1. **Тест-кейсы для API Дом питомца https://petfriends1.herokuapp.com/apidocs/**

|  |  |
| --- | --- |
| Вызов API | Результат выполнения |
| **GET**/api/pets | |
| **GET** /api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Query:  -filter: пустая строка | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| **GET** /api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Query:  -filter: my\_pets | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| **GET** /api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Query:  -filter: строка 255 символов | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| GET /api/pets  Headers:  -auth\_key: 5  Query Params:  -filter=my\_pets | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован  4.Время выполнения: Не дольше 1сек |
| GET /api/pets  Headers:  -auth\_key: abc  Query Params:  -filter=my\_pets | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован  4. Время выполнения: Не дольше 1сек |
| GET /api/pets  Headers:  -auth\_key: корректный ключ api  Query Params:  filter=pets | 1.Код ответа: 400 (Bad request)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о не верном значении параметра filter  4. Время выполнения: Не дольше 1сек |
| PUT/PATCH/DELETE  /api/pets  Headers:  -auth\_key: корректный ключ api  Query Params:  - filter: my\_pets | 1.Код ответа: 405 (Method not allowed)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом HTTP-методе.  4. Время выполнения: Не дольше 1сек |
| GET /api/pets  Headers:  - auth\_key: корректный ключ api  - Content-Type: application/xml; application/x-www-form-urlencoded; application/from-data; text/plain  Query Params:  - filter=pets | 1.Код ответа: 415 (Unsupported content\_type)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом формате данных.  4. Время выполнения: Не дольше 1сек |
| GET /api/pets  Headers:  - auth\_key: корректный ключ api  Query Params:  - filter: строка > 255 символов  - filter: строка спецсимволов  - filter: строка символов из кириллицы | 1.Код ответа: 400 (Bad request)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о не верном значении параметра filter  4. Время выполнения: Не дольше 1сек |
| GET/api/key | |
| GET /api/key  Headers:  - email: существующий email  - password: существующий password | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; date <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой (api\_key) * 4. Время выполнения: Не дольше 1сек |
| GET /api/key  Headers:  - email: не существующий email  - password: существующий password | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что такого пользователя не найдено в базе   * 4.Время выполнения: Не дольше 1сек |
| GET /api/key  Headers:  - email: существующий email  - password: не существующий password | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что такого пользователя не найдено в базе   * 4.Время выполнения: Не дольше 1сек |
| GET /api/key  Headers:  - email: не существующий email  - password: не существующий password | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что такого пользователя не найдено в базе   * 4.Время выполнения: Не дольше 1сек |
| GET /api/key  Headers:  - email: пустая строка  - password: пустая строка | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что обязательные поля не заполнены   * 4.Время выполнения: Не дольше 1сек |
| PUT/PATCH/DELETE  Headers:  - email: существующий email  - password: существующий password | 1.Код ответа: 405 (Method not allowed)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом HTTP-методе.  4. Время выполнения: Не дольше 1сек |
| GET /api/key  Headers:  - email: кирилическими символами  - password: кирилическими символами, спецсимволами | Swagger выдает следующую ощибку: TypeError: Failed to execute 'fetch' on 'Window': Failed to read the 'headers' property from 'RequestInit': String contains non ISO-8859-1 code point.  (Что будет дальше (код ответа, тело ответа, заголовки) не знаю потому что сваггер выдает ошибку выше) |
| POST /api/pets/set\_photo/{pet\_id} | |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Payload:  -корректный графический файл | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: некорректный ключ (пустая строка, числа, кириллица, спецсимволы)  Path Params:  -pet\_id: корректный id питомца  Payload:  -корректный графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Payload:  -текстовый файл | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о некорректном запросе  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Payload:  -“битый” файл ([**base64**](http://www.8bytes.net/2017/11/25/algoritm-kodirovaniya-i-dekodirovaniya-base64/)) | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что графический файл испорчен.  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Payload:  -пустая строка | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не все обязательные поля заполнены.  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца (несуществующий id, количество символов > 255, спецсимволы, кириллица)  Payload:  -графический файл | 1.Код ответа: 404 (Not found)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что питомец с таким id не найден  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца (пустая строка)  Payload:  -графический файл | 1.Код ответа: 404 (Not found)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не все обязательные поля заполнены  4.Время выполнения: Не дольше 1сек |
| POST /api/pets/set\_photo/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Payload:  -загрузить фотографию повторно | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что фотография уже загружена.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple | |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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: ‘ ’,     - - user\_id: guid * 5. Время выполнения: не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Некорректный ключ API (токен с истекшим сроком действия, набор случайных чисел, кириллица, спецсимволы)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Некорректный ключ API (пустая строка)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: некорректное имя (количество символов > 255, спецсимволы)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: некорректное имя (пустая строка)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя  -animal\_type: некорректная порода (количество символов > 255, спецсимволы)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя  -animal\_type: некорректная порода (пустая строка)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (количество символов > 255, спецсимволы, число > 100) | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/create\_pet\_simple  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (пустая строка) | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets | |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Некорректный ключ API (токен с истекшим сроком действия, набор случайных чисел, кириллица, спецсимволы)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Некорректный ключ API (пустая строка)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: некорректное имя (количество символов > 255, спецсимволы)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: некорректное имя (пустая строка)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя  -animal\_type: некорректная порода (количество символов > 255, спецсимволы)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя  -animal\_type: некорректная порода (пустая строка)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (количество символов > 255, спецсимволы, число > 100)  -pet\_photo: графический файл | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (пустая строка)  -pet\_photo: графический файл | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: “битый” файл (base64) | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что графический файл испорчен.  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: текстовый файл | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о некорректном запросе  4.Время выполнения: Не дольше 1сек |
| POST/api/pets  Parameters:  Headers:  -auth\_key: Корректный ключ API  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100).  -pet\_photo: пустая строка | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не все обязательные поля заполнены.  4.Время выполнения: Не дольше 1сек |
| DELETE/api/pets/{pet\_id} | |
| DELETE/api/pets/{pet\_id}  Parameters  Headers:  -auth\_key: Корректный ключ API  Path\_params:  -pet\_id: Корректный Id питомца | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. Время выполнения: не дольше 1сек |
| DELETE/api/pets/{pet\_id}  Parameters  Headers:  -auth\_key: Некорректный ключ API (токен с истекшим сроком действия, набор случайных чисел, кириллица, спецсимволы, число)  Path\_params:  -pet\_id: Корректный Id питомца | * 1.Код ответа 403(Forbidden) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: text/html; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur   3. Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован.   * 4. Время выполнения: не дольше 1сек |
| DELETE/api/pets/{pet\_id}  Parameters  Headers:  -auth\_key: пустая строка  Path\_params:  -pet\_id: Корректный Id питомца | * 1.Код ответа 403 (Forbidden) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: text/html; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur   3. Тело ответа: HTML-код с сообщением о том, что не все обязательны поля заполнены.   * 4. Время выполнения: не дольше 1сек |
| DELETE/api/pets/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца (несуществующий id, количество символов > 255, спецсимволы, кириллица, число) | 1.Код ответа: 404 (Not found)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что питомец с таким id не найден  4.Время выполнения: Не дольше 1сек |
| DELETE/api/pets/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца(пустая строка) | * 1.Код ответа 403 (Forbidden) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: text/html; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur   3. Тело ответа: HTML-код с сообщением о том, что не все обязательны поля заполнены.  4. Время выполнения: не дольше 1сек |
| PUT /api/pets/{pet\_id} | |
| PUT /api/pets/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | * 1.Код ответа 200 (OK) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: application/json; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur * 3. В теле приходит json с корректной структурой   + - 4. 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 * 5. Время выполнения: не дольше 1сек |
| PUT /api/pets/{pet\_id}  Headers:  -auth\_key: некорректный ключ api API (токен с истекшим сроком действия, набор случайных чисел, кириллица, спецсимволы, число)  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | * 1.Код ответа 403(Forbidden) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: text/html; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur   3. Тело ответа: HTML-код с сообщением о том, что пользователь не авторизован.  4. Время выполнения: не дольше 1сек |
| PUT /api/pets/{pet\_id}  Headers:  -auth\_key: некорректный ключ api API (пустая строка)  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | * 1.Код ответа 403(Forbidden) * 2. Заголовки ответа содержат:   + - - Обязательно: content-type: text/html; charset=utf-8; date: <дата>;     - - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur   3. Тело ответа: HTML-код с сообщением о том, что не все обязательные поля заполнены.  4. Время выполнения: не дольше 1сек |
| PUT /api/pets/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца (несуществующий id, количество символов > 255, спецсимволы, кириллица)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 404 (Not found)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что питомец с таким id не найден  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Headers:  -auth\_key: корректный ключ api  Path Params:  -pet\_id: некорректный id питомца (пустая строка)  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не все обязательные поля заполнены  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: некорректное имя (количество символов > 255, спецсимволы)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: некорректное имя (пустая строка)  -animal\_type: порода (количество символов до 255)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя  -animal\_type: некорректная порода (количество символов > 255, спецсимволы)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя  -animal\_type: некорректная порода (пустая строка)  -age: возраст (корректное число от 0 до 100). | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (количество символов > 255, спецсимволы, число > 100) | 1.Код ответа: 400 (Bad Request)  2.Заголовки  -Обязательно: content-type: text/html; charset=utf-8; date: <дата>  -Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о недопустимом вводе параметра.  4.Время выполнения: Не дольше 1сек |
| PUT /api/pets/{pet\_id}  Parameters:  Headers:  -auth\_key: Корректный ключ API  Path Params:  -pet\_id: корректный id питомца  Body:  -name: корректное имя (количество символов до 255)  -animal\_type: порода (количество символов до 255)  -age: некорректный возраст (пустая строка) | 1.Код ответа: 403 (Forbidden)  2.Заголовки  - Обязательно: content-type: text/html; charset=utf-8; date: <дата>  - Опционально: connection: keep-alive; server: gunicorn/20.0.4; via: 1.1 vegur  3.Тело ответа: HTML-код с сообщением о том, что не заполнены все обязательные поля.  4.Время выполнения: Не дольше 1сек |

1. **Пользовательские сценарии.**

|  |  |
| --- | --- |
| 1. **Получить список своих питомцев.** | |
| **Предусловие** | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| **Шаги для повторения** | 1. Выполнить запрос получения ключа API - GET/api/key с корректными данными пользователя. Получаем ключ пользователя. 2. Выполнить запрос получения списка своих питомцев - GET/api/pets, с фильтром my\_pets. |
| **Ожидаемый результат** | В ответе на запрос GET/api/pets с фильтром my\_pets приходит информация со списком питомцев авторизованного пользователя. |
| 1. **Создать 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 без фотографии - POST/api/create\_pet\_simple с корректными данными. Получаем id созданного питомца. |
| Ожидаемый результат | В ответе на запрос GET/api/pets приходит информация со списком питомцев, в котором присутствует созданный питомец с корректными данными. |
| 1. **Добавить фотографию 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. Получить ключ пользователя. 2. Выполнить запрос для получения списка своих питомцев - GET/api/pets, с фильтром my\_pets. Получаем id питомца, фотографию которого нужно добавить. 3. Выполнить запрос для добавления фотографии питомца - POST/api/pets/set\_photo{pet\_id}, добавить графический файл, который будет фотографией питомца. Файл должен быть в формате JPG, JPEG или PNG. |
| Ожидаемый результат | В ответе на запрос GET/api/pets с фильтром my\_pets приходит информация со списком своих питомцев, в котором есть питомец фотографию которого мы добавили выше. |
| 1. **Обновление информации о питомце** | |
| 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. Получить ключ пользователя. 2. Выполнить запрос для получения списка своих питомцев - GET/api/pets, с фильтром my\_pets. Получаем id питомца, данные которого необходимо изменить. 3. Выполнить запрос для изменения данных питомца - PUT/api/pets/{pet\_id}, и в теле запроса передаем данные, которые нужно изменить. |
| Ожидаемый результат | В ответе на запрос GET/api/pets с фильтром my\_pets приходит информация со списком своих питомцев, в котором есть питомец данные которого мы изменили. |
| 1. **Удаление питомца 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. Получить ключ пользователя.       2. Выполнить запрос для получения списка своих питомцев - GET/api/pets, с фильтром my\_pets. Получаем id питомца которого необходимо удалить.       3. Выполнить запрос для удаления питомца - DELETE/api/pets/{pet\_id}. |
| Ожидаемый результат | В ответе на запрос GET/api/pets с фильтром my\_pets приходит информация со списком своих питомцев, удаленный питомец в списке отсутствует. |
| 1. **Создать 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. Получить ключ пользователя. 2. Выполнить запрос создания питомца - POST/api/pets с корректными данными. Получаем id созданного питомца. 3. Выполнить загрузку фотографии - POST/api/pets/set\_photo/{pet\_id} |
| Ожидаемый результат | В ответе на запрос GET/api/pets приходит созданный питомец с корректно загруженным файлом из шага 3 и данными питомца. |