Finds Pets by tags.

pet · https://cbotai.shahrear.msf.bd/docs/finds-pets-by-tags/

GET `/pet/findByTags`

Code Samples

```
curl --request GET \
  --url '/api/v3/pet/findByTags?tags=value' \
  --header 'Authorization: Bearer YOUR_TOKEN'
```

```
const response = await fetch('/api/v3/pet/findByTags?tags=value', {
  method: 'GET',
  headers: {
    'Authorization': 'Bearer YOUR_TOKEN'
  }
});

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

```
import requests

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

response = requests.get("/api/v3/pet/findByTags?tags=value", headers=headers)

print(response.json())
```

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

```
curl_setopt_array($curl, [
  CURLOPT_URL => '/api/v3/pet/findByTags?tags=value',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  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("GET"), "/api/v3/pet/findByTags?tags=value");
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/findByTags?tags=value"))
    .header("Authorization", "Bearer YOUR_TOKEN")
    .method("GET", HttpRequest.BodyPublishers.noBody())
    .build();

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

Multiple tags can be provided with comma separated strings. Use tag1, tag2, tag3 for testing.

Query Parameters

Name	Type	Required	Description
`tags`	array of string	**Required**	Tags to filter by

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 tag value

default — Unexpected error

Response

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