**Тест кейсы для API Дом питомца**

|  |  |  |
| --- | --- | --- |
| **Вызов API** | **Результат выполнения запроса** | **Примечания** |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: зарегистрированный * password: верный | * Код ответа 200 OK * Вход в личный кабинет пользователя * Заголовки ответа содержат:   -content-type: text/json  -date: <дата>   * В теле приходит секретный ключ для дальнейшей работы с API * Время выполнения: не дольше 1с | Позитивный тест с опциональными  параметрами.  Введены корректные данные (email и password), личный кабинет пользователя доступен для дальнейшей работы |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: незарегистрированный * password: любой | * Код ответа 403 Forbidden * Вход в личный кабинет пользователя не выполнен * Заголовки ответа содержат:   -content-type: text/html; charset=utf-8  -date: <дата>   * В теле приходит информация о том, что пользователь с такими данными не найден в базе данных * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Ожидаемый код ответа 401 Unauthorized (запрос не был применен, поскольку ему не хватает действительных учетных данных для целевого ресурса) |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: набор рандомных символов * password: любой | * Код ответа 403 Forbidden * Вход в личный кабинет пользователя не выполнен * Заголовки ответа содержат:   -content-type: text/html; charset=utf-8  -date: <дата>   * В теле приходит информация о том, что пользователь с такими данными не найден в базе данных * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: число * password: любой | * Код ответа 403 Forbidden * Вход в личный кабинет пользователя не выполнен * Заголовки ответа содержат:   - content-type: text/html; charset=utf-8  - date: <дата>   * В теле приходит информация о том, что пользователь с такими данными не найден в базе данных * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: кириллица * password: любой | * Код ответа 403 Forbidden * Вход в личный кабинет пользователя не выполнен * Заголовки ответа содержат:   - content-type: text/html; charset=utf-8  - date: <дата>   * В теле приходит информация о том, что возникла ошибка TypeError * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Не удалось прочитать свойство «заголовки» из «RequestInit»: строка содержит кодовую точку, отличную от ISO-8859-1 |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: пустой ввод * password: любой | * система требует ввод email * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  Email является обязательным полем для ввода |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: зарегистрированный * password: неверный | * Код ответа 403 Forbidden * Вход в личный кабинет пользователя не выполнен * Заголовки ответа содержат:   - content-type: text/html; charset=utf-8  - date: <дата>   * В теле приходит информация о том, что пользователь с такими данными не найден в базе данных * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  Ожидаемый код ответа 401 Unauthorized (запрос не был применен, поскольку ему не хватает действительных учетных данных для целевого ресурса) |
| GET https://petfriends1.herokuapp.com/api/key  Headers:   * email: зарегистрированный * password: пустой | * система требует ввода password * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  Password является обязательным полем для ввода |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате | * Код ответа 200 * В теле приходит HTML 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                             -  user\_id: guid * Заголовки ответа:                      -  content-type: text/html; charset=utf-8                              -   date: <дата>            -  connection: keep-alive                  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie                      -  via: 1.1 vegur * Время выполнения: не дольше 1с | Базовый позитивный тест. При вводе имени на кириллице может возникнуть ошибка (на сайте имя будет отображаться некорректно) |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены цифровые значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате | * Код ответа 200 * В теле приходит HTML 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  -  user\_id: guid   * Заголовки ответа:   -  content-type: text/html; charset=utf8  -   date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест. Тест пройден, но все же вряд ли кто-то назвал бы животное цифрой. Я бы поставила ограничения на ввод только буквенных символов |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены спецсимволы  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате | * Код ответа 200 * В теле приходит HTML 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  -  user\_id: guid   * Заголовки ответа:   -  content-type: text/html; charset=utf-8  -   date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест. Тест пройден, но все же вряд ли кто-то назвал бы животное спецсимволом. Я бы поставила ограничения на ввод только буквенных символов |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: пустая строка  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате | * система требует ввод данных в поле “name” * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  name является обязательным полем для ввода |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены цифровые значения  - age: введены корректные значения в цифровом формате | * Код ответа 200 * В теле приходит HTML 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  -  user\_id: guid   * Заголовки ответа:   -  content-type: text/html; charset=utf-8  -   date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест. Тест пройден, но все же вряд ли существует тип животного с цифровым значением. Я бы поставила ограничения на ввод только буквенных символов.  На сайте информация в карточке питомца может исказиться (вместо имени появятся символы, возраст и тип животного будут не те, что указывал клиент) |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены спецсимволы  - age: введены корректные значения в цифровом формате | * Код ответа 200 * В теле приходит HTML 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  -  user\_id: guid   * Заголовки ответа:   -  content-type: text/html; charset=utf-8  -   date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie                     -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест. Тест пройден, но все же вряд ли существует тип животного со спецсимволом. Я бы поставила ограничения на ввод только буквенных символов.  На сайте информация в карточке питомца может исказиться (вместо имени появятся символы, возраст и тип животного будут не те, что указывал клиент) |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: пустая строка  - age: введены корректные значения в цифровом формате | * система требует ввод данных в поле “animal\_type” * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  animal\_type является обязательным полем для ввода |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены буквенные значения | * система требует ввод цифровых значений * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  age является полем для ввода цифровых значений |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены спецсимволы | * система требует ввод цифровых значений * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом.  age является полем для ввода цифровых значений |
| POST  https://petfriends1.herokuapp.com/api/create\_pet\_simple  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: пустая строка | * система требует ввод цифровых значений * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  age является обязательным полем для ввода |
| POST  https://petfriends1.herokuapp.com/api/pets  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  File:   * корректный файл | * Код ответа 200 * В теле приходит HTML 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  -  user\_id: guid  - pet\_photo (type): image/jpeg   * Заголовки ответа:   -  content-type: text/html;  -  charset=utf-8  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест. При вводе имени на кириллице может возникнуть ошибка (на сайте имя будет отображаться некорректно) |
| POST  https://petfriends1.herokuapp.com/api/pets  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  File:   * текстовый файл | * Код ответа 500 INTERNAL SERVER ERROR * В теле приходит HTML 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  -  user\_id: guid  - pet\_photo (type): application/msword   * Заголовки ответа:   -  content-type: text/html;  -  charset=utf-8  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом. Сервер обнаружил внутреннюю ошибку и не смог выполнить ваш запрос. Либо сервер перегружен, либо ошибка в приложении |
| POST  https://petfriends1.herokuapp.com/api/pets  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  File:   * попытка добавить два фото | * Система не дает возможности добавить в одну карточку два файла * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом |
| POST  https://petfriends1.herokuapp.com/api/pets  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  File:   * пустой ввод | * Система требует добавление файла в графу pet\_photo * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  pet\_photo является обязательным полем для ввода |
| DELETE  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API   Path:   * существующий  ID | * Код ответа 200 * В теле приходит HTML с корректной структурой * В теле приходят URL удаленного объекта * Заголовки ответа:   -  content-type: text/html;  -  charset=utf-8  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест |
| DELETE  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API   Path:   * несуществующий  ID | * Код ответа 200 * В теле приходит HTML с корректной структурой * В теле приходят URL удаленного объекта * Заголовки ответа:   -  content-type: text/html;  -  charset=utf-8  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Негативный тест с некорректным вводом. Удаляется сущность с несуществующим ID |
| DELETE  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API   Path:   * пустой ввод | * Система требует добавление информации в графу pet\_id * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  pet\_id  является обязательным полем для ввода |
| PUT  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  Path:   * существующий  ID | * Код ответа 200 * В теле приходит json с корректной структурой * В теле приходят URL измененного объекта * Заголовки ответа:   -  content-type: application/json  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест |
| PUT  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API * formData:   - name: отсутствует  - animal\_type: введены корректные значения  - age: отсутствует  Path:   * существующий  ID | * Код ответа 200 * В теле приходит json с корректной структурой * В теле приходят URL измененного объекта * Заголовки ответа:   -  content-type: application/json  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест |
| PUT  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API * formData:   - name: введены корректные значения  - animal\_type: введены корректные значения  - age: введены корректные значения в цифровом формате  Path:   * несуществующий  ID | * Код ответа 400 BAD REQUEST * В теле приходит HTML с корректной структурой * В теле приходят URL несуществующего объекта, который пытались изменить * Заголовки ответа:   -  content-type: text/html  -  charset=utf-8  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Негативный тест. Питомец с таким идентификатором не найден |
| PUT  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API * formData:   - name: отсутствует  - animal\_type: отсутствует  - age: отсутствует  Path:   * существующий  ID | * Код ответа 200 * В теле приходит json с корректной структурой * В теле приходят URL объекта, который хотели изменить * Заголовки ответа:   -  content-type: application/json  -  date: <дата>  -  connection: keep-alive  -  server: gunicorn/20.0.4  -  vary: Origin, Cookie  -  via: 1.1 vegur   * Время выполнения: не дольше 1с | Базовый позитивный тест |
| PUT  https://petfriends1.herokuapp.com/api/pets/{pet\_id}  Headers:   * auth\_key: корректный ключ API * formData:   - name: отсутствует  - animal\_type: отсутствует  - age: отсутствует  Path:   * отсутствует | * Система требует ввод данных в графу pet\_id * Время выполнения: не дольше 1с | Негативный тест с пустым вводом.  pet\_id является обязательным полем для ввода |

**Приемочное тестирование API (пользовательские сценарии)**

|  |  |
| --- | --- |
| **Предусловие** | В системе создан пользователь с ником admin@mail.ru и паролем admin1 |
| **1.1 Шаги для повторения (взято из модуля как пример)** | 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} |
| **1.2 Ожидание** | В ответе на запрос GET /api/pets приходит созданный питомец с корректно загруженным файлом из шага 3 и данными питомца |
| **2.1 Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя 2. Выполнить запрос GET/api/pets с корректными данными пользователя. Получаем информацию о созданных объектах, в частности ID каждого питомца 3. Выполнить запрос на удаление питомца — DELETE /api/pets/{pet\_id} с корректными данными и нужным нам ID 4. Получаем ссылку на удаленный объект |
| **2.2 Ожидание** | В ответе на запрос GET /api/pets приходят данные о загруженных питомцах, исключая удаленного |
| **3.1 Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя 2. Выполнить запрос GET/api/pets с корректными данными пользователя. Получаем информацию о созданных объектах, в частности ID каждого питомца 3. Выполнить запрос на изменение данных питомца — PUT /api/pets/{pet\_id} с корректными данными и нужным нам ID. Изменяем информаци. (имя, возраст или тип животного) 4. Получаем ссылку на измененный объект |
| **3.2 Ожидание** | В ответе на запрос GET /api/pets приходят измененные данные в карточке питомца, с которой мы работали |
| **4.1 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} |
| **4.2 Ожидание** | В ответе на запрос GET /api/pets приходит созданный питомец с корректно загруженным файлом из шага 3 и данными питомца |
| **5.1 Шаги для повторения** | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя 2. Выполнить запрос GET/api/pets с корректными данными пользователя. Получаем информацию о созданных объектах, в частности ID каждого питомца |
| **5.2 Ожидание** | В ответе на запрос GET /api/pets приходит список всех созданных питомцев |