Updates a pet in the store with form data.

pet · https://cbotai.shahrear.msf.bd/docs/updates-a-pet-in-the-store-with-form-data/

POST `/pet/{petId}`

Code Samples

```
curl --request POST \
  --url '/api/v3/pet/{petId}' \
  --header 'Authorization: Bearer YOUR_TOKEN'
```

```
const response = await fetch('/api/v3/pet/{petId}', {
  method: 'POST',
  headers: {
    'Authorization': 'Bearer YOUR_TOKEN'
  }
});

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

```
import requests

headers = {
    "Authorization": "Bearer YOUR_TOKEN"
}

response = requests.post("/api/v3/pet/{petId}", headers=headers)

print(response.json())
```

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

```
curl_setopt_array($curl, [
  CURLOPT_URL => '/api/v3/pet/{petId}',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_HTTPHEADER => [
  'Authorization: Bearer YOUR_TOKEN'
  ],
]);

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

echo $response;
```

```
using System.Net.Http;

var client = new HttpClient();
var request = new HttpRequestMessage(new HttpMethod("POST"), "/api/v3/pet/{petId}");
request.Headers.Add("Authorization", "Bearer YOUR_TOKEN");
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/{petId}"))
    .header("Authorization", "Bearer YOUR_TOKEN")
    .method("POST", HttpRequest.BodyPublishers.noBody())
    .build();

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

Updates A pet resource based on the form data.

Path Parameters

Name	Type	Required	Description
`petId`	integer (int64)	**Required**	ID of pet that needs to be updated

Query Parameters

Name	Type	Required	Description
`name`	string	Optional	Name of pet that needs to be updated
`status`	string	Optional	Status of pet that needs to be updated

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 input

default — Unexpected error

Response

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

```
    "status": "available"
}
```