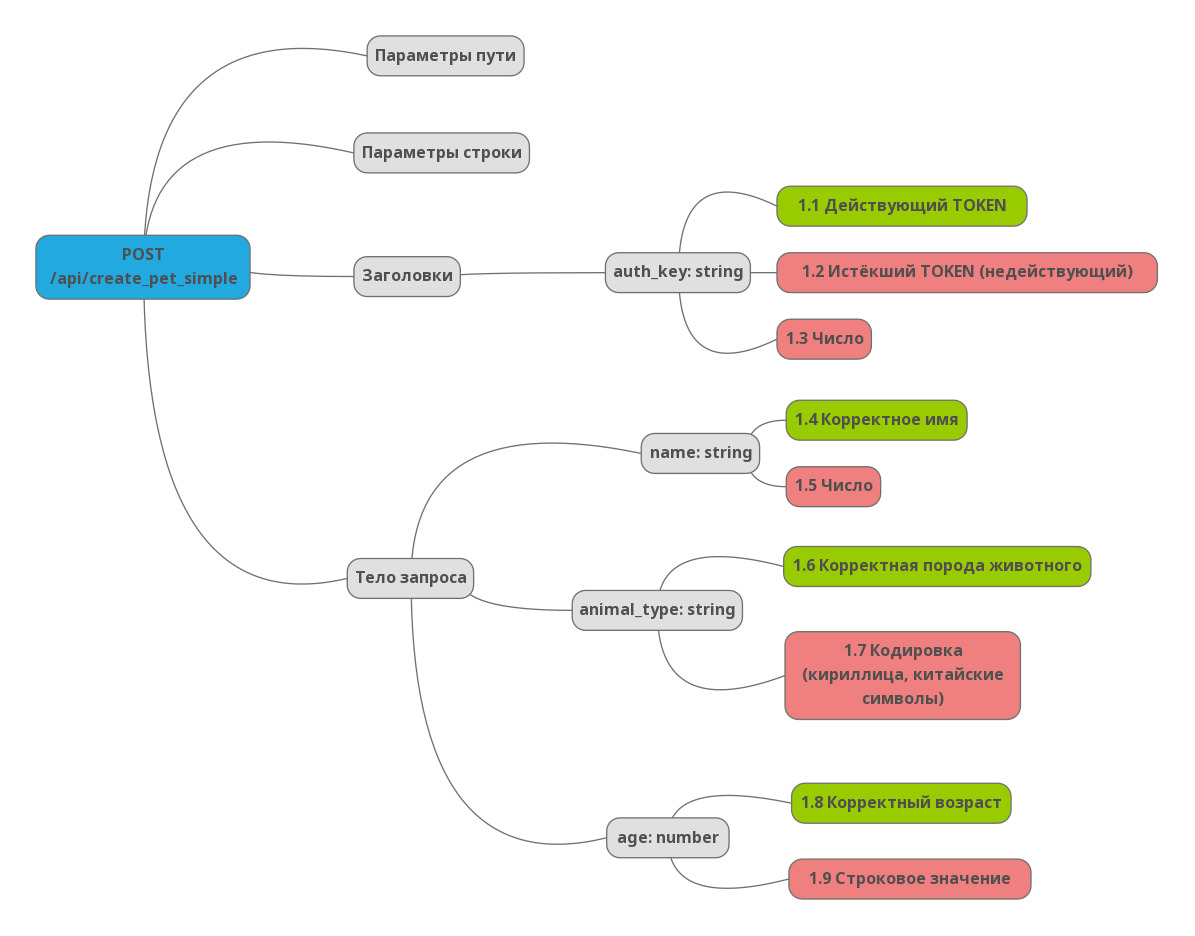
**ЗАДАНИЕ №1.**

**# 1 POST**[**/api​/create\_pet\_simple**](https://petfriends.skillfactory.ru/apidocs/#/default/post_api_create_pet_simple) **- add information about new pet without photo.**

1.1 Диаграмма параметров метода POST/api/create\_pet\_simple для разработки тест-кейсов:

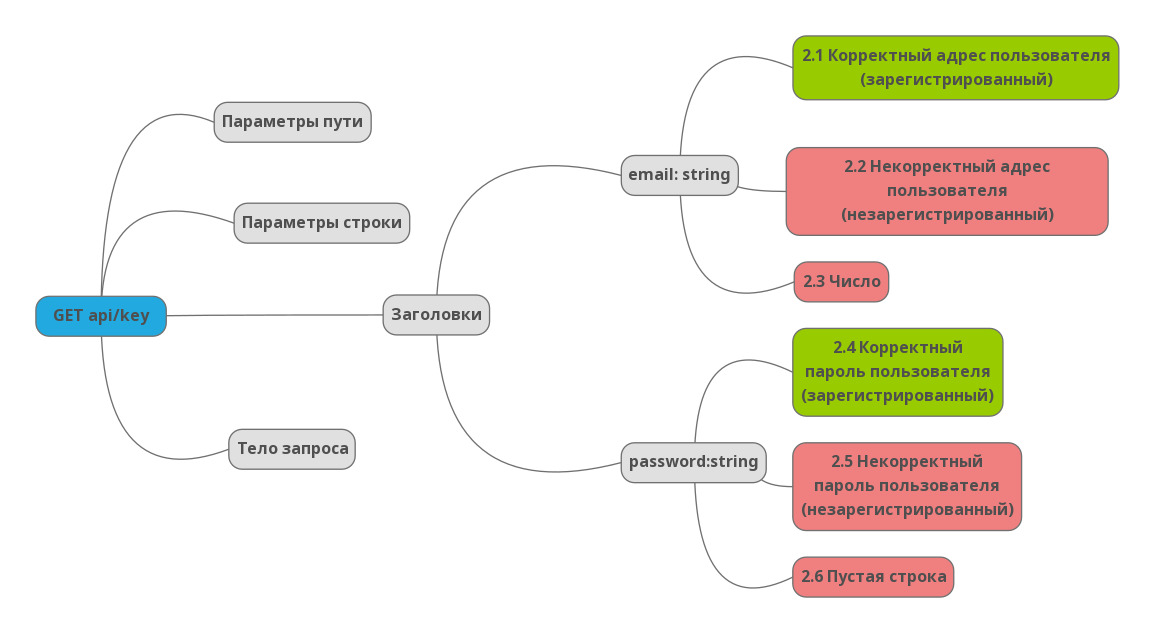
****

1.2 Разработанные тест-кейсы для метода POST/api/create\_pet\_simple на основе Диаграммы 1.1:

| **API вызов** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- |
| **1.1**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "2",  "animal\_type": "дворовый",  "created\_at": "1684850749.1247706",  "id": "1510cb6c-acb2-4b7a-9f3b-0a312cf2d28a",  "name": "Дружок",  "pet\_photo": "",  "user\_id": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 294 * content-type: application/json * date: Tue, 23 May 2023 14:05:49 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца без фото. |
| **1.2**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Истёкший TOKEN (недействующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2 | Код ответа: 403 (Forbidden)  **В теле приходит HTML файл с уведомлением о неверном auth\_key:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>403 Forbidden</title>  <h1>Forbidden</h1>  <p>Please provide &#x27;auth\_key&#x27; Header</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 154 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 06:54:43 GMT * vary: OriginT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест с некорректным вводом истекшего TOKEN. |
| **1.3**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: числовое значение без буквенных символов  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2 | Код ответа: 403 (Forbidden)  **В теле приходит HTML файл с уведомлением о неверном auth\_key:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>403 Forbidden</title>  <h1>Forbidden</h1>  <p>Please provide &#x27;auth\_key&#x27; Header</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 154 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 07:16:08 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест с некорректным API-ключом. |
| **1.4**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "2",  "animal\_type": "дворовый",  "created\_at": "1684913805.1604013",  "id": "e5780d87-5667-40ed-bfb7-cf3bc0d46d81",  "name": "Дружок",  "pet\_photo": "",  "user\_id": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 294 * content-type: application/json * date: Wed, 24 May 2023 07:36:45 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “name” в теле запроса. |
| **1.5**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: : Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: 12345678910;  animal\_type: “дворовый”;  age: 2 | Код ответа: 200 (Created)  **В теле приходит text/html файл с уведомлением о успешном добавлении питомца в БД (pet was successfully added to the database)**  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 4791 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 07:44:21 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на добавление питомца с некорректным параметром “name” (число) в теле запроса.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **1.6**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Рекс”;  animal\_type: “дворовый”;  age: 12. | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "12",  "animal\_type": "дворовый",  "created\_at": "1684914979.0609636",  "id": "27f33137-af0c-43b5-b426-9618e69bfbc4",  "name": "Рекс",  "pet\_photo": "",  "user\_id": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 283 * content-type: application/json * date: Wed, 24 May 2023 07:56:19 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “animal\_type” в теле запроса. |
| **1.7**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Рекс”;  animal\_type: “名叫雷克斯的狗”;  age: 12. | Код ответа: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "12",  "animal\_type": "名叫雷克斯的狗",  "created\_at": "1684915289.2587857",  "id": "b2e9f079-5a6b-4d2d-8bef-01e1fd2af3e1",  "name": "Рекс",  "pet\_photo": "",  "user\_id": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 277 * content-type: application/json * date: Wed, 24 May 2023 08:01:29 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на добавление питомца с некорректной кодировкой параметра “animal\_type” (китайские иероглифы) в теле запроса.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **1.8**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Рекс”;  animal\_type: “дворовый”;  age: 12. | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "12",  "animal\_type": "дворовый",  "created\_at": "1684915625.675662",  "id": "e82cac97-7436-430b-b5de-e5f2884675f5",  "name": "Рекс",  "pet\_photo": "",  "user\_id": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 282 * content-type: application/json * date: Wed, 24 May 2023 08:07:05 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “age” (number) в теле запроса. |
| **1.9**  **POST**  <https://petfriends.skillfactory.ru/api/create_pet_simple>  **Headers:**  auth\_key: Действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Рекс”;  animal\_type: “дворовый”;  age: двенадцать. | **Код ответа**: 403 (Data is incorrect)  **Платформа Flasger не отправляет запрос к эндпоинту https://petfriends.skillfactory.ru/api/create\_pet\_simple с некорректным параметром “age” (строка) в теле запроса.**  **Заголовки ответа:**  Обязательно:   * —-------------------------   Опционально:   * —-------------------------   **Время выполнения:** —------------------- | Негативный тест на добавление питомца с некорректным параметром “age” (строка) в теле запроса. |

**# 2 GET**[**/api​/key**](https://petfriends.skillfactory.ru/apidocs/#/default/get_api_key) **- get API key**

2.1 Диаграмма параметров метода GET/api​/key для разработки тест-кейсов:

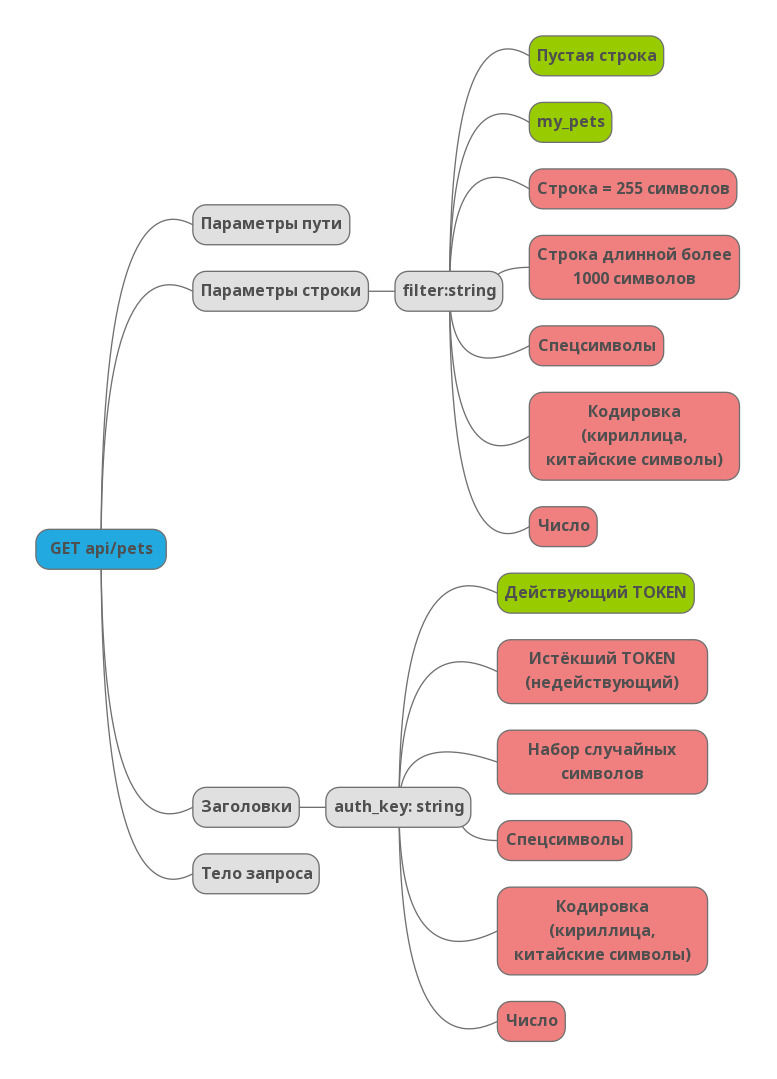
****

2.2 Разработанные тест-кейсы для метода GET/api​/key - Get API key на основе Диаграммы 2.1:

| **2.1**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = корректный почтовый ящик пользователя (зарегистрированный)  password = корректный пароль пользователя (зарегистрированный) | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "key": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: \* * content-length: 67 * content-type: application/json * date: Wed, 24 May 2023 09:22:11 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на верифицированный вход с корректным email. |
| --- | --- | --- |
| **2.2**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = некорректный почтовый ящик пользователя (незарегистрированный)  password = корректный пароль пользователя (зарегистрированный) | **Код ответа**: 403 (Forbidden)  **В теле приходит html файл с корректным сообщением об отсутствии пользователя в БД:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>403 Forbidden</title>  <h1>Forbidden</h1>  <p>This user wasn&#x27;t found in database</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: \* * content-length: 151 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 09:35:23 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на вход с незарегистрированным email. |
| **2.3**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = число(number)  password = корректный пароль пользователя (зарегистрированный) | **Код ответа**: 403 (Forbidden)  **В теле приходит html файл с корректным сообщением об отсутствии пользователя в БД:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>403 Forbidden</title>  <h1>Forbidden</h1>  <p>This user wasn&#x27;t found in database</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: \* * content-length: 151 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 09:41:44 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на вход с некорректным значением email (number вместо string). |
| **2.4**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = корректный почтовый ящик пользователя (зарегистрированный)  password = корректный пароль пользователя (зарегистрированный) | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "key": "d302d6b548a711b9dd96a4b5f741b0888179538e8fe1d039a7eb875e"  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: \* * content-length: 67 * content-type: application/json * date: Wed, 24 May 2023 09:44:19 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на верифицированный вход с корректным password. |
| **2.5**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = корректный почтовый ящик пользователя (зарегистрированный))  password = некорректный пароль пользователя (незарегистрированный) | **Код ответа**: 403 (Forbidden)  **В теле приходит html файл с корректным сообщением об отсутствии пользователя в БД:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>403 Forbidden</title>  <h1>Forbidden</h1>  <p>This user wasn&#x27;t found in database</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: \* * content-length: 151 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 09:46:07 GMT   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на вход с незарегистрированным password. |
| **2.6**  **GET**  <https://petfriends.skillfactory.ru/api/key>  **Headers:**  email = корректный почтовый ящик пользователя (зарегистрированный))  password = некорректный пароль пользователя (пустая строка) | **Код ответа**: 400 (Data is incorrect)  **Платформа Flasger не отправляет запрос к эндпоинту https://petfriends.skillfactory.ru/api/key с незаполненным полем “passsword”.**  **Заголовки ответа:**  Обязательно:   * —-------------------------   Опционально:   * —-------------------------   **Время выполнения:** —------------------- | Негативный тест на вход с незаполненным полем (пустая строка) password. |

**# 3 GET** [**/api​/pets**](https://petfriends.skillfactory.ru/apidocs/#/default/get_api_pets) **- Get list of pets.**

3.1 Диаграмма параметров метода GET /api/pets для разработки тест-кейсов:

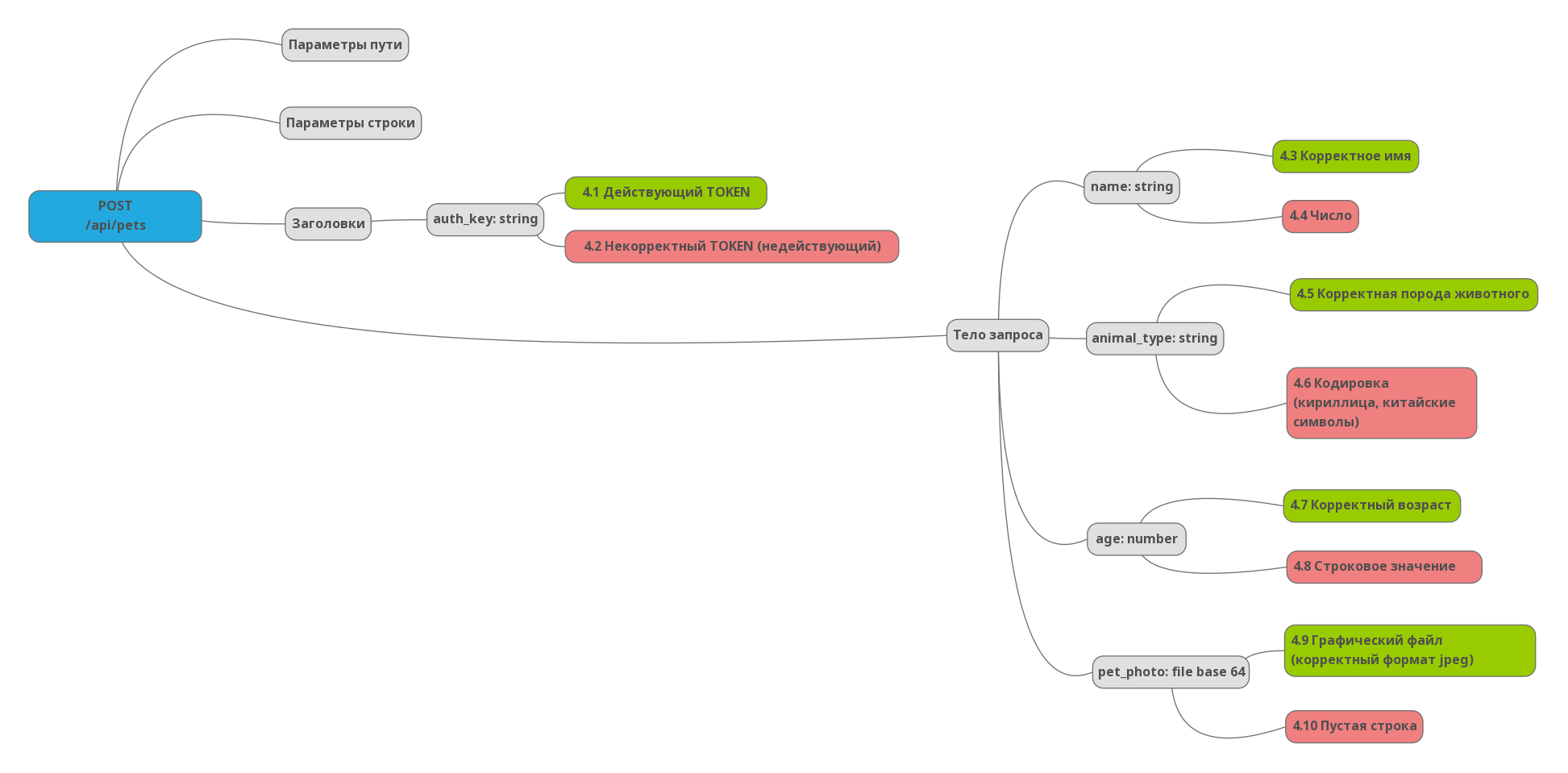


3.2 Разработанные тест-кейсы для метода GET /api/pets - Get list of pets на основе Диаграммы 3.1:

| **API вызов** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- |
| № 1  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter = ' ' (пустая строка) | * Заголовки ответа содержат:   + Обязательно: content-type: text/json; charset=utf-8; date: <дата>;   + Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * Код ответа 200 (OK) * В теле приходит json с корректной структурой * В теле приходят следующие поля в объектах pets:   + age: целое число,   + animal\_type: строка,   + created\_at: строка со снимком времени (timestamp),   + id: uuid,   + name: строка,   + pet\_photo: base64 image,   + user\_id: guid * Время выполнения: не дольше 1с |  |
| № 2  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter = “my\_pets” | * Код ответа 200 (OK) * Содержит список питомцев, принадлежащих пользователю-владельцу ключа API * Заголовки ответа те же, что в тесте #1 * Время выполнения: не дольше 1с | Позитивный тест с опциональными  параметрами.  Так как здесь мы в строке  запроса используем параметр filter,  который не является обязательным,  и устанавливаем его значение =my\_pets.  Мы всё ещё ожидаем, что запрос корректно  выполнится, поэтому сценарий позитивный.  Негативных тестов с корректным вводом  на данном запросе быть не может.  Параметр filter является строкой, а значит любое корректное значение приведет к позитивному исходу запроса с  этим параметром. Оставшийся параметр заголовка  auth\_key обязан быть корректным в этой категории,  т.е. негативных ожиданий мы не сможем вызвать. |
| № 3  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: 5  Query params:  filter=my\_pets | * Код ответа 403 (Forbidden) * Тело ответа: HTML-код c сообщением о неавторизованном пользователе * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Негативный, потому что мы ожидаем  в результатах выполнения ошибку, а с некорректным вводом, потому что auth\_key является string типом, а мы пытаемся передать int. |
| № 4  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: abc  Query params:  filter=my\_pets | * Код ответа 403 (Forbidden) * Тело ответа: HTML-код c сообщением о неавторизованном пользователе * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Негативный, потому что мы ожидаем в результатах выполнения ошибку, а с некорректным вводом, потому что auth\_key не может быть abc, это более длинная строка. |
| № 5  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=pets | * Код ответа 400 (Bad Request) * Тело ответа: HTML код c сообщением о неверном значении параметра filter * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Негативный, потому что мы ожидаем  в результатах выполнения ошибку,  с некорректным вводом, потому что  мы передаем в filter значение,  непредусмотренное системой. |
| № 6  PUT/PATCH/DELETE  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=my\_pets | * Код ответа 405 (Method not allowed) * Тело ответа: HTML-код c сообщением о недопустимом HTTP-методе * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Деструктивное тестирование.  Мы пытаемся сломать систему,  вызывая известный эндпоинт  с неподдерживаемым типом запроса. |
| № 7  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  - auth\_key: корректный ключ API  - Content-Type: application/xml; application/x-www-form-urlencoded; application/form-data; text/plain  Query params:  filter=my\_pets | * Код ответа 415 (Unsupported content-type) * Тело ответа: HTML код c сообщением о недопустимом формате данных * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Здесь через ";" разделены  разные значения заголовка Content-Type.  То есть этот кейс повторяется 4 раза  для каждого значения Content-Type.  Этот тест также является разрушительным,  так как мы пытаемся передать неподдерживаемый  тип данных в запрос. |
| № 8  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=строка > 256 символов  filter=строка спецсимволов  filter=строка символов из кириллицы | * Код ответа 400 (Bad Request) * Тело ответа: HTML-код c сообщением о неверном значении параметра filter. * Заголовки ответа   + Обязательно:     - content-type: text/html charset=utf-8     - date: <дата>   + Опционально:     - connection: keep-alive     - server: gunicorn/20.0.4     - via: 1.1 vegur * Время выполнения: не дольше 1с | Этот тест является разрушительным.  Мы пытаемся привести наше приложение  к падению — строка длиннее,  чем может вмещать в себя тип данных string ,  спецсимволы некорректно обрабатываются  на стороне сервера, кириллица неверно обрабатывается на стороне сервера. |

**# 4 POST** [/api​/pets](https://petfriends.skillfactory.ru/apidocs/#/default/post_api_pets) **- add information about new pet.**

4.1 Диаграмма параметров метода POST /api​/pets для разработки тест-кейсов:

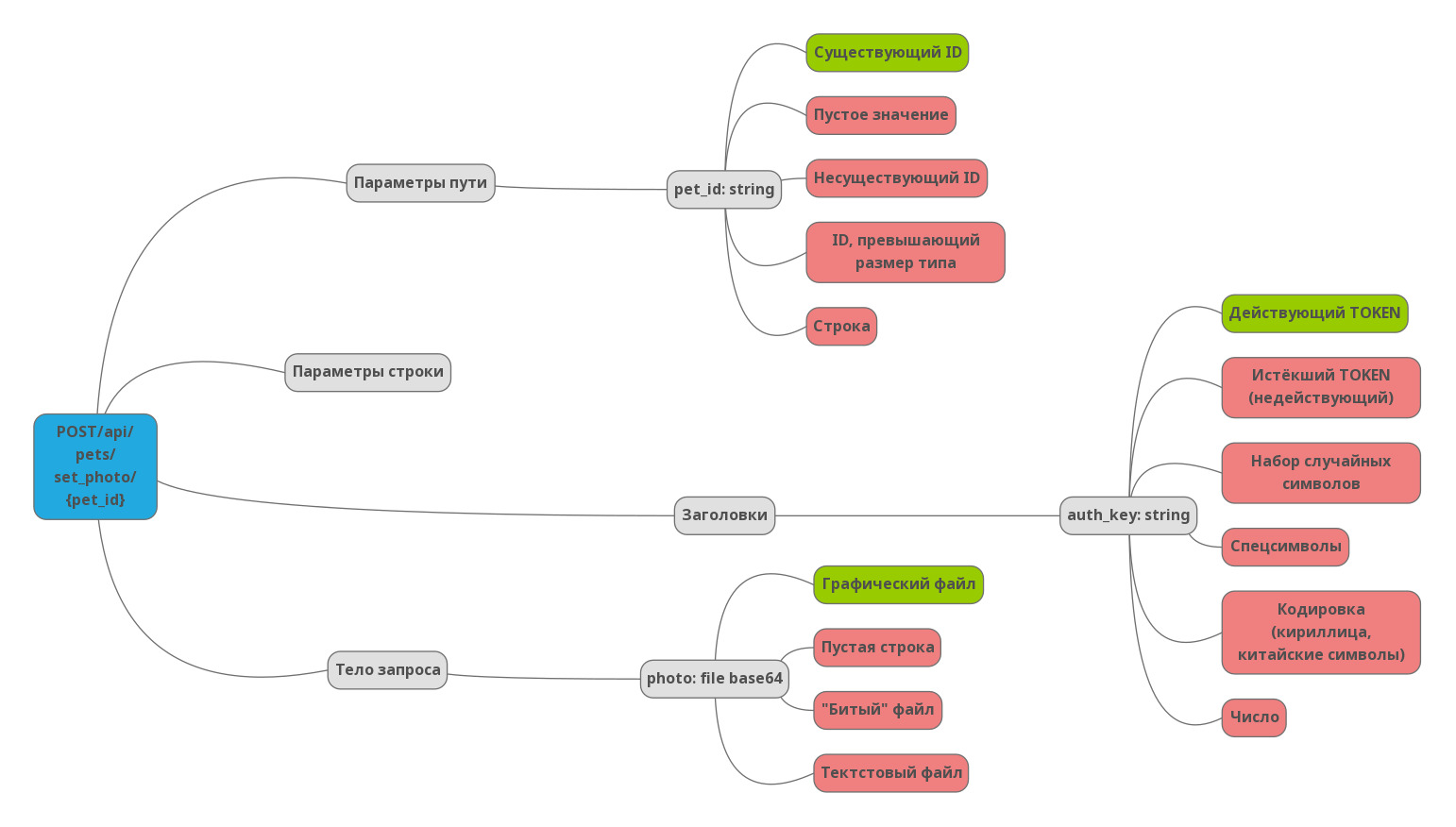


4.2 Разработанные тест-кейсы для метода POST /api/pets - Add information about new pet на основе Диаграммы 4.1:

| **API вызов** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- |
| **4.1**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: действующий TOKEN (корректный ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "\_id": "",  "age": "2",  "animal\_type": "дворовый",  "created\_at": "1684934767.5875218",  "id": "b87568a1-5f1f-4186-b406-f8757aa66416",  "name": "Дружок",  "pet\_photo": "data:image/jpeg;base64  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 12909 * content-type: application/json * date: Wed, 24 May 2023 13:26:07 GMT * vary: Origin   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление всех данных питомца. |
| **4.2**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: некорректный TOKEN (недействующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | Код ответа: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody> ………………………………………………………………….  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 65869 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 13:36:56 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест с некорректным TOKEN (недействующий ключ API).  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **4.3**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody> ………………………………………………………………….  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 32445 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 13:44:53 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “name” в теле запроса. |
| **4.4**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: 12345”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  <tr>  <th scope="row"><img src="data:image/jpeg;base64,......  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 67615 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 13:49:32 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на добавление питомца с некорректным параметром “name”(number) в теле запроса.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему |
| **4.5**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  <tr>  <th scope="row"><img src="data:image/jpeg;base64,......  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 35486 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 13:54:50 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “animal\_type” в теле запроса. |
| **4.6**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type:”名叫雷克斯的狗”;  age: 2  pet\_photo: графический файл .jpeg | Код ответа: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  <tr>  <th scope="row"><img src="data:image/jpeg;base64,......  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 53083 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 14:00:29 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на добавление питомца с некорректной кодировкой параметра “animal\_type” (китайские иероглифы) в теле запроса.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **4.7**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  <tr>  <th scope="row"><img src="data:image/jpeg;base64,......  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 35486 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 13:54:50 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “age” в теле запроса. |
| **4.8**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: два  pet\_photo: графический файл .jpeg | **Код ответа**: 400 (Data is incorrect)  **Платформа Flasger не отправляет запрос к эндпоинту https://petfriends.skillfactory.ru/api/pets с некорректным параметром “age” (строка) в теле запроса.**  **Заголовки ответа:**  Обязательно:   * —-------------------------   Опционально:   * —-------------------------   **Время выполнения:** —------------------- | Негативный тест на добавление питомца с некорректным параметром “age” (строка) в теле запроса. |
| **4.9**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: графический файл .jpeg | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением о создании карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  <tr>  <th scope="row"><img src="data:image/jpeg;base64,......  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 80275 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 14:10:41 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на добавление питомца с корректным параметром “pet\_photo” в теле запроса. |
| **4.10**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “дворовый”;  age: 2  pet\_photo: нет (пустая строка) | **Код ответа**: 400 (Data is incorrect)  **Платформа Flasger не отправляет запрос к эндпоинту https://petfriends.skillfactory.ru/api/pets с некорректным параметром “pet\_photo” (пустая строка) в теле запроса..**  **Заголовки ответа:**  Обязательно:   * —-------------------------   Опционально:   * —-------------------------   **Время выполнения:** —------------------- | Негативный тест на добавление питомца с некорректным параметром “pet\_photo” (пустая строка) в теле запроса. |

**# 5 POST** [**​/api​/pets​/set\_photo​/{pet\_id}**](https://petfriends.skillfactory.ru/apidocs/#/default/post_api_pets_set_photo__pet_id_) **- add photo of pet.**

5.1 Диаграмма параметров метода POST /api/pets/set\_photo/{pet\_id} для разработки тест-кейсов:

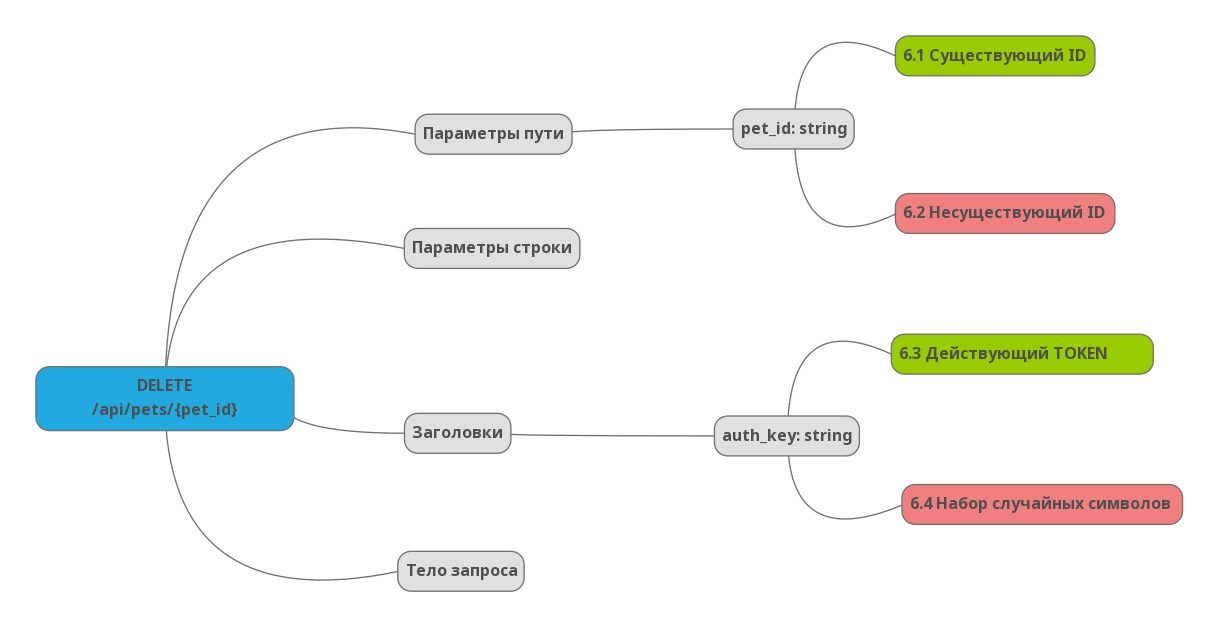


5.2 Разработанные тест-кейсы для метода POST /api/pets/set\_photo/{pet\_id} на основе Диаграммы 5.1:

| **API вызов** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- |
| **POST**  [**https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id**](https://petfriends.skillfactory.ru/api/pets/set_photo/%7Bpet_id)**}**  **Headers:**  **auth\_key: Корректный ключ API**  **Path params:**  **pet\_id=корректный id питомца**  **Payload:**  **корректный файл** | Код ответа 200 (Created)  В теле приходит json с корректной структурой  В теле приходят следующие поля в объектах pets:  age: целое число,  animal\_type: строка,  created\_at: строка со снимком времени (timestamp)  id: uuid  name: строка  pet\_photo: base64 image  user\_id: guid  Заголовки ответа:  Обязательно:  content-type:application/json charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1с | Базовый позитивный тест |
| **POST**  **https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}**  **Headers:**  **auth\_key: Корректный ключ API**  **Path params:**  **pet\_id=корректный id питомца**  **Payload:**  **текстовый файл** | Код ответа 400 (Bad Request)  В теле ответа HTML код c сообщением о некорректном запросе  Заголовки ответа  Обязательно:  content-type:text/html charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1с | Негативный тест с корректным вводом |
| **POST**  https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id=некорректный id питомца  **Payload:**  графический файл | Код ответа 404 (Not Found)  В теле ответа HTML код c сообщением о том, что питомец с таким id не найден  Заголовки ответа  Обязательно:  content-type: text/html charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1с | Негативный тест с корректным вводом |
| **POST**  **https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}**  **Headers:**  **auth\_key: Корректный ключ API**  **Path params:**  **pet\_id=корректный id питомца**  **Payload:**  **загрузить фотографию второй раз** | Код ответа 400 (Bad Request) (здесь будьте внимательны и читайте спецификацию системы, несколько фото одного питомца могут быть допустимы, для нашего примера мы не даем такой возможности)  В теле ответа HTML код c сообщением о том, что фотография уже загружена  Заголовки ответа  Обязательно:  content-type: text/html charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1 | Негативный тест с корректным вводом |
| **POST**  **https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}**  **Headers:**  **auth\_key: Корректный ключ API**  **Path params:**  **pet\_id=корректный id питомца**  **Payload:**  **отсутствует** | Код ответа 400 (Bad Request)  В теле ответа HTML код c сообщением о том, что отсутствует файл  Заголовки ответа  Обязательно:  content-type: text/html charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1с | Негативный тест с некорректным вводом |
| **POST**  **https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}**  **Headers:**  **auth\_key: Корректный ключ API**  **Path params:**  **pet\_id=корректный id питомца**  **Payload:**  **“битый” файл (в** [**base64**](https://developer.mozilla.org/ru/docs/Glossary/Base64) **представлении изменить несколько символов)** | Код ответа 400 (Bad Request)  В теле ответа HTML код c сообщением об испорченном файле  Заголовки ответа  Обязательно:  content-type: text/html charset=utf-8  date: <дата>  Опционально:  connection: keep-alive  server: gunicorn/20.0.4  via: 1.1 vegur  Время выполнения: не дольше 1с | Деструктивное тестирование |

**# 6 DELETE** [/api​/pets​/{pet\_id}](https://petfriends.skillfactory.ru/apidocs/#/default/delete_api_pets__pet_id_) **- delete pet from database.**

6.1 Диаграмма параметров метода DELETE ​/api​/pets​/{pet\_id} - delete pet from database для разработки тест-кейсов:

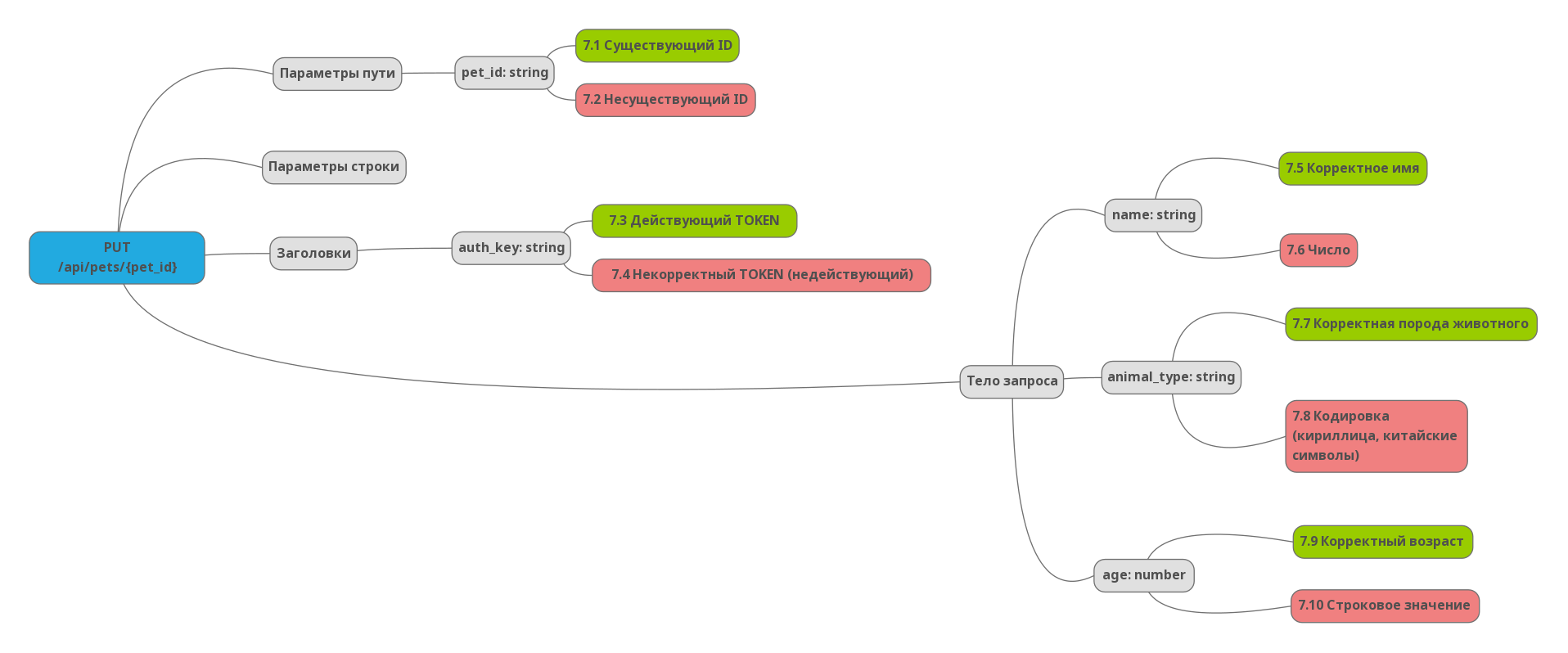


6.2 Разработанные тест-кейсы для метода DELETE ​/api​/pets​/{pet\_id} на основе Диаграммы 6.1:

| **6.1**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Path params:**  pet\_id=корректный id питомца | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением об удалении ранее добавленного питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  </tbody>  </table>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 309 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 15:02:40 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на удаление карточки питомца с существующим id. |
| --- | --- | --- |
| **6.2**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: набор случайных символов  **Path params:**  pet\_id: несуществующий id питомца | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением об удалении несуществующей id карточки питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  </tbody>  </table>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 309 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 15:07:06 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на удаление карточки питомца с не существующим id.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему |
| **6.3**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Path params:**  pet\_id=корректный id питомца | **Код ответа**: 200 (Created)  **В теле приходит text/html файл с уведомлением об удалении ранее добавленного питомца:**  <table class="table table-hover">  <thead>  <tr>  <th scope="col">Фото</th>  <th scope="col">Имя</th>  <th scope="col">Порода</th>  <th scope="col">Возраст</th>  <th scope="col" class="smart\_cell"></th>  </tr>  </thead>  <tbody>  </tbody>  </table>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 309 * content-type: text/html; charset=utf-8 * date: Wed, 24 May 2023 15:11:45 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на удаление карточки питомца с корректным API ключом. |
| **6.4**  **POST**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  auth\_key: некорректный TOKEN  (набор случайных символов)  **Path params:**  pet\_id: несуществующий id питомца | **Код ответа**: 403 (Undocumented)  **В ответе сервера приходит сообщение - TypeError: Failed to execute 'fetch' on 'Window': Failed to read the 'headers' property from 'RequestInit': String contains non ISO-8859-1 code point. Отказ в удалении из-за ошибки ввода (TypeError). Система сработала корректно.**  **Заголовки ответа:**   * —--------------------------------   **Время выполнения:** не дольше 1с | Негативный тест на удаление карточки питомца с некорректным API ключом (набор случайных символов). |

**# 7 PUT** [/api​/pets​/{pet\_id}](https://petfriends.skillfactory.ru/apidocs/#/default/put_api_pets__pet_id_) **- update information about pet.**

7.1 Диаграмма параметров метода PUT /api/pets/{pet\_id} - update information about pet для разработки тест-кейсов:



7.2 Разработанные тест-кейсы для метода PUT ​/api​/pets​/{pet\_id} на основе Диаграммы 7.1:

| **7.1**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Андрюха”;  animal\_type: “домашний”;  age: 50 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "50",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Андрюха",  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15863 * content-type: application/json * date: Thu, 25 May 2023 07:38:44 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на обновление данных карточки питомца с существующим id. |
| --- | --- | --- |
| **7.2**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/79c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Андрюха”;  animal\_type: “домашний”;  age: 50 | **Код ответа**: 400 (Bad Request)  **В теле приходит text/html файл с корректным сообщением о том, что питомец с некорректным id не найден:**  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  <title>400 Bad Request</title>  <h1>Bad Request</h1>  <p>Pet with this id wasn&#x27;t found!</p>  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 151 * content-type: text/html; charset=utf-8 * date: Thu, 25 May 2023 07:52:01 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на добавление на обновление данных карточки питомца с несуществующим id |
| **7.3**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “домашний”;  age: 8 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "8",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Дружок",,  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15856 * content-type: application/json * date: Thu, 25 May 2023 08:04:07 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на обновление данных карточки питомца с корректным API ключом. |
| **7.4**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: некорректный TOKEN (недействующий ключ API)  **Body(formData):**  name: “Дружок”;  animal\_type: “домашний”;  age: 13 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "13",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Дружок",  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15857 * content-type: application/json * date: Thu, 25 May 2023 08:10:56 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на обновление данных карточки питомца с некорректным API ключом.  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему |
| **7.5**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Егорка”;  animal\_type: “домашний”;  age: 10 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "10",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Егорка",  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15857 * content-type: application/json * date: Thu, 25 May 2023 08:17:52 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на обновление данных карточки питомца с корректным параметром “name” в теле запроса. |
| **7.6**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: 123456;  animal\_type: “домашний”;  age: 10 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "10",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "123456",  "pet\_photo": "data:image/jpeg;base64,………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15827 * content-type: application/json * date: Thu, 25 May 2023 08:21:27 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на обновление данных карточки питомца с некорректным параметром “name” в теле запроса (number).  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **7.7**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Валера”;  animal\_type: “домашний”;  age: 11 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "11",  "animal\_type": "домашний",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Валера",  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15857 * content-type: application/json * date: Thu, 25 May 2023 08:26:17 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на обновление данных карточки питомца с корректным параметром “animal\_type” в теле запроса. |
| **7.8**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Валера”;  animal\_type: “名叫雷克斯的狗”;  age: 11 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "11",  "animal\_type": "名叫雷克斯的狗",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Валера",  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15851 * content-type: application/json * date: Thu, 25 May 2023 08:30:11 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Негативный тест на обновление данных карточки питомца с некорректным параметром “animal\_type” в теле запроса (кодировка, китайские иероглифы).  ОБНАРУЖЕНА ОШИБКА:  Создать баг-репорт и завести баг в баг-трекинговую систему. |
| **7.9**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Матроскин”;  animal\_type: “полосатый”;  age: 10 | **Код ответа**: 200 (Created)  **В теле приходит json с корректной структурой:**  {  "age": "10",  "animal\_type": "полосатый",  "created\_at": "1684941496.0510638",  "id": "78c76bb6-96a7-4c54-85fb-7dfb9efe80da",  "name": "Матроскин"  "pet\_photo": "data:image/jpeg;base64………  }  **Заголовки ответа:**  Обязательно:   * access-control-allow-origin: https://petfriends.skillfactory.ru * content-length: 15881 * content-type: application/json * date: Thu, 25 May 2023 08:36:39 GMT * vary: Origin, Cookie   Опционально:   * connection: keep-alive * server: gunicorn/20.0.4 * via: 1.1 vegur   **Время выполнения:** не дольше 1с | Базовый позитивный тест на обновление данных карточки питомца с корректным параметром “age” в теле запроса. |
| **7.10**  **PUT**  <https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da>  **Path params:**  pet\_id=корректный id питомца  **Headers:**  auth\_key: корректный TOKEN (действующий ключ API)  **Body(formData):**  name: “Матроскин”;  animal\_type: “полосатый”;  age: “пятнадцать” | **Код ответа**: —-----------------  **Платформа Flasger не отправляет запрос к эндпоинту https://petfriends.skillfactory.ru/api/pets/78c76bb6-96a7-4c54-85fb-7dfb9efe80da с некорректным параметром “age” (строковое значение) в теле запроса.**  **Сообщение в консоли разработчика:**  **<input type="text" class="invalid" title="Value must be a number" placeholder="age - The new age of the pet" value="пятнадцать">**  **Ошибка ввода данных, система сработала корректно.**  **Заголовки ответа:**  Обязательно:   * —-------------------------   Опционально:   * —-------------------------   **Время выполнения:** —------------------- | Негативный тест на обновление данных карточки питомца с некорректным параметром “age” в теле запроса (строковое значение). |

**ЗАДАНИЕ №2.**

**Пользовательские сценарии для**

**API-тестирования платформы Pet Friends.**

1. **“Как клиент сервиса, я хочу создать простую карточку нового питомца без фотографии”.**

| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| **Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. 2. Выполнить запрос создания питомца - POST /api​/create\_pet\_simple с корректными данными. Получаем id созданного пользователем питомца. |
| **Ожидание** | В ответе на запрос GET /api/pets приходит созданный питомец с корректно загруженными данными из шага 2. |

1. **“Как клиент сервиса, я хочу создать карточку нового питомца и загрузить его фотографию”.**

| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| **Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. 2. Выполнить запрос создания питомца — POST /api/pets с корректными данными. Получаем id созданного пользователем питомца. 3. Выполнить загрузку фотографии — POST /api/pets/set\_photo/{pet\_id}. |
| **Ожидание** | В ответе на запрос GET /api/pets приходит созданный питомец с корректно загруженным файлом из шага 3 и данными питомца. |

1. **“Как клиент сервиса, я хочу получить список всех созданных мной карточек питомцев”.**

| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| **Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. 2. Выполнить запрос создания питомца — POST /api/pets с корректными данными. Получаем id созданного пользователем питомца. 3. Выполнить загрузку фотографии — POST /api/pets/set\_photo/{pet\_id}. |
| **Ожидание** | В ответе на запрос получения списка питомца(ев) — GET /api/pets с выбранным параметром фильтра (строки) - ‘my\_pets’, получаем список созданных пользователем питомцев. |

1. **“Как клиент сервиса, я хочу создать карточку нового питомца, а затем удалить ее из списка моих питомцев”.**

| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| **Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. 2. Выполнить запрос создания питомца — POST /api/pets с корректными данными. Получаем id созданного пользователем питомца. 3. Выполнить загрузку фотографии — POST /api/pets/set\_photo/{pet\_id}. 4. Выполнить запрос получения списка питомца(ев) — GET /api/pets с выбранным параметром фильтра (строки) - ‘my\_pets’ и убедиться, что данные питомца добавлены в БД сервера. 5. Выполнить запрос на удаление питомца из списка шага 4 - DELETE​/api​/pets​/{pet\_id} с корректно указанным id. |
| **Ожидание** | В ответе на запрос получения списка питомцев пользователя — GET /api/pets с выбранным параметром фильтра (строки) - ‘my\_pets’, убеждаемся, что питомец из шага 5 удален. |

1. **“Как клиент сервиса, я хочу обновить данные в карточке созданного мной питомца”.**

| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| **Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. 2. Выполнить запрос создания простой карточки питомца - POST /api​/create\_pet\_simple с корректными данными. Получаем id созданного пользователем питомца. 3. Выполнить запрос получения списка питомца(ев) — GET /api/pets с выбранным параметром фильтра (строки) - ‘my\_pets’ и убедиться, что данные питомца из шага 2 добавлены в БД сервера. 4. Выполнить запрос на обновление данных питомца из шага 2 — PUT​/api​/pets​/{pet\_id} с новыми, корректными данными. |
| **Ожидание** | В ответе на запрос получения списка питомцев пользователя — GET /api/pets с выбранным параметром фильтра (строки) - ‘my\_pets’, убеждаемся, что данные питомца из шага 2 обновлены в соответствии с новыми параметрами шага 4. |