Чек-лист API Petfriends <https://petfriends.skillfactory.ru/apidocs/>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Вызов API | Результат выполнения запроса | Комментарий |
| GET api/pets | | | |
| **1** | **GET**  <https://petfriends.skillfactory.ru/api/pets>  **Headers:**  *auth\_key*: Корректный ключ API | 1) Код ответа 200 (OK)  2) Заголовки ответа содержат:  - Обязательно:   * content-type: text/json; charset=utf-8; * date: <дата>;   - Опционально:   * connection: keep-alive; * server: gunicorn/20.0.4; * via: 1.1 vegur   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 1с | Позитивный тест. Проверяем, все ли работает, если мы передаем все параметры запроса корректно |
| **2** | **GET**  <https://petfriends.skillfactory.ru/api/pets?filter=my_pets>  **Headers:**  *auth\_key:* Корректный ключ API  **Query params:**  filter=my\_pets | 1) Код ответа 200 (OK) 2) Заголовки ответа содержат:  - Обязательно:   * content-type: text/json; charset=utf-8; * date: <дата>;   - Опционально:   * connection: keep-alive; * server: gunicorn/20.0.4; * via: 1.1 vegur   3) Содержит список питомцев, принадлежащих пользователю-владельцу ключа API  4) Время выполнения: не дольше 1с | Позитивный тест. Проверяем, работает ли корректный запрос с опциональным параметром filter=my\_pets (значение параметра система предусматривает) |
| 3 | GET  https://petfriends.skillfactory.ru/api/pets?filter=my\_pets  Headers:  auth\_key: Корректный ключ API  Query params:  filter=my\_pets | 1) Код ответа 200 (OK)  2) Заголовки ответа содержат:  - Обязательно:   * content-type: text/json; * charset=utf-8; date: <дата>;   - Опционально:   * connection: keep-alive; * server: gunicorn/20.0.4; * via: 1.1 vegur   3) Содержит список питомцев, принадлежащих пользователю-владельцу ключа API  4) Время выполнения: не дольше 1с | Позитивный тест. Проверяем, работает ли корректный запрос с опциональным параметром filter=my\_pets (значение параметра система предусматривает) |
| 4 | GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: 5  Query params:  filter=my\_pets | 1) Код ответа 401 (Unauthorized)  2) Тело ответа: HTML-код c сообщением о неавторизованном пользователе  3) 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   4) Время выполнения: не дольше 1с | Негативный тест. Проверяем некорректный ввод - в заголовке auth\_key передаем int, хотя нужно передавать string |
| 5 | GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: abc  Query params:  filter=my\_pets | 1) Код ответа 401 (Unauthorized)  2) Тело ответа: HTML-код c сообщением о неавторизованном пользователе  3) 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   4) Время выполнения: не дольше 1с | Негативный тест. Проверяем некорректный ввод - в заголовке auth\_key передаем некорректное значение abc |
| 6 | GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=pets | 1) Код ответа 400 (Bad Request)  2) Тело ответа: HTML код c сообщением о неверном значении параметра filter  3) 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   4) Время выполнения: не дольше 1с | Негативный тест. Проверяем некорректное значение опционального параметра filter - указываем =pets - такого система не предусматривает |
| 7 | PUT/DELETE  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=my\_pets | 1) Код ответа 405 (Method not allowed)  2) Тело ответа: HTML-код c сообщением о недопустимом HTTP-методе  3) 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   4) Время выполнения: не дольше 1с | Деструктивное тестирование. Вызываем известный эндпоинт /api/pets с неподдерживаемым типом запроса put/delete |
| 8 | GET  https://petfriends.skillfactory.ru/api/pets  Headers:  - auth\_key: корректный ключ API  - Content-Type: application/xml; application/x-www-form-urlencoded; application/form-data; text/plain  Query params:  filter=my\_pets | 1) Код ответа 415 (Unsupported content-type)  2) Тело ответа: HTML код c сообщением о недопустимом формате данных  3) 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   4) Время выполнения: не дольше 1с | Деструктивное тестирование. Пытаемся передать неподдерживаемый  тип данных в запрос |
| 9 | GET  https://petfriends.skillfactory.ru/api/pets  Headers:  auth\_key: корректный ключ API  Query params:  filter=строка > 256 символов  filter=строка спецсимволов  filter=строка символов из кириллицы | 1) Код ответа 400 (Bad Request)  2) Тело ответа: HTML-код c сообщением о неверном значении параметра filter.  3) 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   4) Время выполнения: не дольше 1с | Деструктивное тестирование. Пытаемся передать символов больше, чем вмещает string, спецсимволы и кириллицу |
| GET api/key | | | |
| 10 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный, зарегистрированный email  password: корректный пароль | 1) Код ответа 200 ОК  2) Тело ответа: json с корректной структурой  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Позитивный тест. Проверяем, как отрабатывает сервер при передаче корректных email и пароля (существующий пользователь) |
| 11 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный, незарегистрированный email  password: корректный пароль | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что такой пользователь незарегистрирован  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Негативный тест. Проверяем доступ с корректным, но незарегистрированным email |
| 12 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: некорректный email  password: корректный пароль | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что формат email некорректный  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем доступ с некорректным email |
| 13 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный email  password: некорректный пароль | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что пароль некорректный  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем доступ с некорректным паролем |
| 14 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: пустая строка  password: корректный пароль | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что строку email необходимо заполнить  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем доступ с пустой строкой в заголовке email |
| 15 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный email  password: пустая строка | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что строку пароля необходимо заполнить  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем доступ с пустой строкой в заголовке пароль |
| 16 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный email  password: пароль короче минимально разрешенной длины | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что пароль слишком короткий  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем, как сервер обрабатывает пароль короче минимально разрешенной длины |
| 17 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный email  password: пароль длиннее максимально разрешенной длины | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что пароль слишком длинный  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем, как сервер обрабатывает пароль длиннее максимально разрешенной длины |
| 18 | GET  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный email  password: корректный пароль + пробел | 1) Код ответа 403 Forbidden  2) Тело ответа: HTML-код c сообщением о том, что пароль некорректный  3) Заголовки ответа  - Обязательно:   * content-type: application/json * date: <дата>   4) Время выполнения: не дольше 1с | Проверяем ввод некорректных символов. На данный момент **баг** – система пропускает пароль с пробелом |
| 19 | POST/PUT/DELETE  https://petfriends.skillfactory.ru/api/key  Headers:  email: корректный, зарегистрированный email  password: корректный пароль | 1) Код ответа 405 (Method not allowed)  2) Тело ответа: HTML-код c сообщением о недопустимом HTTP-методе  3) Заголовки ответа  - Обязательно:   * content-type: text/html charset=utf-8 * date: <дата>   4) Время выполнения: не дольше 1с | Вызываем известный эндпоинт /api/key с неподдерживаемым типом запроса post/put/delete |
| POST api/pets/set\_photo/{pet\_id} | | | | |
| 20 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **Payload:**  корректный файл | * 1) Код ответа 200 (Created) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) 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 * 5) Время выполнения: не дольше 1с | Позитивный тест. Проверяем, все ли работает, если мы передаем все параметры запроса корректно |
| 21 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **Payload:**  Текстовый файл | * 1) Код ответа 400 (Bad request) * 2) В теле ответа приходит HTML код c сообщением о некорректном запросе * 3) 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 * 4) Время выполнения: не дольше 1с | Негативный тест с корректным вводом. Прикрепляем текстовый файл |
| 22 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: некорректный id питомца  **Payload:**  корректный файл | * 1) Код ответа 404 (Not found) * 2) В теле ответа приходит HTML код c сообщением о том, что питомец с таким id не найден * 3) 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 * 4) Время выполнения: не дольше 1с | Негативный тест. Передаем некорректный id питомца |
| 23 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **Payload:**  Загрузить фотографию второй раз | * 1) Код ответа 400 (Bad request) (представим, что в данной системе можно загрузить только одно фото) * 2) В теле ответа приходит HTML код c сообщением о том, что фото уже загружено * 3) 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 * 4) Время выполнения: не дольше 1с | Негативный тест. Пробуем повторно загрузить фото |
| 24 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **Payload:**  отсутствует | * 1) Код ответа 400 (Bad request) * 2) В теле ответа приходит HTML код c сообщением о том, что фото отсуствует * 3) 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 * 4) Время выполнения: не дольше 1с | Негативный тест. Не прикладываем файл |
| 25 | **POST** https://petfriends.skillfactory.ru/api/pets/set\_photo/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **Payload:**  “битый” файл (в [base64](http://www.8bytes.net/2017/11/25/algoritm-kodirovaniya-i-dekodirovaniya-base64/) представлении изменить несколько символов) | * 1) Код ответа 400 (Bad request) * 2) В теле ответа приходит HTML код c сообщением об испорченном файле * 3) 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 * 4) Время выполнения: не дольше 1с | Деструктивное тестирование. Пробуем передать битый файл |
| POST api/create\_pet\_simple | | | |
| 26 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на латинице * animal\_type: порода на латинице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive   5) Время выполнения: не дольше 1с | Проверяем ввод данных на латинице |
| 27 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive   5) Время выполнения: не дольше 1с | Проверяем ввод данных на кириллице |
| 28 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: пустая строка * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive   5) Время выполнения: не дольше 1с | Проверяем ввод с пустой строкой вместо имени.  На данный момент **баг** – система пропускает пустую строку |
| 29 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: пустая строка * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод с пустой строкой вместо породы.  На данный момент **баг** – система пропускает пустую строку |
| 30 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: пустая строка | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод с пустой строкой вместо возраста.  На данный момент **баг** – система пропускает пустую строку |
| 31 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: символ нижнего подчеркивания \_ * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – спецсимволов в строке имени.  На данный момент **баг** – система пропускает строку, состоящую из спецсимвола |
| 32 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: восклицательный знак * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – спецсимволов в строке породы.  На данный момент **баг** – система пропускает строку, состоящую из спецсимвола |
| 33 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: вопросительный знак | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – спецсимволов в строке возраста.  На данный момент **баг** – система пропускает строку, состоящую из спецсимвола |
| 34 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: отрицательное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – отрицательного числа в строке возраста.  На данный момент **баг** – система пропускает отрицательное число |
| 35 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: пять | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – вводим значение строки вместо числа в строке возраста.  На данный момент **баг** – система пропускает строку, где ожидается только int |
| 36 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: число float | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод некорректных данных – вводим float вместо int в строке возраста.  На данный момент **баг** – система пропускает float, где ожидается только int |
| 37 | **POST** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Некорректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Создаем питомца с некорректным ключом API.  На данный момент **баг** – создание питомца с некорректным ключом API возможно |
| 38 | **GET/PUT/DELETE** https://petfriends.skillfactory.ru/api/create\_pet\_simple  **Headers:**  auth\_key: Некорректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | 1) Код ответа 405 (Method not allowed)  2) Тело ответа: HTML-код c сообщением о недопустимом HTTP-методе  3) Заголовки ответа  - Обязательно:   * content-type: text/html charset=utf-8 * date: <дата>   4) Время выполнения: не дольше 1с | Вызываем известный эндпоинт /api/create\_pet\_simple с неподдерживаемым типом запроса get/put/delete |
| POST api/pets | | | |
| 39 | **POST** https://petfriends.skillfactory.ru/api/pets  **Headers:**  auth\_key: Корректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число * pet\_photo: корректный файл | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive   5) Время выполнения: не дольше 1с | Делаем запрос со всеми корректными данными |
| 40 | **POST** https://petfriends.skillfactory.ru/api/pets  **Headers:**  auth\_key: Некорректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число * pet\_photo: файл в формате jpg | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Добавляем информацию о питомце с некорректным ключом API.  На данный момент **баг** – добавление информации о питомце с некорректным ключом API возможно |
| 41 | **POST** https://petfriends.skillfactory.ru/api/pets  **Headers:**  auth\_key: Некорректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число * pet\_photo: файл в формате txt | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Вкладываем текстовый файл вместо изображения.  На данный момент **баг** – система пропускает файл в формате txt |
| 42 | **POST** https://petfriends.skillfactory.ru/api/pets  **Headers:**  auth\_key: Некорректный ключ API  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число * pet\_photo: без фото | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Вводим все данные корректно, но не выбираем фото  На данный момент **баг** – система пропускает запрос без фото |
| PUT api/pets/{pet\_id} | | | |
| 43 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем обновление данных питомца, корректно заполнив все поля |
| 44 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: пустая строка * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод, не заполнив необязательную строку – имя |
| 45 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: имя на кириллице * animal\_type: пустая строка * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод, не заполнив необязательную строку – порода |
| 46 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое пустая строка | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Проверяем ввод, не заполнив необязательную строку – возраст |
| 47 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Некорректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) 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 * 4) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 5) Время выполнения: не дольше 1с | Обновляем информацию о питомце с некорректным ключом API.  На данный момент **баг** – обновление информации о питомце с некорректным ключом API возможно |
| 48 | **PUT** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: несуществующий id питомца/пустое значение  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | * 1) Код ответа 400 (Bad request) * 2) В теле ответа приходит HTML код c сообщением о том, что питомец с таким id не найден * 3) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 4) Время выполнения: не дольше 1с | Проверяем обновление питомца с несуществующим id/пустым значением id |
| 49 | **GET/POST** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца  **FormData:**   * name: имя на кириллице * animal\_type: порода на кириллице * age: целое положительное число | 1) Код ответа 405 (Method not allowed)  2) Тело ответа: HTML-код c сообщением о недопустимом HTTP-методе  3) Заголовки ответа  - Обязательно:   * content-type: text/html charset=utf-8 * date: <дата>   4) Время выполнения: не дольше 1с | Вызываем известный эндпоинт /api/pets/{pet\_id} с неподдерживаемым типом запроса get/post |
| Delete api/pets/{pet\_id} | | | |
| 50 | **DELETE** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: корректный id питомца | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive   4) Время выполнения: не дольше 1с | Удаляем питомца, введя все корректные данные |
| 51 | **DELETE** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Некорректный ключ API  **Path params:**  pet\_id: корректный id питомца | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 4) Время выполнения: не дольше 1с | Удаляем информацию о питомце с некорректным ключом API.  На данный момент **баг** – удаление информации о питомце с некорректным ключом API возможно |
| 52 | **DELETE** https://petfriends.skillfactory.ru/api/pets/{pet\_id}  **Headers:**  auth\_key: Корректный ключ API  **Path params:**  pet\_id: несуществующий id питомца/пустое значение id | * 1) Код ответа 200 (OK) * 2) В теле приходит json с корректной структурой * 3) Заголовки ответа: * Обязательно:   + content-type:application/json charset=utf-8   + date: <дата> * Опционально:   + connection: keep-alive * 4) Время выполнения: не дольше 1с | Проверяем удаление питомца с несуществующим id/пустым значением id  На данный момент **баг** – удаление информации о питомце с несуществующим id/пустым значением id возвращает статус 200 (ОК) |