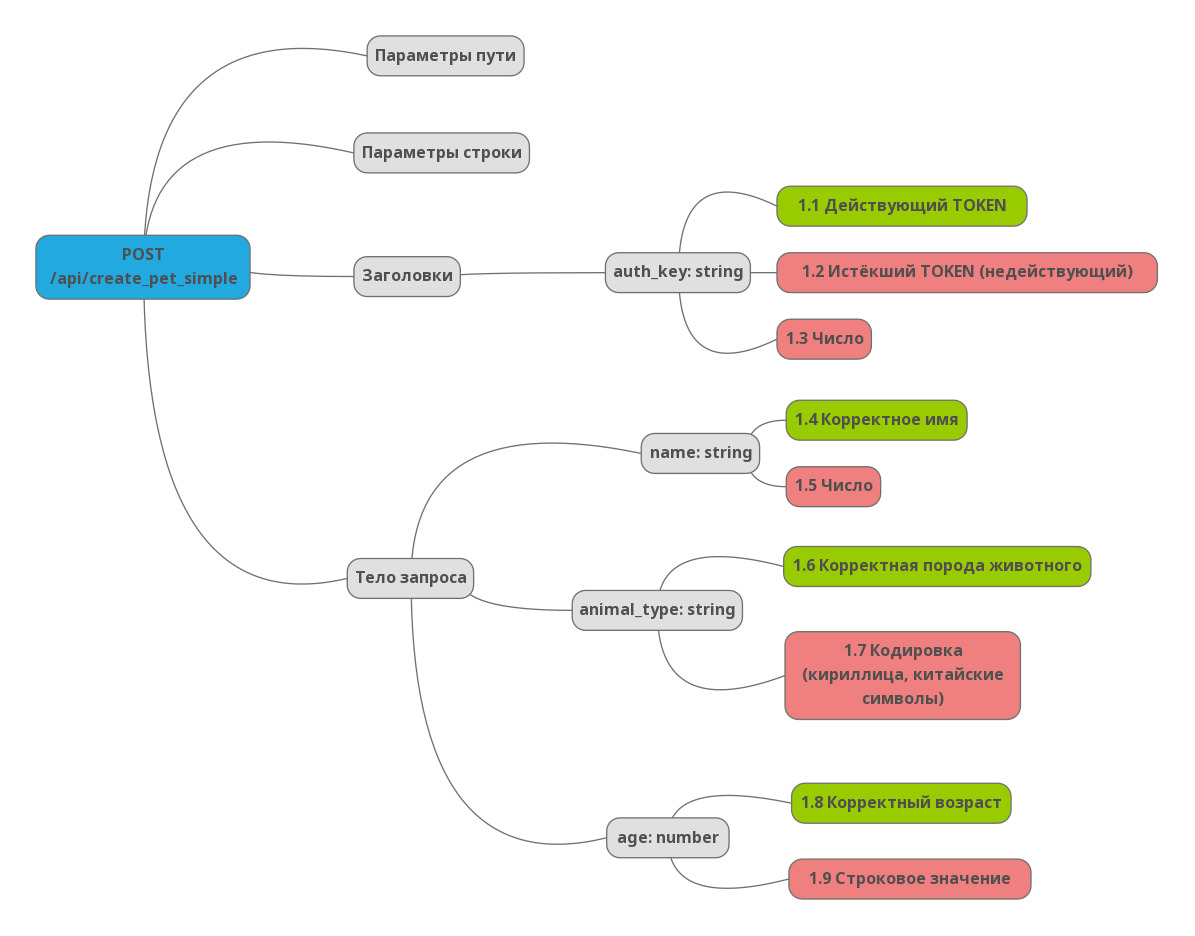
**# 1 POST**[**/api​/create\_pet\_simple**](https://petfriends.skillfactory.ru/apidocs/#/default/post_api_create_pet_simple) **- add information about new pet without photo.**

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

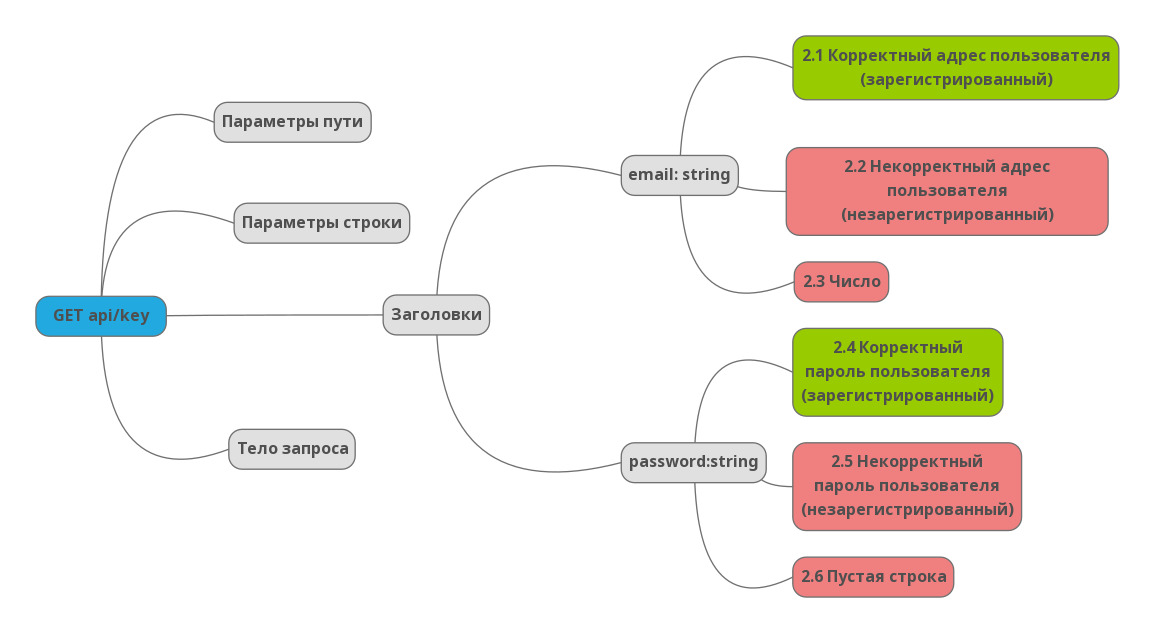
****

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:

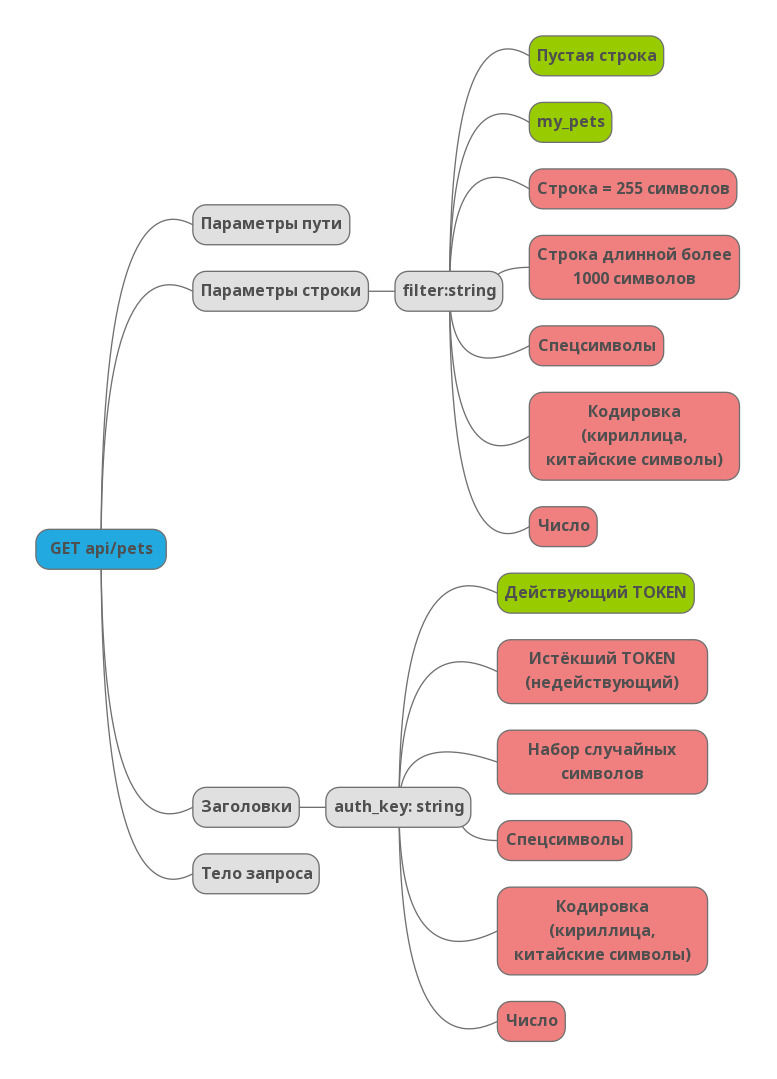
****

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:**  {  "\_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 верифицированный вход с корректным |
| --- | --- | --- |

**# 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:

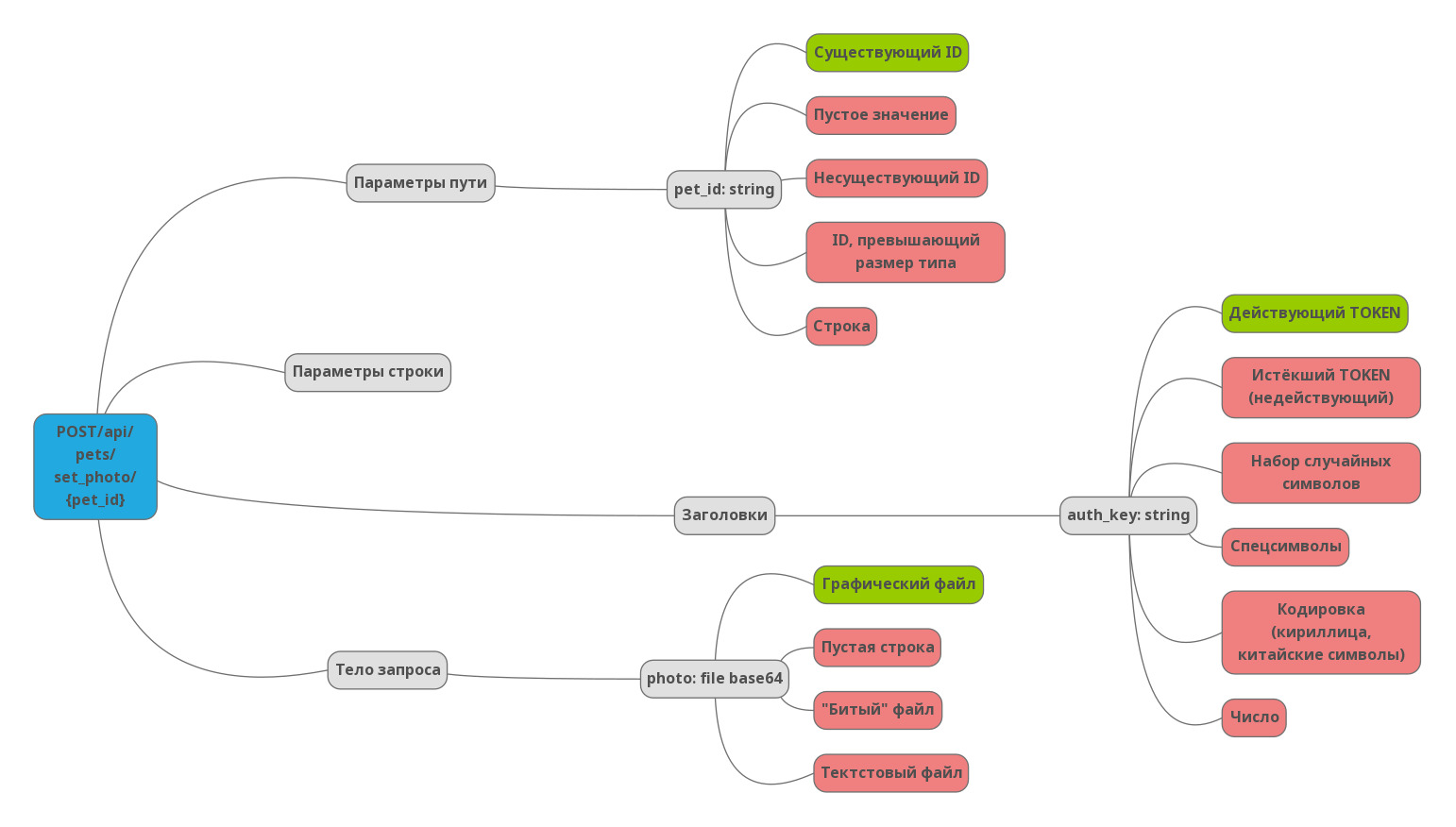


3.2 Разработанные тест-кейсы для метода GET /api/pets - Get list of pets на основе Диаграммы 3.1:

| **API вызов** | **Ожидаемый результат запроса** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- | --- |
| № 1  GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter = ' ' (пустая строка) | * Код ответа   + 200 OK * Заголовки   + Обязательно:     - content-type: application/json     - data: <дата>   + Опционально:     - connection: keep-alive * Тело ответа:   + Корректный json -   поля в теле ответа для сущности pet:   * + - "age": целое число,     - "animal\_type" строка,     - "created\_at": строка со снимком времени,     - "id": uuid,     - "name": строка,     - "pet\_photo": строка в формате base64,     - "user\_id": guid. * Время ответа:   + Меньше 1 сек. | * 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 * Заголовки   + Обязательно:     - content-type: application/json     - data: <дата>   + Опционально:     - connection: keep-alive * Тело ответа:   + Корректный json -   поля в теле ответа для сущности pet:   * + - "age": целое число,     - "animal\_type" строка,     - "created\_at": строка со снимком времени,     - "id": uuid,     - "name": строка,     - "pet\_photo": строка в формате base64,     - "user\_id": guid. * Время ответа:   + Меньше 1 сек. | * 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. |

**# 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** | **Ожидаемый результат** | **Результат выполнения запроса** | **Примечание** |
| --- | --- | --- | --- |
| **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:**  **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 |