

POST: /pet

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "category": {
      "id": 1,
      "name": "dogs"
    },
    "name": "doggie",
    "photoUrls": [
      "www.image.com"
    ],
    "tags": [
      {
        "id": 1,
        "name": "puppy"
      }
    ],
    "status": "sold"
  }'
```

First I would test the with valid payload and valid headers with like above.

Once I send the post Request to /pet endpoint like above I would check the following list in order.

- 1- Check if the status code is “200”
- 2- Check if the response body is the same with the following json

```
{
  "id": 1,
  "category": {
    "id": 1,
    "name": "dogs"
  },
  "name": "doggie",
  "photoUrls": [
    "www.image.com"
  ],
  "tags": [
    {
      "id": 1,
      "name": "puppy"
    }
  ],
  "status": "sold"
}
```

```
}
```

3- Check if response header includes same headers with following header list

- access-control-allow-headers: Content-Type,api_key,Authorization
- access-control-allow-methods: GET,POST,DELETE,PUT
- access-control-allow-origin: *
- content-type: application/json
- date: Wed,21 Dec 2022 18:24:05 GMT
- server: Jetty(9.2.9.v20150224)

4- Verify the pet is created on DataBase with the same value on the table as should be

5- After the successful case is complete I would test the following negative case with the same approach.

405

Invalid input

PUT: /pet

```
curl -X 'PUT' \
'https://petstore.swagger.io/v2/pet' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "id": 1,
  "category": {
    "id": 1,
    "name": "cat"
  },
  "name": "cats",
  "photoUrls": [
    "www.image1.com"
  ],
  "tags": [
    {
      "id": 2,
      "name": "kitty"
    }
  ],
  "status": "availbe"
}'
```

I would find an already exists pet from DB or I can create a new one first to test the PUT (update)

Then I would test it with the same approach. But this time I would check whether the Database record is updated with the same data or not. After a Put request, the entity should update with new data which is passed with the PUT request.

And also there are different negative cases that should be tested

400	Invalid ID supplied
404	Pet not found
405	Validation exception

GET: /pet/{petId}

I would find an already existing pet from DB or I can create a new one first to test the GET

```
curl -X 'GET' \
  'https://petstore.swagger.io/v2/pet/{existpetid}' \
  -H 'accept: application/json'
```

1- I would verify the response body fields includes the same value with the database record that with same pet id

2- I would send a GET request with the invalid supplied id after that I would check the status code is 400

3- I would send a GET request with the id that not exists on the database after that I would check the status code is 404 pet not found