| **Вызов API** | **Результат выполнения запроса (ожидаемый)** | **Комментарии** |
| --- | --- | --- |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: malymaks91@gmail.com password: Putgetpost2345 | * + Заголовки ответа содержат:     - Обязательно: content-type: число application/json, date: <дата>.     - Опционально: access-control-allow-origin: \*   + Код ответа 200 (OK)   + В теле приходит json с ключом API   + В теле приходит ‘key’: ключ API (str).   + Время выполнения: не дольше 1 с | Базовый позитивный сценарий. |

| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: MALYMAKS91@GMAIL.COM password: Putgetpost2345 | * + Код ответа 200 (OK)   + Заголовки ответа содержат:     - Обязательно: content-type: число application/json, date: <дата>.     - Опционально: access-control-allow-origin: \*   + Время выполнения: не дольше 1с | Позитивный тест с вводом валидного email в верхнем регистре |
| --- | --- | --- |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: malymaks91@gmail.com password: PutGetPost2345 | * + 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:       * access-control-allow-origin: \*       * content-length: число   + Время выполнения: не дольше 1 с | Негативный тест с некорректным вводом пароля. |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: malymaks91@gmail.com password: 28478475298752 | * + Код ответа 403   + Тело ответа: HTML-код c сообщением о неавторизованном пользователе   + Заголовки ответа     - Обязательно:       * content-type: text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: \*       * content-length: число   + Время выполнения: не дольше 1 с | Негативный тест с некорректным вводом пароля type: int. |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: 123465834314 password: Putgetpost2345 | * + Код ответа 403   + Тело ответа: HTML-код c сообщением о неавторизованном пользователе   + Заголовки ответа     - Обязательно:       * content-type: text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: \*       * content-length: 151   + Время выполнения: не дольше 1 с | Негативный тест с некорректным вводом числа(int) в поле email |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: malymaks91@gmail.com 1) password =строка > 256 символов 2) 1 символ(“j” или “и” или “<”) 3) 120 символов.  4) password = ‘’(пустая str)  5) password =строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–>  6) password =строка символов из кириллицы. (уаоаывотопатвыштыв)  7) Password = китайские иероглифы. (繁体字) | 1, 2, 3, 4):   * + 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:       * access-control-allow-origin: \*       * content-length:число Время выполнения: не дольше 1 с   5) - *Undocumented*  *6)- Undocumented*  *7) - Undocumented* | Негативные тесты на пароль (разные данные, длина) при валидном email |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  password: Putgetpost2345 malymaks91@gmail.com  1) email =строка > 256 символов, 2)1 символ(“j” или “и” или “<”), 3)120 символов. 4) email = ‘’(пустая str) 5) email =строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 6) email =строка символов из кириллицы. (уаоаывотопатвыштыв) 7) email = китайские иероглифы. (繁体字) | 1, 2, 3, 4):   * + Код ответа 403   + Тело ответа: HTML-код c сообщением о неавторизованном пользователе   + Заголовки ответа     - Обязательно:       * content-type: text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: \*       * content-length: 151   + Время выполнения: не дольше 1 с   5) - *Undocumented*  *6)- Undocumented*  *7) - Undocumented* | Негативные тесты на email (разные данные, длина) при валидном пароле |
| **GET**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/key  **Headers:** auth\_key: Корректный ключ API **Formdata:** Без параметров. Предусловие. 1)<https://github.com/QAMax0/AutoTestsApiPetfriends> скачать себе проект. 2) Использовать этот тест test\_get\_api\_key\_for\_valid\_user() в этом файле test\_pet\_friends.py для теста этого запроса. 3) Внести изменения в тест. headers = {} status, result = requests.post("https://petfriends.skillfactory.ru/" + 'api/key'**,** headers=headers) Убрать атрибуты у теста def test\_get\_api\_key\_for\_valid\_user() | * + Код ответа 400 (Bad request)   + Тело ответа: HTML-код c сообщением об ошибке утверждения, что status == 200.   + Время выполнения: не дольше 1 с | Негативный тест-кейс по запросу ключа API. Запрос без параметров. |
| **PUT/PATCH/DELETE**  **https://petfriends.skillfactory.ru/api/key**  **Headers:**  **auth\_key: корректный ключ API** | * + 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>   + Время выполнения: не дольше 1с | Деструктивное тестирование.  Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживаемым типом запроса. |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **Formdata:**  name (str) = имя  animal\_type (str) = тип питомца  age (int) = возраст питомца | * + 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: строка пустая     - name: строка     - pet\_photo: строка пустая     - user\_id: guid   + Заголовки ответа:     - Обязательно:       * content-type:application/json charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: число vary: Origin, Cookie   + Время выполнения: не дольше 1с | Базовый позитивный тест-кейс по созданию питомца без фото |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:** name (str) = строка  animal\_type (str) = строка  age (int) = возраст питомца (число в PyCharm) в тесте добавления питомца без фото Предусловие. 1)<https://github.com/QAMax0/AutoTestsApiPetfriends> скачать себе проект. 2) Использовать этот тест def test\_post\_add\_a\_new\_pet\_without\_a\_photo в этом файле test\_pet\_friends.py. для теста этого запроса. | * + Код ответа 200   + Тело ответа     - {'\_id': str, 'age': **int**, 'animal\_type': str, 'created\_at': str, 'id': str, 'name': str,, 'pet\_photo': пустая str, 'user\_id': 'str’}   + Время выполнения: не дольше 1с | Позитивный тест-кейс age = int  Слева ожидаемый результат. |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: НЕкорректный ключ API   1. Убрать 1 символ из корректного ключа. 2. Добавить 1 символ 3. **Ввести в поле кавычки ‘’** 4. Поменять в корректном ключе буквы на цифры (длина должна сохраниться)   **FormData:**  name (str) = имя  animal\_type (str) = тип питомца  age (int) = возраст питомца | Для тестов 1, 2, 3, 4:   * + Код ответа 403   + Тело ответа: HTML-код c сообщением о неавторизованном пользователе   + Заголовки ответа     - Обязательно:       * content-type: text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: \*       * content-length: число   + Время выполнения: не дольше 1 с | Негативный тест-кейс по созданию питомца без фото с некорректным ключом |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**  name = 1)int, 2)dict(Пример - {name: nik}, 3)List([name, 2])  animal\_type 1)int, 2)dict(Пример - {amimal\_type: dog}, 3)List([animal\_type, pip])  age (int) = возраст питомца (число) | Для всех 3-х тестов:   * + Код ответа 200   + 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: строка пустая     - name: строка     - pet\_photo: строка пустая     - user\_id: строка (ключ API)   + Заголовки ответа     - Обязательно:       * content-type: application/json       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru   + Время выполнения: не дольше 1с | Негативный тест animal\_type, name (оба параметра для каждого теста) =1) int (число) 2) dict(словарь) 3) list (список). |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**  name (str) =  1)строка > 256 символов, 2) 1 символ 3) 120 символов.  animal\_type (str) = 1)строка > 256 символов, 2) 1 символ 3) 120 символов.  age (int) = возраст питомца 1)строка > 256 (число), 2) 1 символ (число) 3) 120 символов.(число). | Для всех 3-х тестов:   * + Код ответа 200   + 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: ID питомца из my\_pets     - name: строка     - pet\_photo: строка пустая     - user\_id: ключ API (строка auth\_key)   + Заголовки ответа     - Обязательно:       * content-type: application/json       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru   + Время выполнения: не дольше 1с | Негативный тест animal\_types, name, age (int) = строка более 256 символом, с 1 или со 120. |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**  name (str) = строка  animal\_type (str) = строка  age (str) = возраст питомца (строка в PyCharm в тесте добавления питомца без фото) | * + 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:       * access-control-allow-origin: https://petfriends.skillfactory.ru   + Время выполнения: не дольше 1с | Негативный тест age = str в Pycharm |
| **POST**  https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**  **name:** 1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字) **animal\_type:**  1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字) **age:** int | Для всех 3-х тестов:   * + Код ответа 200   + 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: ID питомца из my\_pets     - name: строка     - pet\_photo: строка пустая     - user\_id: ключ API (строка auth\_key)   + Заголовки ответа     - Обязательно:       * content-type: application/json       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru   + Время выполнения: не дольше 1с | Негативные тесты на ввод в поля name, animal\_type некорректных значений при age = int |
| **POST**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/create\_pet\_simple  **Headers:** auth\_key: Корректный ключ API **Formdata:** Без параметров. Предусловие. 1)<https://github.com/QAMax0/AutoTestsApiPetfriends> скачать себе проект. 2) Использовать этот тест def test\_post\_add\_a\_new\_pet\_without\_a\_photo в этом файле test\_pet\_friends.py. для теста этого запроса. 3) Внести изменения в тест. headers = {'auth\_key': auth\_key['key']} data = {} status, result = requests.post("https://petfriends.skillfactory.ru/" + 'api/pets'**,** headers=headers**,** data=data)Убрать атрибуты у теста def test\_post\_add\_a\_new\_pet\_without\_a\_photo() | * + Код ответа 400 (Bad request)   + Тело ответа: HTML-код c сообщением об ошибке утверждения, что status == 200.   + Время выполнения: не дольше 1 с | Негативный тест-кейс по созданию питомца без фото. Запрос без параметров. |
| **PUT/PATCH/DELETE**  **https://petfriends.skillfactory.ru/api/create\_pet\_simple**  **Headers:**  **auth\_key: корректный ключ API**  **FormData:**  name (str) = имя  animal\_type (str) = тип питомца  age (int) = возраст питомца | * + 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>   + Время выполнения: не дольше 1с | Деструктивное тестирование.  Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживаемым типом запроса. |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id = корректный ID питомца существующего (str)  **Formdata (данные, которые хотите обновить):**  name (str) = имя животного animal\_type (str)= тип животного age (int) = возраст питомца. | * + 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: ID питомца (строка)     - name: строка     - pet\_photo: строка пустая     - user\_id: ключ API (строка auth\_key)   + Заголовки ответа:     - Обязательно:       * content-type:application/json charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: число vary: Origin, Cookie   + Время выполнения: не дольше 1с | Базовый позитивный тест. Обновление данных существующего питомца. |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:** auth\_key - (НЕкорректный ключ API):   1. Убрать 1 символ из корректного ключа. 2. Добавить 1 символ к ключу API 3. Ввести в поле кавычки **‘’** 4. Поменять в корректном ключе буквы на цифры (длина должна сохраниться)   **Path params:**  pet\_id = корректный ID питомца существующего (str)  **FormData:** name (str) = имя animal\_type (str) = тип питомца age (int) = возраст питомца | Для тестов 1, 2, 3, 4:   * + Код ответа 403   + Тело ответа: HTML-код c сообщением о неавторизованном пользователе   + Заголовки ответа     - Обязательно:       * content-type: text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: \*       * content-length: число   + Время выполнения: не дольше 1 с | Негативныe тест-кейсы по созданию питомца без фото с некорректным ключом.  /api/pets Get метод для запроса списка питомцев с их ID. |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:** pet\_id =   1. НЕкорректный ID питомца. Взять Id существующего питомца из my\_pets (списка животных) и поменять одну цифру и букву английскую в строке. (длина такая же должна остаться). 2. Добавить к ID корректному питомца 1 символ   **Formdata (данные, которые хотите обновить):** name (str)= имя английскими буквами animal\_type (str): тип животного англ. буквами (Например Dog) age (int) = возраст питомца. | Для тестов 1,2:   * + 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:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Негативный тест. на проверку pet\_id:   1. убрать одну букву или цифру из ID питомца 2. Проверить длину. Добавить 1 символ |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id} **Headers:** auth\_key: Корректный ключ API **Path params:** pet\_id = корректный ID существующего питомца **Formdata (данные, которые хотите обновить): name:** 1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字) **animal\_type:**  1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字) **age:** int | Для тестов 1,2,3:   * + Код ответа 200   + 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: ID питомца из my\_pets     - name: строка     - pet\_photo: строка пустая     - user\_id: ключ API (строка auth\_key)   + Заголовки ответа:     - Обязательно:       * content-type:application/json charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: число vary: Origin, Cookie   + Время выполнения: не дольше 1с. | Негативный тест. на ввод некорректных данных name, animal\_type при age = int. |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:** pet\_id = корректный ID существующего питомца  **FormData:**  **name** (str) = 1)строка > 256 символов, 2) 1 символ 3) 120 символов.  **animal\_type** (str) = 1) строка > 256 символов, 2) 1 символ 3) 120 символов  **age** (int) = возраст питомца (число). 1) > 256 символов (целое число) 2) 1 символ (целое число) 3) 120 символов (целое число) | Для тестов 1,2,3:   * + Код ответа 200   + 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: ID питомца (строка)     - name: строка     - pet\_photo: строка пустая     - user\_id: ключ API (строка auth\_key)   + Заголовки ответа:     - Обязательно:       * content-type:application/json charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: число vary: Origin, Cookie   + Время выполнения: не дольше 1с. | Негативный тест animal\_types, name = строка более 256 символом, с 1 или со 120.  age (int) = число 256, 1, 120 цифр. |
| **PUT**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:** auth\_key: Корректный ключ API **Formdata:** Без параметров. Предусловие. 1)<https://github.com/QAMax0/AutoTestsApiPetfriends> скачать себе проект. 2) Использовать этот тест def test\_put\_update\_pet() в этом файле test\_pet\_friends.py для теста этого запроса. 3) Внести изменения в тест. headers = {'auth\_key': auth\_key['key']} data = {} status, result = requests.put("https://petfriends.skillfactory.ru/" + 'api/pets' + pet\_id**,** headers=headers**,** data=data)Убрать атрибуты у теста def test\_put\_update\_pet() | * + Код ответа 400 (Bad Request)   + Тело ответа: HTML-код c сообщением о AssertionError (ошибки утверждения что код ответа == 200)   + Время выполнения: не дольше 1 с | Негативный тест-кейс по обновлению информации питомца. Запрос без параметров. |
| **POST/PATCH/DELETE**  **https://petfriends.skillfactory.ru/api/pets/pet\_id**  **Headers:**  **auth\_key: корректный ключ API**  **FormData:**  name (str) = имя  animal\_type (str) = тип питомца  age (int) = возраст питомца | * + 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>   + Время выполнения: не дольше 1с | Деструктивное тестирование.  Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживае- мым типом запроса. |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:**  auth\_key: корректный ключ API(str)  **Path params:**  pet\_id = корректный ID питомца существующего (str) | * + Код ответа 200   + В теле ответа HTML код c сообщением о корректном запросе   + Заголовки ответа     - Обязательно:       * content-type:text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Базовый позитивный тест. Удаление существующего питомца из database. |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:** auth\_key: НЕкорректный ключ API(str)   1. Убрать 1 символ из корректного ключа. 2. Добавить 1 символ (\*) к ключу API 3. **Введите в поле 0** 4. Добавить фигурные скобки ключу API слева { и справа }. Тип set получится или строка с фигурными скобками слева и справа. 5. Добавить квадратные скобки ключу API слева [ и справа ]. Тип получится list. Или строка с квадратными скобками слева и справа. 6. ввести > 256 символов английскими буквами и цифрами (int). 7. Ввести 120 символов. **Path params:** pet\_id = корректный ID питомца существующего (str) | Для всех 7-ми тестов:   * + Код ответа 403 (неверный auth\_key)   + В теле ответа HTML код c сообщением о некорректном запросе   + Заголовки ответа     - Обязательно:       * content-type:text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Негативные тесты Удаление существующего питомца из database с некорректными значениями и длиной auth\_key.  pet\_id - корректный ID питомца существующего (str) для всех 7-ми тестов |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:** auth\_key: корректный ключ API(str)  **Path params:** pet\_id = НЕкорректный ID питомца существующего (str)   1. Убрать 1 символ из корректного pet\_id 2. Добавить 1 символ (\*) к корректному pet\_id 3. **Введите в поле 0** 4. Добавить фигурные скобки к pet\_id слева { и справа }. Тип set получится или строка с фигурными скобками слева и справа. 5. Добавить квадратные скобки pet\_id слева [ и справа ]. Тип получится list. Или строка с квадратными скобками слева и справа. 6. ввести > 256 символов английскими буквами и цифрами (int). 7. Ввести 120 символов. | Для всех 7-ми тестов:   * + Код ответа 400   + В теле ответа HTML код c сообщением о некорректном запросе   + Заголовки ответа     - Обязательно:       * content-type:text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Негативные тест-кейсы. Удаление существующего питомца из database с некорректными значениями и длиной pet\_id.  Auth\_key - корректный ключ API для всех 7-ми тестов. |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers: auth\_key**: НЕкорректный ключ API(str) 1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字)  **Path params:**: pet\_id = корректный ID питомца существующего (str) | Для 3-х тестов:  Код ответа 403 (неверный ключ API)   * + В теле ответа HTML код c сообщением о некорректном запросе   + Заголовки ответа     - Обязательно:       * content-type:text/html charset=utf-8       * date: <дата>     - Опционально:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Негативные тест-кейсы. Удаление существующего питомца из database с некорректными значениями auth\_key |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers: auth\_key**: корректный ключ API(str)  **Path params:**: **pet\_id**: 1)строка спецсимволов “♣☺♂” , “”‘~!@#$%^&\*()?>,./\<][ /\*<!–“”, “${code}”;–> 2) строка символов из кириллицы. (уаоаывотопатвыштыв) 3) китайские иероглифы. (繁体字) | Для 3-х тестов:  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:       * access-control-allow-origin: https://petfriends.skillfactory.ru       * content-length: (число) vary: Origin, Cookie   + Время выполнения: не дольше 1с | Негативные тест-кейсы. Удаление существующего питомца из database с некорректными значениями pet\_id |
| **DELETE**  [https://petfriends.skillfactory.ru /api](https://petfriends.skillfactory.ru/api)/pets/{pet\_id}  **Headers:** auth\_key: Корректный ключ API **Formdata:** Без параметров. Предусловие. 1)<https://github.com/QAMax0/AutoTestsApiPetfriends> скачать себе проект. 2) Использовать этот тест def test\_delete\_pet() в этом файле test\_pet\_friends.py. для теста этого запроса. 3) Внести изменения в тест. headers = {'auth\_key': auth\_key['key']} status, result = requests.post("https://petfriends.skillfactory.ru/" + 'api/pets' + pet\_id**,** headers=headers) Убрать атрибуты у теста def test\_delete\_pet() | * + Код ответа 400 (Bad request)   + Тело ответа: HTML-код c сообщением об ошибке утверждения, что status == 200.   + Время выполнения: не дольше 1 с | Негативный тест-кейс по удалению питомца. Запрос без параметров. |
| **PUT/PATCH/POST**  **https://petfriends.skillfactory.ru/api/pets/{pet\_id}**  **Headers:**  auth\_key: корректный ключ API(str)  **Path params:**  pet\_id = корректный ID питомца существующего (str) | * + 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>   + Время выполнения: не дольше 1с | Деструктивное тестирование.  Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживаемым типом запроса. |

11111  **Первый позитивный сценарий “Создание питомца без фото и добавление фото ему”.**

| 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. 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. |

**Второй позитивный сценарий “Обновление информации о питомце конкретного пользователя”.**

| 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 списка моих питомцев — GET /api/pets?filter=my\_pets с корректными данными. Получаем id нужного пользователю питомца. 3. Выполнить обновление информации о питомце— PUT/api/pets/{pet\_id}. |
| Ожидание | В ответе на запрос GET /api/pets?filter=my\_pets приходит питомец с корректной обновлённой информацией из шага 3. |

**Третий позитивный сценарий “Удаление информации о питомце конкретного пользователя”.**

| 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 списка моих питомцев — GET /api/pets?filter=my\_pets с корректными данными. Получаем id нужного пользователю питомца для удаления из списка его питомцев (my\_pets) и Database в целом. 3. Выполнить удаление информации питомца с id из 2 шага — DELETE/api/pets/{pet\_id}. |
| Ожидание | В ответе на запрос GET /api/pets?filter=my\_pets приходит список питомцев пользователя, где нет информации об удалённом питомце. |

**Первый негативный сценарий “Удаление питомца с использованием некорректного ключа API(auth\_key)”**

| 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 списка моих питомцев — GET /api/pets?filter=my\_pets с корректными данными. Получаем id нужного пользователю питомца для удаления из списка его питомцев (my\_pets) и Database в целом. 3. Выполнить удаление информации питомца с id из 2 шага используя некорректный auth\_key (Например - \*) — DELETE/api/pets/{pet\_id}. (**Swagger PetFriends API v1)** |
| Ожидание | В ответе на запрос GET /api/pets?filter=my\_pets приходит список питомцев пользователя, где есть информация о питомце, которого хотели удалить. Питомец не должен был быть удалён при использовании некорректного ключа auth\_key. |

**Второй негативный сценарий “Добавление фото некорректного формата существующему питомцу ”**

| 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 списка моих питомцев — GET /api/pets?filter=my\_pets с корректными данными. Получаем id нужного пользователю питомца для добавления фото питомцу с пустой строкой значения ключа pet\_photo (pet\_photo: ‘’) 3. Выполнить добавление фото питомцу с id из 2 шага используя некорректный тип file(.bmp) для параметра pet\_photo — POST/api/pets/set\_photo/{pet\_id}. (**Swagger PetFriends API v1)** |
| Ожидание | В ответе на запрос GET /api/pets?filter=my\_pets приходит список питомцев пользователя, где есть информация о питомце, у которого в параметре pet\_photo пустая строка. |

**Третий негативный сценарий “Обновление информации**

**о существующем питомце с некорректным auth\_key”**

| 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 списка моих питомцев — GET /api/pets?filter=my\_pets с корректными данными. Получаем id нужного пользователю питомца для обновления информации о нём. Фотографируем данные этого питомца. 3. Выполнить обновление информации о питомце с id из 2 шага используя некорректный ayth\_key (Например - $) — PUT/api/pets/{pet\_id} (**Swagger PetFriends API v1)** |
| Ожидание | В ответе на запрос GET /api/pets?filter=my\_pets приходит список питомцев пользователя, где информация о питомце с id из 2 шага не была обновлена. |