

## API Test Report – PetStore Demo

### 1. Informações Gerais

- API testada: Swagger PetStore (<https://petstore3.swagger.io/>)
- Data do teste: 07/08/2025
- Testador: Mário Queiroz
- Ferramentas utilizadas:
  - Postman
  - Docker Desktop

### 2. Configuração do Teste

- Ambiente: API pública de demonstração
- Módulo testado: Store
- Método HTTP testado: GET, POST e DELETE.
- Formato de dados: JSON
- Dados utilizados para o pedido(order) teste:

```
{
  "id": 192,
  "petId": 198772,
  "quantity": 7,
  "shipDate": "2025-08-07T23:31:44.173Z",
  "status": "approved",
  "complete": true
}
```

### 3. Casos de Teste

Teste nº	Endpoint	Método	Entrada	Esperado	Obtido	Status
T001	/store/order	POST	{ "id": 192, "petId": 198772, "quantity": 7, "shipDate": "2025-08-07T23 :31:44.173Z", "status": "approved", "complete": true }	200 OK	200 OK	PASS

T002	/store/order	POST	{ "id": 192, "petId": 198772, "quantity": 7, }	400 Invalid Order	200 OK	FAIL
T003	/store/order	POST	{ }	400 Invalid Order	200 OK	FAIL
T004	/store/order	POST	{ "id": 192, }	400 Bad Request	400 Bad Request	PASS
T005	/store/order/{orderId}	GET	orderId=192	200 OK	200 OK	PASS
T006	/store/order/{orderId}	GET	orderId=99999	404 Not Found	404 Not Found	PASS
T007	/store/order/{orderId}	DELETE	orderId=192	200 OK	200 OK	PASS
T008	/store/order/{orderId}	DELETE	orderId=192	404 Not Found	404 Not Found	PASS
T009	/store/inventory	GET	N/A	200 OK	200 OK	PASS

#### 4. Observações

- Devido a grande quantidade de erros baseados em servidor, foi decidido que a testagem seria através de uma instância local, por meio de um container, com os arquivos fornecidos através do repositório do PetStore.

- POST /store/order

- A API não trata bem as requisições com ausência de campos, em que o campo "complete" é definido automaticamente como "false" caso não seja definido na request. Todavia, o campo "status", que segundo a documentação deveria aceitar apenas "placed", "delivered" e "approved", não possui quaisquer verificações, aceitando valores aleatórios. Além disso, constata-se que a API aceita pedidos vazios, em que o id apresenta a auto-increment, no entanto, pedidos vazios não deveriam ser aceitos.

## 5. Evidências

T002:

The screenshot displays a REST client interface with a dark theme. At the top, the URL bar shows 'HTTP store / Place an order for a pet'. Below this, the request method is set to 'POST' and the URL is '{{url}}/order'. The 'Send' button is visible. The 'Body' tab is selected, showing a raw JSON request:

```
1 {
2   "id": 192,
3   "petId": 198772,
4   "quantity": 7
5 }
6
7
```

The response status is '200 OK' with a time of '13 ms' and a size of '357 B'. The response body is shown in JSON format:

```
1 {
2   "id": 192,
3   "petId": 198772,
4   "quantity": 7,
5   "complete": false
6 }
```

T003:

The screenshot displays a REST client interface with a dark theme. At the top, the URL bar shows 'store / Place an order for a pet' with a 'Save' button. Below this, the request configuration shows a 'POST' method and a placeholder URL '{{url}}/order', with a 'Send' button. The 'Body' tab is selected, showing a 'raw' JSON body with a simple object structure: { }. The bottom section shows the response status '200 OK' with a response time of 22 ms and a size of 352 B. The response body is displayed as a formatted JSON object: { "id": 195, "petId": 0, "quantity": 0, "complete": false }.

store / Place an order for a pet

POST {{url}}/order

Send

Params Auth Headers (9) Body Scripts Settings

raw JSON

```
1 {
2
3 }
```

Body 200 OK • 22 ms • 352 B • Save Response

{ JSON Preview Visualize

```
1 {
2   "id": 195,
3   "petId": 0,
4   "quantity": 0,
5   "complete": false
6 }
```