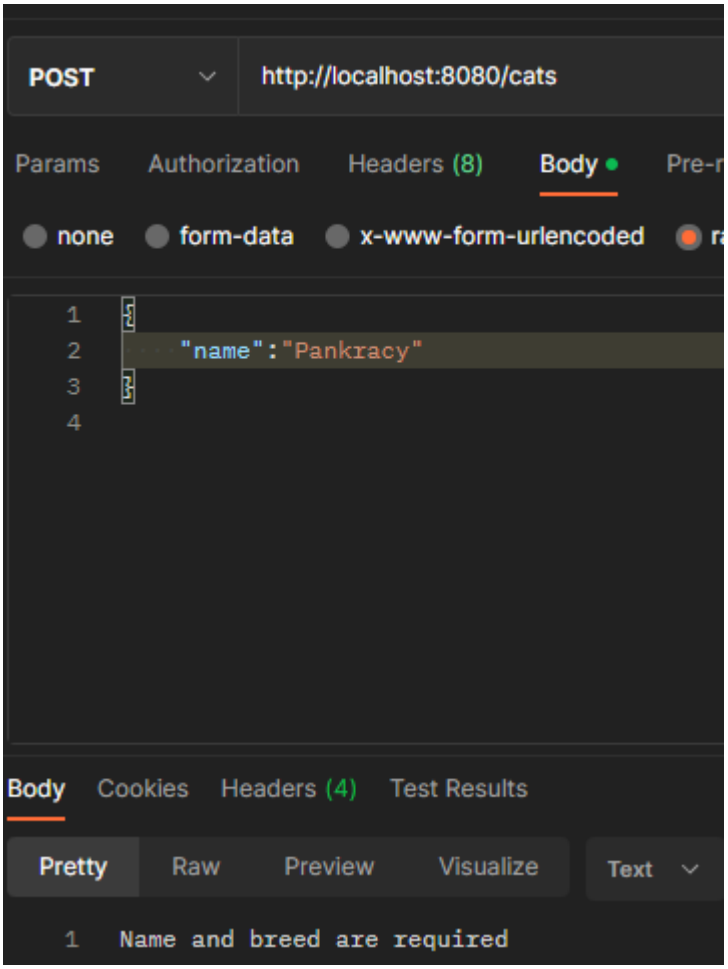
Obsługa błędu 404 (Not Found) dla żądania GET:

```java
@GetMapping("/cats/{id}")
ResponseEntity<?> getCat(@PathVariable("id") Long id) {
    Optional<Cat> cat = repository.findById(id);
    return cat.map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
}
```



Obsługa błędu 400 (Bad Request) dla żądania POST:

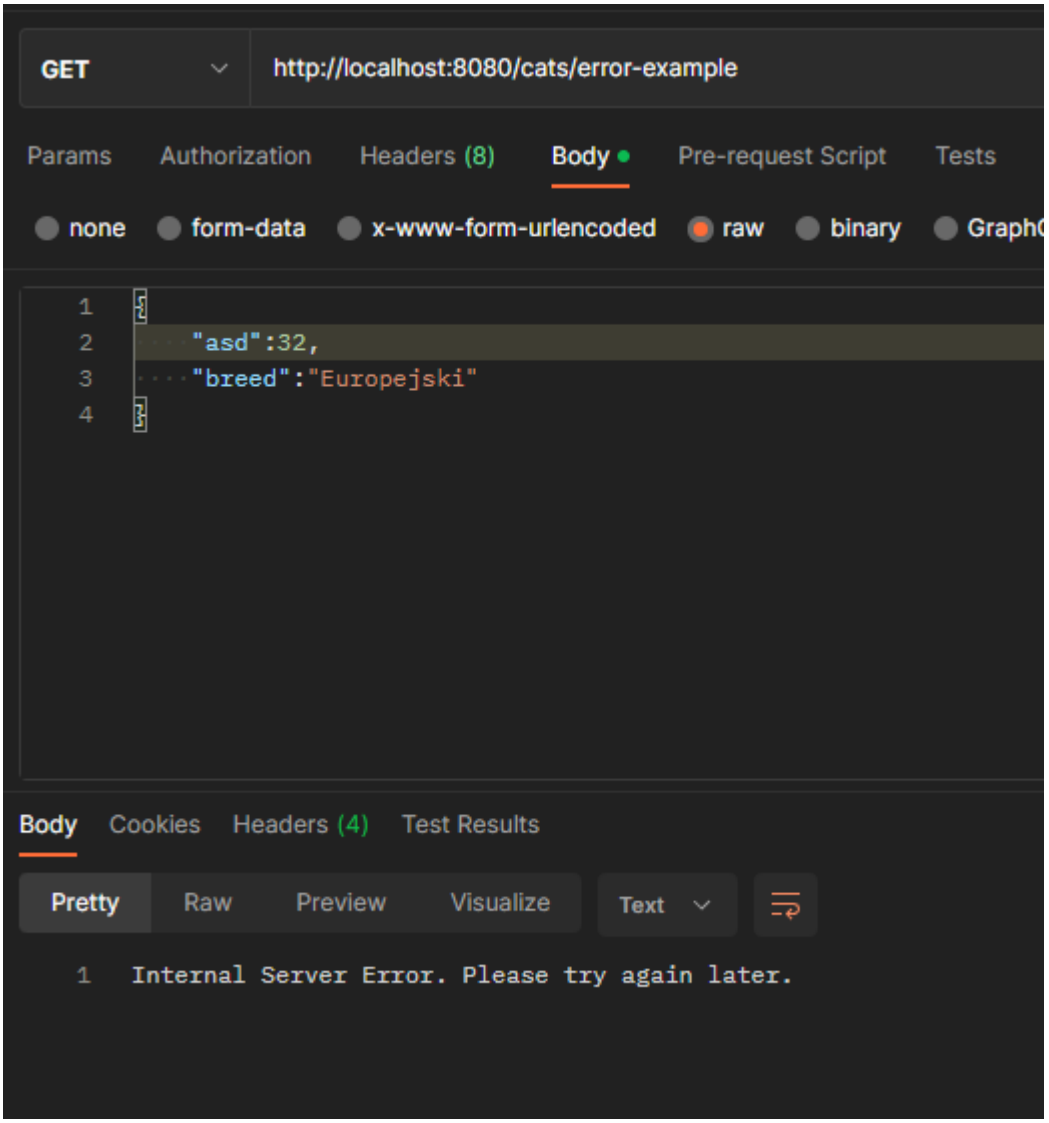
```java
@PostMapping("/cats")
ResponseEntity<?> newCat(@RequestBody Cat newCat) {
    if (newCat.getName() == null || newCat.getBreed() == null) {
        return ResponseEntity.badRequest().body("Name and breed are required");
    }
    Cat savedCat = repository.save(newCat);
    return ResponseEntity.ok(savedCat);
}
```



Obsługa błędu 500 (Internal Server Error) dla nieoczekiwanych sytuacji:
Dołożyłem metodę generateError() w celu pokazania jak działa ten wyjątek.

```
@ExceptionHandler(Exception.class)
public ResponseEntity<String> handleException(Exception e) {
    e.printStackTrace();

    // HTTP status 500
    return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR)
            .body("Internal Server Error. Please try again later.");
}
// example of an error
@GetMapping("/cats/error-example")
public ResponseEntity<String> generateError() {
    throw new RuntimeException("This is a simulated internal server error");
}
```



Strona html + jss:

Cat Information

Cat Name: Oskar Cat Breed: Kru Add Cat

Fetch Cat Information

Cat Name: Felix, Breed: Mieszaniec
Cat Name: Filemon, Breed: Maine Coon
Cat Name: Oskar, Breed: Kru