

Сводный документ: тест-кейсы для API 'Дом Питомца'

(<https://petfriends.skillfactory.ru/>)

Составитель: Малинин Сергей Александрович

POST /api/create_pet_simple

Вызов API	Результат выполнения запроса	Примечание
POST https://petfriends.skillfactory.ru/api/create_pet_simple Headers: auth_key: <API ключ> Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "4" }	Код ответа 200 (OK) Тело ответа: { "id": "<id питомца>", "name": "Маффин", "animal_type": "Лабрадор", "age": "4" } Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1с	Позитивный тест с корректными данными.
POST https://petfriends.skillfactory.ru/api/create_pet_simple Headers: auth_key: (empty) Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "4" }	Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о неавторизованном доступе 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 без авторизационного ключа.
POST https://petfriends.skillfactory.ru/api/create_pet_simple Headers: auth_key: <API ключ> Body: { "animal_type": "Лабрадор", "age": "4" }	Код ответа 400 (Bad Request) Тело ответа: {"error": "Поле 'name' обязательно"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с отсутствующим обязательным полем 'name'.

	Время выполнения: не дольше 1с	
POST https://petfriends.skillfactory.ru/api/create_pet_simple Headers: auth_key: <API ключ> Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "-4" }	Код ответа 400 (Bad Request) Тело ответа: {"error": "Возраст не может быть отрицательным"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с некорректным значением 'age' (отрицательное значение).
POST https://petfriends.skillfactory.ru/api/create_pet_simple Headers: auth_key: <API ключ> Body: (empty)	Время выполнения: не дольше 1с Код ответа 400 (Bad Request) Тело ответа: {"error": "Тело запроса не может быть пустым"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с пустым телом запроса.
POST /api/create_pet_simple — граничные значения для бизнес-логики	Время выполнения: не дольше 1с Код ответа 200 (ОК) для корректных граничных значений, код ответа 400 или 422 для некорректных значений.	Проверка граничных значений для логики приложения, таких как максимальный возраст или длина имени питомца.
POST /api/create_pet_simple — граничные значения для типов данных	Код ответа 400 (Bad Request) для некорректных типов данных, код ответа 200 (ОК) для корректных типов.	Проверка обработки значений типов данных на границах допустимого диапазона, таких как максимальная длина строки.
POST /api/create_pet_simple — проверка всех составляющих запроса	Тест включает проверку заголовков, тела запроса, кода ответа и времени выполнения (не дольше 1 секунды).	Полная проверка корректности всех элементов запроса.
POST /api/create_pet_simple — Проверка граничных значений для бизнес-логики Пример: Добавление питомца с максимальной длиной имени (255 символов) POST https://petfriends.skillfactory.ru/api/pets Headers:	Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Тело ответа: { "id": "<pet_id>", "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA	Проверка на успешное добавление питомца с максимальной длиной имени.

GET /api/key

Вызов API	Результат выполнения запроса	Примечание
GET https://petfriends.skillfactory.ru/api/key Headers: email: stron@gmail.com password: qwertyuiop	Код ответа 200 (OK) Тело ответа: {"key": "<API ключ>"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не больше 1с	Позитивный тест с корректным логином и паролем.
GET https://petfriends.skillfactory.ru/api/key Headers: email: invalid_user password: qwertyuiop	Код ответа 403 (Forbidden) Тело ответа: HTML-код с 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 больше 1с	Негативный тест с некорректным логином. Логин должен быть корректным email, а введен неизвестный пользователь.
GET https://petfriends.skillfactory.ru/api/key Headers: email: stron@gmail.com password: invalid_password	Код ответа 403 (Forbidden) Тело ответа: HTML-код с 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 больше 1с	Негативный тест с некорректным паролем. Пароль должен быть корректным, но введен неверный пароль.
GET https://petfriends.skillfactory.ru/api/key Headers: (empty)	Код ответа 400 (Bad Request) Тело ответа: HTML-код с 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	Негативный тест без обязательных заголовков. Для авторизации обязательны email и password.

[illegible]

<p>AAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAA AA", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</p>		
<p>GET /api/key — Проверка граничных значений для типов данных Пример: Добавление питомца с некорректным возрастом (отрицательное значение) POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "-1", "pet_photo": "images/001.jpg" }</p>	<p>Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."</p>	<p>Проверка обработки некорректных типов данных.</p>
<p>GET /api/key — Проверка всех составляющих запроса Пример: Проверка заголовков, тела запроса, кода ответа и времени выполнения. GET https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets</p>	<p>Код ответа 200 (OK). Тело ответа: { "pets": [...] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.</p>	<p>Полная проверка корректности всех элементов запроса.</p>

GET /api/pets

Вызов API	Результат выполнения запроса	Примечание
POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: <pre>{ "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</pre>	Код ответа 200 (OK) Тело ответа: { "id": "<pet_id>", "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "<photo_url>" } Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не больше 1с	Позитивный тест с корректными данными и фото.
POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: (empty) Body: <pre>{ "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</pre>	Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о неавторизованном доступе 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 больше 1с	Негативный тест без авторизационного ключа.
POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: <pre>{ "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</pre>	Код ответа 400 (Bad Request) Тело ответа: {"error": "Поле 'name' обязательно"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не больше 1с	Негативный тест с отсутствующим обязательным полем 'name'.
POST https://petfriends.skillfactory.ru/api/pets Headers:	Код ответа 400 (Bad Request) Тело ответа: {"error": "Формат изображения некорректен"}	Негативный тест с некорректным форматом изображения.

GET https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets	Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.	
---	---	--

GET /api/pets

Вызов API	Результат выполнения запроса	Примечание
GEThttps://petfriends.skillfactory.ru/api/pets?filter=my_pets Headers: auth_key: Корректный ключ API Query params: filter=my_pets	Код ответа 200 (OK) Содержит список питомцев, принадлежащих пользователю-владельцу ключа API Заголовки ответа те же, что в тесте #1 Время выполнения: не дольше 1с	Позитивный тест с опциональными параметрами. Так как здесь мы в строке запроса используем параметр filter, который не является обязательным, и устанавливаем его значение =my_pets. Мы всё ещё ожидаем, что запрос корректно выполнится, поэтому сценарий позитивный. Негативных тестов с корректным вводом на данном запросе быть не может. Параметр filter является строкой, а значит любое корректное значение приведет к позитивному исходу запроса с этим параметром. Оставшийся параметр заголовка auth_key обязан быть корректным в этой категории, т.е. негативных ожиданий мы не сможем вызвать.
GEThttps://petfriends.skillfactory.ru/api/pets Headers: auth_key: 5 Query params: filter=my_pets	Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о неавторизованном пользователе Заголовки ответаОбязательно: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 auth_key является string типом, а мы пытаемся передать int.
GEThttps://petfriends.skillfactory.ru/api/pets	Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о	Негативный тест с некорректным вводом.

<p>Headers: auth_key: abc Query params: filter=my_pets</p>	<p>неавторизованном пользователе Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Негативный, потому что мы ожидаем в результатах выполнения ошибку, а с некорректным вводом, потому что auth_key не может быть abc, это более длинная строка.</p>
<p>GEThttps://petfriends.skillfactory.ru/api/pets Headers: auth_key: корректный ключ API Query params: filter=pets</p>	<p>Код ответа 400 (Bad Request) Тело ответа: HTML код с сообщением о неверном значении параметра filter Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Негативный тест с некорректным вводом.</p> <p>Негативный, потому что мы ожидаем в результатах выполнения ошибку, с некорректным вводом, потому что мы передаем в filter значение, непредусмотренное системой.</p>
<p>PUT/PATCH/DELETE https://petfriends.skillfactory.ru/api/pets Headers: auth_key: корректный ключ API Query params: filter=my_pets</p>	<p>Код ответа 405 (Method not allowed) Тело ответа: HTML-код с сообщением о недопустимом HTTP-методе Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Деструктивное тестирование. Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживаемым типом запроса.</p>
<p>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</p> <p>Query params: filter=my_pets</p>	<p>Код ответа 415 (Unsupported content-type) Тело ответа: HTML код с сообщением о недопустимом формате данных Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Здесь через ";" разделены разные значения заголовка Content-Type. То есть этот кейс повторяется 4 раза для каждого значения Content-Type.</p> <p>Этот тест также является разрушительным, так как мы пытаемся передать неподдерживаемый тип данных в запрос.</p>

<pre> AAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA AA", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" } </pre>		
<p>GET /api/pets (Пример) — Проверка граничных значений для типов данных Пример: Добавление питомца с некорректным возрастом (отрицательное значение) POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body:</p> <pre> { "name": "Маффин", "animal_type": "Лабрадор", "age": "-1", "pet_photo": "images/001.jpg" } </pre>	<p>Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."</p>	<p>Проверка обработки некорректных типов данных.</p>
<p>GET /api/pets (Пример) — Проверка всех составляющих запроса Пример: Проверка заголовков, тела запроса, кода ответа и времени выполнения. GET https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets</p>	<p>Код ответа 200 (OK). Тело ответа:</p> <pre> { "pets": [...] } </pre> <p>Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.</p>	<p>Полная проверка корректности всех элементов запроса.</p>

POST /api/pets/set_photo/{pet_id}

Вызов API	Результат выполнения запроса	Примечание
<p>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</p>	<p>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: base64 image user_id: guid</p>	<p>Базовый позитивный тест</p>

	<p>Заголовки ответа:Обязательно: content-type:application/json charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	
<p>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</p>	<p>Код ответа 400 (Bad Request) В теле ответа HTML код с сообщением о некорректном запросе Заголовки ответаОбязательно:content-type:text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	Негативный тест с корректным вводом
<p>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: графический файл</p>	<p>Код ответа 404 (Not Found) В теле ответа HTML код с сообщением о том, что питомец с таким id не найден Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	Негативный тест с корректным вводом
<p>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</p>	<p>Код ответа 400 (Bad Request) (здесь будьте внимательны и читайте спецификацию системы, несколько фото одного питомца могут быть допустимы, для нашего примера мы не даем такой возможности) В теле ответа HTML код с сообщением о том, что фотография уже загружена Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата></p> <p>Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p>	Негативный тест с корректным вводом

	Время выполнения: не дольше 1с	
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	Код ответа 400 (Bad Request) В теле ответа HTML код с сообщением о том, что отсутствует файл Заголовки ответаОбязательно: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
nan	nan	nan
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 представлении изменить несколько символов)	Код ответа 400 (Bad Request) В теле ответа HTML код с сообщением об испорченном файле Заголовки ответаОбязательно: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
POST /api/pets/set_photo/{pet_id} (Пример) — граничные значения для бизнес-логики	Код ответа 200 (ОК) для корректных граничных значений, код ответа 400 или 422 для некорректных значений.	Проверка граничных значений для логики приложения, таких как максимальный возраст или длина имени питомца.
POST /api/pets/set_photo/{pet_id} (Пример) — граничные значения для типов данных	Код ответа 400 (Bad Request) для некорректных типов данных, код ответа 200 (ОК) для корректных типов.	Проверка обработки значений типов данных на границах допустимого диапазона, таких как максимальная длина строки.
POST /api/pets/set_photo/{pet_id} (Пример) — проверка всех составляющих запроса	Тест включает проверку заголовков, тела запроса, кода ответа и времени выполнения (не дольше 1 секунды).	Полная проверка корректности всех элементов запроса.
POST /api/pets/set_photo/{pet_id} (Пример) — Проверка граничных значений для бизнес-логики Пример: Добавление питомца с максимальной длиной имени (255 символов)	Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Тело ответа: { "id": "<pet_id>", "name":	Проверка на успешное добавление питомца с максимальной длиной имени.

filter=my_pets		
----------------	--	--

DELETE /api/pets/{pet_id}

Вызов API	Результат выполнения запроса	Примечание
DELETE https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ>	Код ответа 200 (OK) Тело ответа: {"message": "Питомец удален"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1с	Позитивный тест с корректным идентификатором питомца.
DELETE https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: (empty)	Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о неавторизованном доступе 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 без авторизационного ключа.
DELETE https://petfriends.skillfactory.ru/api/pets/invalid_pet_id Headers: auth_key: <API ключ>	Код ответа 404 (Not Found) Тело ответа: {"error": "Питомец не найден"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1с	Негативный тест с некорректным идентификатором питомца.
DELETE https://petfriends.skillfactory.ru/api/pets/ Headers: auth_key: <API ключ>	Код ответа 400 (Bad Request) Тело ответа: {"error": "Некорректный запрос"} Заголовки ответа: Обязательно: content-type: application/json date: <дата>	Негативный тест с пустым идентификатором питомца.

	<p>Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	
DELETE https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: invalid_key	<p>Код ответа 403 (Forbidden) Тело ответа: HTML-код с сообщением о неавторизованном доступе 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</p> <p>Время выполнения: не дольше 1с</p>	Негативный тест с некорректным авторизационным ключом.
DELETE https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets	<p>Код ответа 200 (OK) Тело ответа: { "message": "Все ваши питомцы удалены" } Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	Позитивный тест: Удаление всех питомцев пользователя с фильтром filter=my_pets.
DELETE https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: (empty)	<p>Код ответа 403 (Forbidden) Тело ответа: {"error": "У вас нет прав на удаление всех питомцев"} Заголовки ответа: Обязательно: content-type: application/json date: <дата> Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	Негативный тест: Попытка удаления всех питомцев без фильтра, что запрещено политикой доступа.
DELETE /api/pets/{pet_id} — граничные значения для бизнес-логики	Код ответа 200 (OK) для корректных граничных значений, код ответа 400 или 422 для некорректных значений.	Проверка граничных значений для логики приложения, таких как

<pre>"name": "Маффин", "animal_type": "Лабрадор", "age": "-1", "pet_photo": "images/001.jpg" }</pre>		
<p>DELETE /api/pets/{pet_id} — Проверка всех составляющих запроса</p> <p>Пример: Проверка заголовков, тела запроса, кода ответа и времени выполнения.</p> <p>GET</p> <p>https://petfriends.skillfactory.ru/api/pets</p> <p>Headers:</p> <p>auth_key: <API ключ></p> <p>Query params:</p> <p>filter=my_pets</p>	<p>Код ответа 200 (OK).</p> <p>Тело ответа:</p> <pre>{ "pets": [...] }</pre> <p>Заголовки ответа:</p> <p>- content-type: application/json</p> <p>- date: <дата></p> <p>Время выполнения запроса: < 1 сек.</p>	<p>Полная проверка корректности всех элементов запроса.</p>

PUT /api/pets/{pet_id}

Вызов API	Результат выполнения запроса	Примечание
<p>PUT</p> <p>https://petfriends.skillfactory.ru/api/pets/{pet_id}</p> <p>Headers:</p> <p>auth_key: <API ключ></p> <p>Body:</p> <pre>{ "name": "Бублик", "animal_type": "Пудель", "age": "3" }</pre>	<p>Код ответа 200 (OK)</p> <p>Тело ответа:</p> <pre>{ "id": "<pet_id>", "name": "Бублик", "animal_type": "Пудель", "age": "3" }</pre> <p>Заголовки ответа:</p> <p>Обязательно:</p> <p>content-type: application/json</p> <p>date: <дата></p> <p>Опционально:</p> <p>connection: keep-alive</p> <p>server: gunicorn/20.0.4</p> <p>via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Позитивный тест с корректными данными.</p>
<p>PUT</p> <p>https://petfriends.skillfactory.ru/api/pets/{pet_id}</p> <p>Headers:</p> <p>auth_key: (empty)</p> <p>Body:</p> <pre>{ "name": "Бублик", "animal_type": "Пудель", "age": "3" }</pre>	<p>Код ответа 403 (Forbidden)</p> <p>Тело ответа: HTML-код с сообщением о неавторизованном доступе</p> <p>Заголовки ответа:</p> <p>Обязательно:</p> <p>content-type: text/html; charset=utf-8</p> <p>date: <дата></p> <p>Опционально:</p> <p>connection: keep-alive</p> <p>server: gunicorn/20.0.4</p> <p>via: 1.1 vegur</p> <p>Время выполнения: не дольше 1с</p>	<p>Негативный тест без авторизационного ключа.</p>
<p>PUT</p>	<p>Код ответа 404 (Not Found)</p>	<p>Негативный тест с</p>

https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "age": "-1", "animal_type": "Лабрадор", "name": "Маффин", "pet_photo": "images/001.jpg" } 		
PUT /api/pets/{pet_id} — Проверка всех составляющих запроса Пример: Проверка заголовков, тела запроса, кода ответа и времени выполнения. GET https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets 	Код ответа 200 (OK). Тело ответа: { "pets": [...] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек. 	Полная проверка корректности всех элементов запроса.

Business Logic Tests

Вызов API	Результат выполнения запроса	Примечание
1. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "age": "4", "animal_type": "Лабрадор", "name": "Маффин", "pet_photo": "images/001.jpg" } 2. PUT https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ> Body: { "age": "3", "animal_type": "Пудель", "name": "Бублик", } 3. DELETE https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: 	1. Код ответа 200 (OK). Питомец создан. 2. Код ответа 200 (OK). Данные питомца обновлены. 3. Код ответа 200 (OK). Питомец удален. 	Проверка полной логики: создание, обновление и удаление питомца.

<p>auth_key: <API ключ></p> <p>1. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</p> <p>2. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "name": "Бублик", "animal_type": "Пудель", "age": "3", "pet_photo": "images/002.jpg" }</p> <p>3. GET https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Query params: filter=my_pets</p>	<p>1. Код ответа 200 (OK). Первый питомец добавлен. 2. Код ответа 200 (OK). Второй питомец добавлен. 3. Код ответа 200 (OK). Возвращен список питомцев, содержащий добавленных питомцев.</p>	<p>Добавление нескольких питомцев и получение списка всех питомцев пользователя.</p>
<p>1. DELETE https://petfriends.skillfactory.ru/api/pets/invalid_pet_id Headers: auth_key: <API ключ></p>	<p>1. Код ответа 404 (Not Found). Ошибка: Питомец не найден.</p>	<p>Удаление несуществующего питомца, проверка корректной обработки ошибки.</p>
<p>1. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body: { "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</p> <p>2. PUT https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ> Body:</p>	<p>1. Код ответа 200 (OK). Питомец создан. 2. Код ответа 400 (Bad Request). Ошибка: Некорректные данные.</p>	<p>Проверка обработки некорректных данных при обновлении питомца.</p>

<pre>{ "name": "", "animal_type": "", "age": "invalid_age" }</pre>		
<p>1. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body:</p> <pre>{ "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</pre> <p>2. GET https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ></p>	<p>1. Код ответа 200 (ОК). Питомец создан. 2. Код ответа 200 (ОК). Данные питомца совпадают с вводом.</p>	<p>Проверка логики получения данных питомца после его создания.</p>
<p>1. POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <API ключ> Body:</p> <pre>{ "name": "Маффин", "animal_type": "Лабрадор", "age": "4", "pet_photo": "images/001.jpg" }</pre> <p>2. DELETE https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ></p> <p>3. GET https://petfriends.skillfactory.ru/api/pets/{pet_id} Headers: auth_key: <API ключ></p>	<p>1. Код ответа 200 (ОК). Питомец создан. 2. Код ответа 200 (ОК). Питомец удален. 3. Код ответа 404 (Not Found). Ошибка: Питомец не найден.</p>	<p>Удаление питомца и проверка его недоступности в системе.</p>