

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №3

по дисциплине: Тестирование программных систем
тема: «Тестирование REST API»

Выполнил: ст. группы ПВ-223
Игнатьев Артур Олегович

Проверили:
Бабенко Анастасия Александровна

Лабораторная работа №3

Тестирование REST API

Цель работы: научиться проектировать тестовые сценарии для тестирования REST API.

Вариант 3

Тест-кейсы

1. Добавление питомца (POST /pet).
 - 1.1. Позитивный сценарий (корректные данные).
 - 1.2. Добавление лишних полей.
 - 1.3. Некорректные типы данных.
 - 1.4. Отсутствие обязательных полей.
2. Поиск питомцев по статусу (GET /pet/findByStatus).
 - 2.1. Корректный запрос с одним статусом.
 - 2.2. Корректный запрос с несколькими статусами.
 - 2.3. Некорректный статус.
 - 2.4. Отсутствие параметра status.
 - 2.5. Неизвестный, но валидный формат статуса.

Ход выполнения работы

1. Добавление питомца (POST /pet).

1.1. Позитивный сценарий (корректные данные).

Ожидаемый результат: успешное добавление, код 201.

Фактический результат: код 200, питомец добавлен.

The screenshot shows the Postman application interface. On the left, the sidebar displays 'NTK-Hub's Workspace' with sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace shows a collection named 'My Collection' containing a POST 1.1 endpoint and a GET pet/findByStatus endpoint. The POST 1.1 endpoint is selected, showing its configuration: Method: POST, URL: https://petstore.swagger.io/v2/pet, Headers: (9), Body (selected), and Body type: raw JSON. The JSON body is defined as:

```
1 {  
2   "name": "TestDog",  
3   "photoUrls": ["https://example.com/dog.jpg"],  
4   "status": "available"  
5 }
```

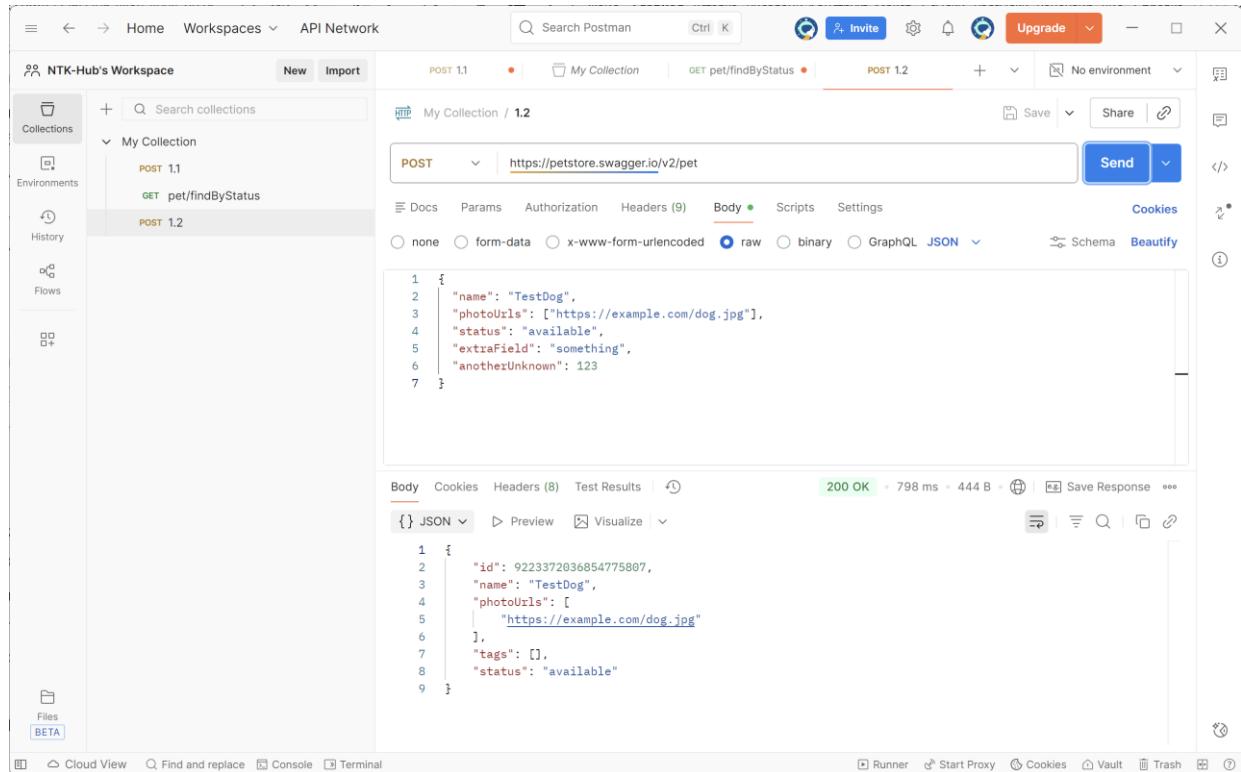
Below the body, the response section shows a 200 OK status with 707 ms duration and 444 B size. The response body is displayed as:

```
{ } JSON ▾ ▶ Preview ▾ Visualize ▾  
1 {  
2   "id": 9223372036854775807,  
3   "name": "TestDog",  
4   "photoUrls": [  
5     "https://example.com/dog.jpg"  
6   ],  
7   "tags": [],  
8   "status": "available"  
9 }
```

1.2. Добавление лишних полей.

Ожидаемый результат: сервер должен возвращать ошибку, если запрос содержит избыточные данные 422.

Фактический результат: код 200 OK, питомец добавлен, лишние поля игнорируются.



The screenshot shows the Postman interface with the following details:

- Workspace:** NTK-Hub's Workspace
- Collection:** My Collection
- Request:** POST 1.2 (selected)
- URL:** <https://petstore.swagger.io/v2/pet>
- Body:** Raw JSON (selected) containing:

```
1 {
2     "name": "TestDog",
3     "photoUrls": ["https://example.com/dog.jpg"],
4     "status": "available",
5     "extraField": "something",
6     "anotherUnknown": 123
7 }
```
- Response:** 200 OK (798 ms, 444 B)
Raw JSON response:

```
1 {
2     "id": 9223372036854775807,
3     "name": "TestDog",
4     "photoUrls": [
5         "https://example.com/dog.jpg"
6     ],
7     "tags": [],
8     "status": "available"
9 }
```

1.3. Некорректные типы данных.

Ожидаемый результат: в случае некорректного типа сервер должен ошибку 422.

Фактический результат: сервер выдаёт необрабатываемую ошибку с кодом 500.

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, Environments, History, Flows, and Files (BETA).
- Top Bar:** Home, Workspaces, API Network, Search Postman, Invite, Upgrade, and various icons.
- Middle Left:** NTK-Hub's Workspace, New, Import, My Collection / 1.3, POST 1.1, GET pet/findByStatus, POST 1.2, and GET 1.3.
- Middle Right:** Save, Share, Send, and Headers tab (selected), showing raw JSON body content:

```
1 {  
2   "name": "TestDog",  
3   "photoUrls": "not_an_array"  
4 }
```
- Bottom Right:** Cookies, Schema, Beautify, and other UI elements.
- Bottom:** Body, Cookies, Headers (8), Test Results, and a 500 Server Error message: "500 Server Error" with status 673 ms, 400 B, and a "Save Response" button.
- Bottom Left:** Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, and a Help icon.

1.4. Отсутствие обязательных полей.

Ожидаемый результат: при их отсутствии сервер должен выдавать ошибку 422.

Фактический результат: сервер успешно позволяет создать объект без полей **вовсе**.

The screenshot shows the Postman application interface. On the left, the sidebar displays 'NTK-Hub's Workspace' with collections, environments, history, and flows. The main area shows a collection named 'My Collection / 1.4'. A POST request titled 'POST 1.4' is selected, with the URL 'https://petstore.swagger.io/v2/pet'. The 'Body' tab is active, showing the raw JSON input:

```
1
2 "status": "available"
3 }
```

Below the request, the response details are shown: '200 OK' status, '722 ms' duration, and '398 B' size. The response body is displayed as:

```
{ } JSON
```

```
1 {
2   "id": 9223372036854775807,
3   "photoUrls": [],
4   "tags": [],
5   "status": "available"
6 }
```

At the bottom, there are various navigation and utility buttons like Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, etc.

2. Поиск питомцев по статусу (GET /pet/findByStatus).

2.1. Корректный запрос с одним статусом.

Ожидаемый результат: код 200, возвращён большой массив питомцев с указанным статусом.

Фактический результат: код 200, возвращён массив из тысяч объектов.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'History', and 'Flows'. The main area shows a 'My Collection' folder containing several requests: POST 1.1, GET 2.1 (selected), GET 2.2, POST 1.2, POST 1.3, and POST 1.4. The 'GET 2.1' request is highlighted with a red border. The request details panel shows a GET method pointing to <https://petstore.swagger.io/v2/pet/findByStatus?status=available>. The 'Params' tab is selected, showing a table with a single row: 'status' set to 'available'. Below this, the 'Body' tab shows a JSON response with a large array of objects, each representing a pet. The first few objects in the array are:

```
11712     "photoUrls": [
11713         "https://example.com/photo1.jpg"
11714     ],
11715     "tags": [
11716         {
11717             "id": 1,
11718             "name": "test-tag"
11719         }
11720     ],
11721     "status": "available"
11722 ]
11723 ]
```

The 'Test Results' panel at the bottom indicates a 200 OK response with 1.43 seconds duration and 124.27 KB size. There are buttons for 'Save Response' and other actions like 'Runner' and 'Start Proxy'.

2.2. Корректный запрос с несколькими статусами.

Ожидаемый результат: код 200, массив содержит питомцев всех указанных статусов.

Фактический результат: код 200, возвращён очень большой массив.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'Environments', 'History', and 'Flows'. The main area displays a collection named 'My Collection' containing several requests: POST 1.1, GET 2.1, GET 2.2 (which is selected), POST 1.2, POST 1.3, and POST 1.4. The 'GET 2.2' request is detailed in the center, showing it's a GET request to the URL <https://petstore.swagger.io/v2/pet/findByStatus?status=available,pending,sold>. The 'Params' tab is selected, showing a 'status' parameter with values 'available', 'pending', and 'sold'. Below this, the 'Body' tab shows a JSON response with multiple items, each containing photo URLs, tags, and a pending status. The bottom status bar indicates a 200 OK response with a duration of 1.73 seconds and a size of 179.09 KB.

2.3 Некорректный статус.

Ожидаемый результат: сервер должен выдавать ошибку 400.

Фактический результат: код 200, возвращён пустой массив [].

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'History', 'Flows', and 'Files (BETA)'. The main area shows a 'My Collection' folder containing several API endpoints. One endpoint, 'GET 2.3', is selected. The URL is set to `https://petstore.swagger.io/v2/pet/findByStatus?status=invalid`. The 'Params' tab is active, showing a table with a single row: 'status' with value 'invalid'. Below the table, the 'Body' tab shows a JSON response: `{ } JSON` with a preview of `[]`. The status bar at the bottom indicates a `200 OK` response with `832 ms`, `328 B`, and other details like `Save Response`.

2.4. Отсутствие параметра status.

Ожидаемый результат: сервер должен выдавать ошибку 400.

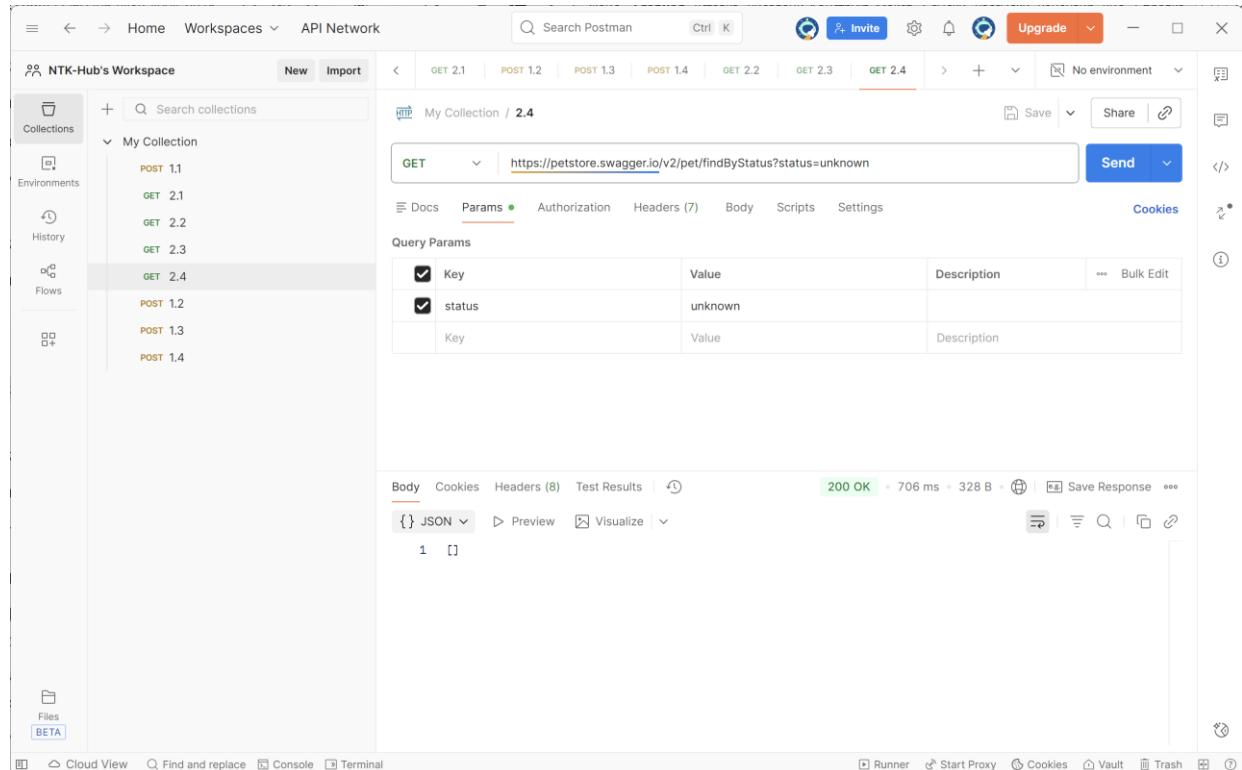
Фактический результат: код 200, возвращён пустой массив [].

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'History', 'Flows', and 'Files (BETA)'. The main area shows a 'My Collection' folder containing several requests: POST 1.1, GET 2.1, GET 2.2, and GET 2.3. GET 2.3 is currently selected. The URL bar shows 'https://petstore.swagger.io/v2/pet/findByStatus'. The 'Params' tab is active, showing an empty table for query parameters. Below the table, the response section shows a 200 OK status with a response body of '[]'. The bottom navigation bar includes 'Body', 'Cookies', 'Headers (8)', 'Test Results', and other tools like 'Cloud View', 'Find and replace', 'Console', 'Terminal', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and help icons.

2.5. Неизвестный, но валидный формат статуса.

Ожидаемый результат: код 200, пустой массив [].

Фактический результат: код 200, возвращён пустой массив [].



The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, Environments, History, Flows, and Files (BETA). The main area displays a collection named "My Collection" containing six requests: POST 1.1, GET 2.1, GET 2.2, GET 2.3, GET 2.4 (which is currently selected), POST 1.2, POST 1.3, and POST 1.4. The selected request, GET 2.4, has a URL of <https://petstore.swagger.io/v2/pet/findByStatus?status=unknown>. The "Params" tab is active, showing a table with a row for "status" having a value of "unknown". Below the table, the "Body" tab shows the response body as an empty array: []. The status bar at the bottom indicates a 200 OK response with 706 ms duration and 328 B size.

Вывод: в ходе лабораторной работы научились проектировать тестовые сценарии для тестирования REST API. Познакомились с инструментом тестирования REST API - Postman.