| Base URL: | https://api.cert.skroy.ru/ |  |
| --- | --- | --- |
| Swagger | https://api.cert.skroy.ru/schema/swagger-ui/ |  |
| Component часть 1 | *GET* запрос /api/component/ |  |
| Вызов API | Результат выполнения запроса | Примечание |
| **GET**  https://api.cert.skroy.ru/api/component/  **Headers:**  page (integer)  (query) необязательный | * Код ответа 200 (OK) * Тело ответа: Содержит список компонентов * Заголовки ответа   + Обязательно:     - content-type: application/json     - date:<дата>   + Опционально:     - connection: keep-alive     - server: nginx/1.18.0 (Ubuntu)     - via: 1.1 vegur * Время выполнения: не дольше 1с | Позитивный тест Негативных тестов с корректным вводом на данном запросе быть не может. |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: Корректный email password: 12345 | * 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 password является string типом, а мы пытаемся передать int. |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: Корректный email password: a | * 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 password не может быть a, это более длинная строка. |
| **PUT/PATCH/DELETE**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: Корректный email password: Корректный password | * 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. |
| **GET**  https://petfriends.skillfactory.ru/api/key  **Headers:**  email: Корректный email password: Корректный password  - Content-Type: application/xml; application/x-www-form- urlencoded; application/form-data; text/plain | * 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 не поддерживаемый тип данных в запрос. |

| Домашняя работа часть 2 | Запросы POST/PUT/DELETE |  |
| --- | --- | --- |
| Вызов API | Результат выполнения запроса | Примечание |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number) | * 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: none   + 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/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: none  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number) | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимости * передачи в запросе ключа * 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, отсутствует auth\_key в запросе |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=64  animal\_type=вид питомца (string)  age=возраст питомца (number) | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном   синтаксисе в 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 некорректным вводом, name = number, хотя должен быть string |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=two | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном   синтаксисе в 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 некорректным вводом, age = string, хотя должен быть number |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: 343  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number) | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимости   передачи в запросе ключа   * 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, auth\_key не может быть int 343, это  должно быть более длинное значение в формате string |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=>256 символов (string)  name=строка спецсимволов  animal\_type=вид питомца (string)  age=возраст питомца (number) | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном   синтаксисе в 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 некорректным вводом name |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number) | * 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. |
| **POST**  [https://petfriends.skillfactory.ru/api/create\_pet\_simple](https://petfriends.skillfactory.ru//api/create_pet_simple)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  - Content-Type: application/xml; application/x-www-form-  urlencoded; application/form-data; text/plain | * 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 не поддерживаемый  тип данных в запрос. |

| Вызов API | Результат выполнения запроса | Примечание |
| --- | --- | --- |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **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: none   + 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](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: none  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимости передачи в запросе ключа * 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, отсутствует auth\_key в запросе |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=123  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом, name = number, хотя должен быть string |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=alone  **Payload**:  корректный файл | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом, age = string, хотя должен быть number |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: 1234  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимоcти передачи в запросе ключа * 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, auth\_key не может быть int 1234, это должно быть более длинное значение в формате string |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=>256 символов (string)  name=строка спецсимволов  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл | * Код ответа 400 (Bad Request) * В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом name |
| **PUT/PATCH/DELETE**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл | * 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. |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  корректный файл  Content-Type: application/xml; application/x-www-form-  urlencoded; application/form-data; text/plain | * 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 не поддерживаемый  тип данных в запрос. |
| **POST**  [https://petfriends.skillfactory.ru/api/pets](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **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](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **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](https://petfriends.skillfactory.ru//api/pets)  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string)  animal\_type=вид питомца (string)  age=возраст питомца (number)  **Payload**:  “битый” файл (в 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 |

| Вызов API | Результат выполнения запроса | Примечание |
| --- | --- | --- |
| **DELETE**  <https://petfriends.skillfactory.ru/api/pets/>{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **Path** **params**:  pet\_id=корректный id питомца | * Код ответа 200 (Created) * Тело ответа приходит пустым * 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 | Базовый позитивный тест |
| **DELETE**  <https://petfriends.skillfactory.ru/api/pets/>{pet\_id}  **Headers**:  auth\_key: Sf132  **Path** **params**:  pet\_id=корректный id питомца | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимости передачи в запросе корректного ключа * 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, в запросе передается не существующий auth\_key |
| **DELETE**  <https://petfriends.skillfactory.ru/api/pets/>{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **Path** **params**:  pet\_id=не корректный id питомца (удалили из  корректного pet\_id последние два символа) | * Код ответа 403 (Forbidden) * В теле ответа HTML код c сообщением о необходимсоти передачи в запросе корректного значения ключа * 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, в запросе передается не существующий  pet\_id |
| **PUT/PATCH/GET**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id=корректный id 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>   + 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. |
| **DELETE**  <https://petfriends.skillfactory.ru/api/pets/>{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **Path** **params**:  pet\_id=корректный id питомца  Content-Type: application/xml; application/x-www-form-  urlencoded; application/form-data; text/plain | * 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. |

| Вызов API | Результат выполнения запроса | Примечание |
| --- | --- | --- |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id 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: uuid   + name: строка   + pet\_photo: none   + 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 |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id питомца  Content-Type: application/xml; application/x-www-form-  urlencoded; application/form-data; text/plain | * 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. |
| **GET/POST/PATCH/DELETE**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **Path** **params**:  pet\_id=корректный id 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>   + 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. |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: none  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id питомца | • Код ответа 403 (Forbidden)  • В теле ответа HTML код c сообщением о необходимости передачи в запросе ключа  • 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, отсутствует auth\_key в запросе |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=767 animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id питомца | • Код ответа 400 (Bad Request)  • В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом, name = number, хотя должен быть string |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=what  **Path** **params**:  pet\_id=корректный id питомца | • Код ответа 400 (Bad Request)  • В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом, age = str, хотя должен быть number |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: 545  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id питомца | • Код ответа 403 (Forbidden)  • В теле ответа HTML код c сообщением о необходимости передачи в запросе ключа  • 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, auth\_key не может быть int 545, это  должно быть более длинное значение в формате str |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name= >256 символов (string)  name=строка спецсимволов  animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=корректный id питомца | • Код ответа 400 (Bad Request)  • В теле ответа HTML код c сообщением о некорректном синтаксисе в 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 некорректным вводом name |
| **PUT**  https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers**:  auth\_key: Корректный ключ API  **FormData**:  name=имя питомца (string) animal\_type=вид питомца (string) age=возраст питомца (number)  **Path** **params**:  pet\_id=не корректный id питомца (удалили из корректного pet\_id последние два символа) | • Код ответа 4403 (Forbidden)  • В теле ответа HTML код c сообщением о необходимости передачи в запросе корректного значения  • 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, в запросе передается несуществующий pet\_id |

Пользовательские сценарии

1 сценарий

| Предусловие | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| --- | --- |
| Шаги для повторения | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. |
|  | 2. Выполнить запрос создания питомца — POST /api/pets с корректными данными. Получаем id созданного пользователем питомца. |
|  | 3. Выполнить загрузку фотографии — POST /api/pets/set\_photo/{pet\_id}. |
|  |  |
| Ожидание | В ответе на запрос GET /api/pets приходит созданный питомец с корректно загруженным файлом из шага 3 и данными питомца. |
| 2 сценарий |  |
| Предусловие | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| Шаги для повторения | 1. Выполнить запрос GET /api/key с некорректным ником пользователя. Получаем код ответа 400 |
|  |  |
|  |  |
|  |  |
| Ожидание | В ответе на запрос GET /api/key приходит html сообщение: "Этот пользователь не найден в базе данных" |
| 3 сценарий |  |
| Предусловие | В системе создан пользователь с ником test\_user и паролем P@ssw0rd |
| Шаги для повторения | 1. Выполнить запрос GET /api/key с некорректным паролем пользователя. Получаем код ответа 400 |
|  |  |
|  |  |
|  |  |
| Ожидание | В ответе на запрос GET /api/key приходит html сообщение: "Этот пользователь не найден в базе данных" |
| 4 сценарий |  |
| Предусловие | В системе создан пользователь с ником test\_user и паролем P@ssw0rd и питомец с id=id |
| Шаги для повторения | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. |
|  | 2. Выполнить запрос всех питомцев — GET/api/pets. Получаем список всех питомцев и их id. |
|  |  |
|  | 3. Выполнить запрос изменения питомца — PUT /api/pets/{pet\_id} с корректными данными и некорректным id. Получаем код ответа 404 (Not Found) |
| Ожидание | В ответе на запрос PUT/api/pets/{pet\_id} приходит html сообщение:" Питомец с таким id не найден" |
| 5 сценарий |  |
| Предусловие | В системе создан пользователь с ником test\_user и паролем P@ssw0rd и питомец с id=id |
| Шаги для повторения | 1. Выполнить запрос GET /api/key с корректными данными пользователя. Получаем ключ для пользователя. |
|  | 2. Выполнить запрос всех питомцев — GET/api/pets. Получаем список всех питомцев и их id. |
|  |  |
|  | 3. Выполнить запрос изменения питомца — PUT /api/pets/{pet\_id} с некорректными данными и корректным id. Получаем код ответа 400 (Bad Request) |
| Ожидание | В ответе на запрос PUT/api/pets/{pet\_id} приходит питомец с корректно загруженным файлом и прежними данными питомца. |