Update an existing pet.

pet · https://cbotai.shahrear.msf.bd/docs/update-an-existing-pet/

PUT `/pet`

Code Samples

```
curl --request PUT \
  --url '/api/v3/pet' \
  --header 'Authorization: Bearer YOUR_TOKEN' \
  --header 'Content-Type: application/json' \
  --data '{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}'
```

```
const response = await fetch('/api/v3/pet', {
  method: 'PUT',
  headers: {
    'Authorization': 'Bearer YOUR_TOKEN',
    'Content-Type': 'application/json'
```

```
  },
  body: JSON.stringify({
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
})
});

const data = await response.json();
console.log(data);
```

```
import requests

headers = {
    "Authorization": "Bearer YOUR_TOKEN",
    "Content-Type": "application/json"
}

payload = {
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
```

```
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}

response = requests.put("/api/v3/pet", headers=headers, json=payload)

print(response.json())
```

```
<?php
$curl = curl_init();

curl_setopt_array($curl, [
  CURLOPT_URL => '/api/v3/pet',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'PUT',
  CURLOPT_HTTPHEADER => [
  'Authorization: Bearer YOUR_TOKEN',
  'Content-Type: application/json'
  ],
  CURLOPT_POSTFIELDS => '{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
```

```
    ],
    "status": "available"
}',
]);

$response = curl_exec($curl);
curl_close($curl);

echo $response;
```

```
using System.Net.Http;
using System.Text;

var client = new HttpClient();
var request = new HttpRequestMessage(new HttpMethod("PUT"), "/api/v3/pet");
request.Headers.Add("Authorization", "Bearer YOUR_TOKEN");
request.Content = new StringContent(@"{
    ""id"": 10,
    ""name"": ""doggie"",
    ""category"": {
        ""id"": 1,
        ""name"": ""Dogs""
    },
    ""photoUrls"": [
        ""string""
    ],
    ""tags"": [
        {
            ""id"": 1,
            ""name"": ""string""
        }
    ],
    ""status"": ""available""
}", Encoding.UTF8, "application/json");
var response = await client.SendAsync(request);
var body = await response.Content.ReadAsStringAsync();
Console.WriteLine(body);
```

```
import java.net.URI;
import java.net.http.*;
```

```
HttpClient client = HttpClient.newHttpClient();
HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create("/api/v3/pet"))
    .header("Authorization", "Bearer YOUR_TOKEN")
    .header("Content-Type", "application/json")
    .method("PUT", HttpRequest.BodyPublishers.ofString("""
{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}
"""))
    .build();

HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
System.out.println(response.body());
```

Update an existing pet by Id.

Request Body

Content type: `application/json` — **required**

Field	Type	Required	Description
`id`	integer (int64)	Optional	

Field	Type	Required	Description
`name`	string	**Required**	
`category`	object	Optional	
`category.id`	integer (int64)	Optional	
`category.name`	string	Optional	
`photoUrls`	array of string	**Required**	
`tags`	array of object	Optional	
`status`	string	Optional	pet status in the store Allowed: available, pending, sold


```
{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}
```


Responses

200 — Successful operation

Field	Type	Required	Description

`id`	integer (int64)	Optional	
`name`	string	**Required**	
`category`	object	Optional	
`category.id`	integer (int64)	Optional	
`category.name`	string	Optional	
`photoUrls`	array of string	**Required**	
`tags`	array of object	Optional	
`status`	string	Optional	pet status in the store Allowed: available, pending, sold


```
{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}
```

400 — Invalid ID supplied

404 — Pet not found

422 — Validation exception

default — Unexpected error

Response

```
200
{
    "id": 10,
    "name": "doggie",
    "category": {
        "id": 1,
        "name": "Dogs"
    },
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 1,
            "name": "string"
        }
    ],
    "status": "available"
}
```