Сводный документ: тест-кейсы для API 'Дом Питомца' (https://petfriends.skillfactory.ru/)

Составитель: Малинин Сергей Александрович

POST /api/create_pet_simple

Вызов АРІ	Результат выполнения запроса	Примечание
POST	Код ответа 200 (ОК)	Позитивный тест с
https://petfriends.skillfactory.ru/api/create	Тело ответа: {	корректными данными.
_pet_simple	"id": " <id питомца="">",</id>	
Headers:	"name": "Маффин",	
auth_key: <api ключ=""></api>	"animal_type": "Лабрадор",	
Body:	"age": "4"	
\	}	
"name": "Маффин",	Заголовки ответа:	
"animal_type": "Лабрадор",	Обязательно:	
"age": "4"	content-type: application/json	
}	date: <дата>	
	Опционально:	
	connection: keep-alive	
	server: gunicorn/20.0.4	
	via: 1.1 vegur	
	Время выполнения: не дольше 1с	
POST	Код ответа 403 (Forbidden)	Негативный тест без
https://petfriends.skillfactory.ru/api/create	Тело ответа: HTML-код с сообщением о	авторизационного ключа.
_pet_simple	неавторизованном доступе	
Headers:	Заголовки ответа:	
auth_key: (empty)	Обязательно:	
Body:	content-type: text/html; charset=utf-8	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	date: <дата>	
"name": "Маффин",	Опционально:	
"animal_type": "Лабрадор", "age": "4"	connection: keep-alive	
_	server: gunicorn/20.0.4	
}	via: 1.1 vegur	
	Время выполнения: не дольше 1с	
POST	Код ответа 400 (Bad Request)	Негативный тест с
https://petfriends.skillfactory.ru/api/create	Тело ответа: {"error": "Поле 'name'	отсутствующим
pet simple	обязательно"}	обязательным полем 'name'.
Headers:	Заголовки ответа:	Total Control of the
auth key: <api ключ=""></api>	Обязательно:	
Body:	content-type: application/json	
 {	date: <дата>	
"animal type": "Лабрадор",	Опционально:	
"age": "4"	connection: keep-alive	
}	server: gunicorn/20.0.4	
	via: 1.1 vegur	

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

auth_key: <api ключ=""> Body: { "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</api>	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	Πηορεηκα οδηαδοτκα
POST /api/create_pet_simple — 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" }</api>	Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."	Проверка обработки некорректных типов данных.
POST /api/create_pet_simple — 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</api>	Код ответа 200 (ОК). Тело ответа: { "pets": [] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.	Полная проверка корректности всех элементов запроса.

GET /api/key

Вызов АРІ	Результат выполнения запроса	Примечание
GET https://petfriends.skillfactory.ru/api/key Headers: email: stron@gmail.com password: qwertyuiop	Код ответа 200 (ОК) 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</api>	Позитивный тест с корректным логином и паролем.
GET https://petfriends.skillfactory.ru/api/key Headers: email: invalid_user password: qwertyuiop	Время выполнения: не дольше 1с 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	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	Время выполнения: не дольше 1с 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	Негативный тест с некорректным паролем. Пароль должен быть корректным, но введен неверный пароль.
GET https://petfriends.skillfactory.ru/api/key Headers: (empty)	Время выполнения: не дольше 1с 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.

	via: 1.1 vegur	
GET https://petfriends.skillfactory.ru/api/key Headers: email: (empty) password: (empty)	Время выполнения: не дольше 1с 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	Негативный тест с пустыми значениями логина и пароля. Поля не должны быть пустыми.
GET /api/key — 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.
GET /api/key — 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.
GET /api/key — 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.
GET /арі/key — Проверка граничных значений для бизнес-логики Пример: Добавление питомца с максимальной длиной имени (255 символов) POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <api ключ=""> Body: { "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</api>	Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Тело ответа: { "id": " <pet_id>", "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</pet_id>	Проверка на успешное добавление питомца с максимальной длиной имени.

ААААААААААААААААААААААААААААААААААААА	Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."	Проверка обработки некорректных типов данных.
GET /арі/кеу — 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</api>	Код ответа 200 (ОК). Тело ответа: { "pets": [] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.	Полная проверка корректности всех элементов запроса.

GET /api/pets

Вызов АРІ	Результат выполнения запроса	Примечание
POST	Код ответа 200 (ОК)	Позитивный тест с
https://petfriends.skillfactory.ru/api/pets	Тело ответа: {	корректными данными и
Headers:	"id": " <pet id="">",</pet>	фото.
auth key: <api ключ=""></api>	"name": "Маффин",	
Body:	"animal type": "Лабрадор",	
{	"age": "4",	
"name": "Маффин",	"pet photo": " <photo url="">"</photo>	
"animal type": "Лабрадор",	}	
"age": "4",	Заголовки ответа:	
"pet photo": "images/001.jpg"	Обязательно:	
per_photo : mages/001.jpg	content-type: application/json	
,	date: <gara></gara>	
	Опционально:	
	connection: keep-alive	
	server: gunicorn/20.0.4	
	via: 1.1 vegur	
	via. 1.1 vegui	
	Время выполнения: не дольше 1с	
POST	Код ответа 403 (Forbidden)	Негативный тест без
	` '	
https://petfriends.skillfactory.ru/api/pets Headers:	Тело ответа: НТМL-код с сообщением о	авторизационного ключа.
	неавторизованном доступе	
auth_key: (empty)	Заголовки ответа:	
Body:	Обязательно:	
{ 	content-type: text/html; charset=utf-8	
"name": "Маффин",	date: <zara></zara>	
"animal_type": "Лабрадор",	Опционально:	
"age": "4",	connection: keep-alive	
"pet_photo": "images/001.jpg"	server: gunicorn/20.0.4	
}	via: 1.1 vegur	
	Время выполнения: не дольше 1с	
POST	Код ответа 400 (Bad Request)	Негативный тест с
https://petfriends.skillfactory.ru/api/pets	Тело ответа: {"error": "Поле 'name'	отсутствующим
Headers:	обязательно"}	обязательным полем 'name'.
neaders. auth key: <api ключ=""></api>	Заголовки ответа:	оолзательным полем паше.
	Заголовки ответа. Обязательно:	
Body:		
{	content-type: application/json	
"animal_type": "Лабрадор",	date: <pre><pre><pre><pre>date: <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	
"age": "4",	Опционально:	
"pet_photo": "images/001.jpg"	connection: keep-alive	
}	server: gunicorn/20.0.4	
	via: 1.1 vegur	
	Время выполнения: не дольше 1с	
POST	Код ответа 400 (Bad Request)	Негативный тест с
https://petfriends.skillfactory.ru/api/pets	Тело ответа: {"error": "Формат	некорректным форматом
Headers:	изображения некорректен"}	изображения.
Treaucis.	изооражения некорректен }	изооражения.

auth_key: <api ключ=""></api>	Заголовки ответа:	
Body:	Обязательно:	
{	content-type: application/json	
"name": "Маффин",	date: <дата>	
"animal type": "Лабрадор",	Опционально:	
"age": "4",	connection: keep-alive	
"pet photo": "images/invalid file.txt"	server: gunicorn/20.0.4	
]}	via: 1.1 vegur	
,		
	Время выполнения: не дольше 1с	
POST	Код ответа 200 (ОК)	Позитивный тест с
https://petfriends.skillfactory.ru/api/pets	Тело ответа: {	граничным значением
Headers:	"id": " <pet_id>",</pet_id>	имени (очень длинное имя).
auth_key: <api ключ=""></api>	"name":	
Body:	"AAAAAAAAAAAAAAAAAAAA	
\ {	AAAAAAAAAAAAAAAAAAAAA	
"name":	AAAAAAAAAAAAAAAAAAAAAA	
"AAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAA,",	
AAAAAAAAAAAAAAAAAAAAAA	"animal type": "Лабрадор",	
AAAAAAAAAAAAAAAAAAAAAAA	"age": "4",	
AAAAAAAAAAAAAAAA,",	"pet photo": " <photo url="">"</photo>	
"animal_type": "Лабрадор",	}	
"age": "4",	Заголовки ответа:	
"pet photo": "images/001.jpg"	Обязательно:	
}	content-type: application/json	
	date: <дата>	
	Опционально:	
	connection: keep-alive	
	server: gunicorn/20.0.4	
	via: 1.1 vegur	
	, 212 . 1 <i>g</i>	
	Время выполнения: не дольше 1с	
GET /api/pets — граничные значения	Код ответа 200 (ОК) для корректных	Проверка граничных
для бизнес-логики	граничных значений, код ответа 400 или	значений для логики
	422 для некорректных значений.	приложения, таких как
		максимальный возраст или
		длина имени питомца.
GET /api/pets — граничные значения	Код ответа 400 (Bad Request) для	Проверка обработки
для типов данных	некорректных типов данных, код ответа	значений типов данных на
	200 (ОК) для корректных типов.	границах допустимого
		диапазона, таких как
		максимальная длина
		строки.
	I	- F >

GET /api/pets — проверка всех	Тест включает проверку заголовков,	Полная проверка
составляющих запроса	тела запроса, кода ответа и времени	корректности всех
oo waxaa aan aan gaa aa	выполнения (не дольше 1 секунды).	элементов запроса.
GET /api/pets — Проверка граничных	Код ответа 200 (ОК). Питомец успешно	Проверка на успешное
значений для бизнес-логики	создан с указанными данными.	добавление питомца с
Пример: Добавление питомца с	Тело ответа:	максимальной длиной
максимальной длиной имени (255	fesio orbera.	имени.
символов)	"id": " <pet id="">",</pet>	MINICITY.
POST	"name":	
https://petfriends.skillfactory.ru/api/pets	"AAAAAAAAAAAAAAAAAAAAA	
Headers:	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
auth key: <api ключ=""></api>	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
Body:	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
Body.	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
"name":	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
	· · · · · · · · · · · · · · · · · · ·	
AAAAAAAAAAAAAAAAAAAAAAAA	"animal_type": "Лабрадор",	
AAAAAAAAAAAAAAAAAAAAA	"age": "4",	
AAAAAAAAAAAAAAAAAAA	"pet_photo": " <photo_url>"</photo_url>	
AAAAAAAAAAAAAAAAAAAA	}	
AAAAAAAAAAAAAAAAAAAAAA		
AAAAAAAAAAAAAA, "animal_type": "Лабрадор", "age": "4",		
"pet_photo": "images/001.jpg" }		
GET /api/pets — Проверка граничных	Код ответа 400 (Bad Request). Питомец	Проверка обработки
значений для типов данных	не добавлен. Сообщение об ошибке:	некорректных типов
Пример: Добавление питомца с	"Возраст не может быть	данных.
некорректным возрастом	отрицательным."	
(отрицательное значение)		
POST		
https://petfriends.skillfactory.ru/api/pets		
Headers:		
auth_key: <api ключ=""></api>		
Body:		
{		
"name": "Маффин",		
"animal_type": "Лабрадор",		
"age": "-1",		
"pet_photo": "images/001.jpg"		
}		
GET /api/pets — Проверка всех	Код ответа 200 (ОК).	Полная проверка
составляющих запроса	Тело ответа:	корректности всех
Пример: Проверка заголовков, тела	 {	элементов запроса.
запроса, кода ответа и времени	"pets": []	
выполнения.]}	

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

GET /api/pets

Вызов АРІ	Результат выполнения запроса	Примечание
GEThttps://petfriends.skillfactory.ru/api/p	Код ответа 200 (ОК)	Позитивный тест с
ets?filter=my pets	Содержит список питомцев,	опциональными
Headers:	принадлежащих пользователю-	параметрами.
auth key: Корректный ключ API	владельцу ключа АРІ	параметрами.
Query params:	Заголовки ответа те же, что в тесте #1	Так как здесь мы в строке
filter=my pets	Время выполнения: не дольше 1с	запроса используем
mice my_pers	Бремя выполнения: не дольше те	параметр filter,
		который не является
		обязательным,
		и устанавливаем его
		значение =my pets.
		Мы всё ещё ожидаем, что
		запрос корректно
		выполнится, поэтому
		сценарий позитивный.
		, 1
		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/p	Код ответа 403 (Forbidden)	Негативный тест с
ets	Тело ответа: HTML-код с сообщением о	некорректным вводом.
Headers:	неавторизованном пользователе	
auth_key: 5	Заголовки ответаОбязательно:content-	Негативный, потому что мы
Query params:	type: text/html charset=utf-8	ожидаем
filter=my_pets	date: <дата>	в результатах выполнения
		ошибку, а с некорректным
	Опционально:connection: keep-alive	вводом, потому что
	server: gunicorn/20.0.4	auth_key является string
	via: 1.1 vegur	типом, а мы пытаемся
		передать int.
	Время выполнения: не дольше 1с	
GEThttps://petfriends.skillfactory.ru/api/p	Код ответа 403 (Forbidden)	Негативный тест с
ets	Тело ответа: HTML-код с сообщением о	некорректным вводом.
Cio	1 0.10 01be1a. 1111vill-rod c cooomeniem 0	покоррективым вводом.

Headers: auth_key: abc Query params: filter=my_pets	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	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.
	Время выполнения: не дольше 1с	
GEThttps://petfriends.skillfactory.ru/api/p ets Headers: auth_key: корректный ключ API Query params: filter=pets	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>	Негативный тест с некорректным вводом. Негативный, потому что мы ожидаем в результатах выполнения ошибку,
	Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1c	с некорректным вводом, потому что мы передаем в filter значение, непредусмотренное системой.
PUT/PATCH/DELETE https://petfriends.skillfactory.ru/api/pets Headers: auth_key: корректный ключ API Query params: filter=my_pets	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> Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Деструктивное тестирование. Мы пытаемся сломать систему, вызывая известный эндпоинт с неподдерживаемым типом запроса.
GET (Control of the Control of the C	Время выполнения: не дольше 1с Код ответа 415 (Unsupported content-	Здесь через ";" разделены
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	type) Тело ответа: HTML код с сообщением о недопустимом формате данных Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата>	разные значения заголовка Сопtent-Туре. То есть этот кейс повторяется 4 раза для каждого значения Сопtent-Туре.
Query params: filter=my_pets	Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1c	Этот тест также является разрушительным, так как мы пытаемся передать неподдерживаемый тип данных в запрос.

GET https://petfriends.skillfactory.ru/api/pets Headers:	Код ответа 400 (Bad Request) Тело ответа: HTML-код с сообщением о неверном значении параметра filter.	Этот тест является разрушительным.
auth_key: корректный ключ API Query params: filter=строка > 256 символов filter=строка спецсимволов	Заголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата>	Мы пытаемся привести наше приложение к падению — строка длиннее,
filter=строка символов из кириллицы	Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	чем может вмещать в себя тип данных string, спецсимволы некорректно обрабатываются
	Время выполнения: не дольше 1с	на стороне сервера, кириллица неверно обрабатывается на стороне сервера.
GET /api/pets (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.
GET /api/pets (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.
GET /api/pets (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.
GET /api/pets (Пример) — Проверка граничных значений для бизнес-логики Пример: Добавление питомца с максимальной длиной имени (255 символов) POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <api ключ=""> Body: { "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</api>	Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Тело ответа: { "id": " <pet_id>", "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</pet_id>	Проверка на успешное добавление питомца с максимальной длиной имени.

ААААААААААААААААААААААААААААААААААААА	Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."	Проверка обработки некорректных типов данных.
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</api>	Код ответа 200 (ОК). Тело ответа: { "pets": [] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.	Полная проверка корректности всех элементов запроса.

POST /api/pets/set_photo/{pet_id}

Вызов АРІ	Результат выполнения запроса	Примечание
POST	Код ответа 200 (Created)	Базовый позитивный тест
https://petfriends.skillfactory.ru/api/pets/se	В теле приходит json с корректной	
t_photo/{pet_id}	структурой	
Headers:	В теле приходят следующие поля в	
auth_key: Корректный ключ API	объектах pets:age: целое число,	
Path params:	animal_type: строка,	
pet_id=корректный id питомца	created_at: строка со снимком времени	
Payload:	(timestamp)	
корректный файл	id: uuid	
	пате: строка	
	pet_photo: base64 image	
	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	
POST https://petfriends.skillfactory.ru/api/pets/se t_photo/{pet_id} Headers: auth_key: Корректный ключ API Path params: pet_id=корректный id питомца Payload: текстовый файл	Время выполнения: не дольше 1с 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> Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с корректным вводом
POST https://petfriends.skillfactory.ru/api/pets/se t_photo/{pet_id} Headers: auth_key: Корректный ключ API Path params: pet_id=некорректный id питомца Payload: графический файл	Время выполнения: не дольше 1с 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> Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с корректным вводом
POST https://petfriends.skillfactory.ru/api/pets/se t_photo/{pet_id} Headers: auth_key: Корректный ключ API Path params: pet_id=корректный id питомца Payload: загрузить фотографию второй раз	Время выполнения: не дольше 1с 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>	Негативный тест с корректным вводом
	Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	

Время выполнения: не дольше 1с Код ответа 400 (Bad Request) В теле ответа HTML код с сообщением отом, что отсутствует файл Ваголовки ответаОбязательно:content-type: text/html charset=utf-8 date: <дата> Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	Негативный тест с некорректным вводом
•	
•	
าจท	nan
мап Код ответа 400 (Bad Request) В теле ответа HTML код с сообщением об испорченном файле Ваголовки ответаОбязательно:content- type: text/html charset=utf-8 date: <дата> Опционально:connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur	лап Деструктивное тестирование
Время выполнения: не лольше 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.
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.
Гест включает проверку заголовков,	Полная проверка
гела запроса, кода ответа и времени	корректности всех
выполнения (не дольше 1 секунды).	элементов запроса.
Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Гело ответа:	Проверка на успешное добавление питомца с максимальной длиной имени.
Вр Ко Те Не 20	ремя выполнения: не дольше 1с од ответа 200 (ОК) для корректных ваничных значений, код ответа 400 или 22 для некорректных значений. од ответа 400 (Bad Request) для екорректных типов данных, код ответа 200 (ОК) для корректных типов.

POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <api ключ=""> Body: { "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</api>	"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
POST /api/pets/set_photo/{pet_id} (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: { "name": "Маффин", "animal_type": "Лабрадор", "age": "-1", "pet_photo": "images/001.jpg" }</api>	Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."	Проверка обработки некорректных типов данных.
POST /api/pets/set_photo/{pet_id} (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:</api>	Код ответа 200 (ОК). Тело ответа: { "pets": [] } Заголовки ответа: - content-type: application/json - date: <дата> Время выполнения запроса: < 1 сек.	Полная проверка корректности всех элементов запроса.

filter=my_pets

DELETE /api/pets/{pet_id}

Вызов АРІ	Результат выполнения запроса	Примечание
DELETE	Код ответа 200 (ОК)	Позитивный тест с
https://petfriends.skillfactory.ru/api/pets/{	Тело ответа: {"message": "Питомец	корректным
pet_id}	удален"}	идентификатором питомца.
Headers:	Заголовки ответа:	
auth_key: <api ключ=""></api>	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	
DELETE	Код ответа 403 (Forbidden)	Негативный тест без
https://petfriends.skillfactory.ru/api/pets/{	Тело ответа: HTML-код с сообщением о	авторизационного ключа.
pet_id}	неавторизованном доступе	
Headers:	Заголовки ответа:	
auth_key: (empty)	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	
DELETE	Код ответа 404 (Not Found)	Негативный тест с
https://petfriends.skillfactory.ru/api/pets/in	Тело ответа: {"error": "Питомец не	некорректным
valid_pet_id	найден"}	идентификатором питомца.
Headers:	Заголовки ответа:	
auth_key: <api ключ=""></api>	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	
DELETE	Код ответа 400 (Bad Request)	Негативный тест с пустым
https://petfriends.skillfactory.ru/api/pets/	Тело ответа: {"error": "Некорректный	идентификатором питомца.
Headers:	запрос"}	
auth_key: <api ключ=""></api>	Заголовки ответа:	
	Обязательно:	
	content-type: application/json	
	date: <дата>	

DELETE https://petfriends.skillfactory.ru/api/pets/{ pet_id} Headers: auth_key: invalid_key	Опционально: connection: keep-alive server: gunicorn/20.0.4 via: 1.1 vegur Время выполнения: не дольше 1с 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	Негативный тест с некорректным авторизационным ключом.
DELETE https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <api ключ=""> Query params: filter=my_pets</api>	Время выполнения: не дольше 1с Код ответа 200 (ОК) 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	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)</api>	Время выполнения: не дольше 1с 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	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 (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.
DELETE /api/pets/{pet id} — граничные	Код ответа 400 (Bad Request) для	Проверка обработки
значения для типов данных	некорректных типов данных, код ответа	значений типов данных на
значения для типов данных	200 (ОК) для корректных типов.	границах допустимого
	200 (ОК) для корректных типов.	диапазона, таких как
		максимальная длина
DELETE /- n:/n -t-/(n-t :1)	Т	строки.
DELETE /api/pets/{pet_id} — проверка	Тест включает проверку заголовков,	Полная проверка
всех составляющих запроса	тела запроса, кода ответа и времени	корректности всех
DELETE / :/ . /(: 1) II	выполнения (не дольше 1 секунды).	элементов запроса.
DELETE /api/pets/{pet_id} — Проверка	Код ответа 200 (ОК). Питомец успешно	Проверка на успешное
граничных значений для бизнес-логики	создан с указанными данными.	добавление питомца с
Пример: Добавление питомца с	Тело ответа:	максимальной длиной
максимальной длиной имени (255	{	имени.
символов)	"id": " <pet_id>",</pet_id>	
POST	"name":	
https://petfriends.skillfactory.ru/api/pets	"AAAAAAAAAAAAAAAAAAAAA	
Headers:	AAAAAAAAAAAAAAAAAAAAAA	
auth_key: <api ключ=""></api>	AAAAAAAAAAAAAAAAAAAAAAA	
Body:	AAAAAAAAAAAAAAAAAAAAAAAA	
 {	AAAAAAAAAAAAAAAAAAAAAAAAA	
"name":	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
"AAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAA,",	
AAAAAAAAAAAAAAAAAAA	"animal_type": "Лабрадор",	
AAAAAAAAAAAAAAAAAAA	"age": "4",	
AAAAAAAAAAAAAAAAAAAA	"pet_photo": " <photo_url>"</photo_url>	
AAAAAAAAAAAAAAAAAAAA	}	
AAAAAAAAAAAAAAAAAAAA		
AAAAAAAAAAAAAAAAAAAA		
AA",		
"animal_type": "Лабрадор",		
"age": "4",		
"pet_photo": "images/001.jpg"		
}		
DELETE /api/pets/{pet_id} — Проверка	Код ответа 400 (Bad Request). Питомец	Проверка обработки
граничных значений для типов данных	не добавлен. Сообщение об ошибке:	некорректных типов
Пример: Добавление питомца с	"Возраст не может быть	данных.
некорректным возрастом	отрицательным."	
(отрицательное значение)		
POST		
https://petfriends.skillfactory.ru/api/pets		
Headers:		
auth_key: <api ключ=""></api>		
Body:		
{		

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

PUT /api/pets/{pet_id}

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

1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	T (II II III III	
https://petfriends.skillfactory.ru/api/pets/in	Тело ответа: {"error": "Питомец не	некорректным pet_id.
valid_pet_id	найден"}	
Headers:	Заголовки ответа:	
auth_key: <api ключ=""></api>	Обязательно:	
Body:	content-type: application/json	
{	date: <дата>	
"name": "Бублик",	Опционально:	
"animal_type": "Пудель",	connection: keep-alive	
"age": "3"	server: gunicorn/20.0.4	
}	via: 1.1 vegur	
	D 1	
DI IT	Время выполнения: не дольше 1с	
PUT	Код ответа 400 (Bad Request)	Негативный тест с
https://petfriends.skillfactory.ru/api/pets/{	Тело ответа: {"error": "Поле 'name'	отсутствующим
pet_id}	обязательно"}	обязательным полем 'name'.
Headers:	Заголовки ответа:	
auth_key: <api ключ=""></api>	Обязательно:	
Body:	content-type: application/json	
{ 	date: <дата>	
"animal_type": "Пудель",	Опционально:	
"age": "3"	connection: keep-alive	
}	server: gunicorn/20.0.4	
	via: 1.1 vegur	
	Drawa nyanyawa ya ya ya ya ya 1a	
PUT	Время выполнения: не дольше 1с Код ответа 200 (ОК)	Позитивный тест с
https://petfriends.skillfactory.ru/api/pets/{	Тело ответа: {	
pet id}	"id": " <pet id="">",</pet>	граничным значением имени (очень длинное имя).
Headers:	"name":	имени (очень длинное имя).
auth key: <api ключ=""></api>	"AAAAAAAAAAAAAAAAAAAAA	
Body:	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
bouy.	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
"name":	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
"AAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA		
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	AAAAAAAAAAAAAA,", "animal type": "Пудель",	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	"age": "3"	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	age. 3	
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	Sarahanin amera:	
	Заголовки ответа: Обязательно:	
AAA",		
"animal_type": "Пудель", "age": "3"	content-type: application/json	
age . 3	date: <дата>	
}	Опционально:	
	connection: keep-alive	
	server: gunicorn/20.0.4	
	via: 1.1 vegur	

	Время выполнения: не дольше 1с	
PUT /api/pets/{pet_id} — 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.
PUT /api/pets/{pet_id} — 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.
PUT /api/pets/{pet_id} — 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.
РUТ /арі/реts/{pet_id} — Проверка граничных значений для бизнес-логики Пример: Добавление питомца с максимальной длиной имени (255 символов) POST https://petfriends.skillfactory.ru/api/pets Headers: auth_key: <api ключ=""> Body: { "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</api>	Код ответа 200 (ОК). Питомец успешно создан с указанными данными. Тело ответа: { "id": " <pet_id>", "name": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</pet_id>	Проверка на успешное добавление питомца с максимальной длиной имени.
PUT /api/pets/{pet_id} — Проверка граничных значений для типов данных Пример: Добавление питомца с некорректным возрастом (отрицательное значение) POST	Код ответа 400 (Bad Request). Питомец не добавлен. Сообщение об ошибке: "Возраст не может быть отрицательным."	Проверка обработки некорректных типов данных.

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

Business Logic Tests

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

auth key: <api ключ=""></api>		
1. POST	1. Код ответа 200 (ОК). Первый питомец	Добавление нескольких
https://petfriends.skillfactory.ru/api/pets	добавлен.	питомцев и получение
Headers:	2. Код ответа 200 (ОК). Второй питомец	списка всех питомцев
auth key: <api ключ=""></api>	добавлен.	пользователя.
Body:	3. Код ответа 200 (ОК). Возвращен	nonbsobaresis.
Body.	список питомцев, содержащий	
"name": "Маффин",	добавленных питомцев.	
"animal type": "Лабрадор",	досивленных питомцев.	
"age": "4",		
"pet photo": "images/001.jpg"		
pet_photo: mages/001.jpg		
2. POST		
https://petfriends.skillfactory.ru/api/pets		
Headers:		
auth key: <api ключ=""></api>		
Body:		
{		
"name": "Бублик",		
"animal type": "Пудель",		
"age": "3",		
"pet photo": "images/002.jpg"		
pet_photo : mages/002.jpg		
3. GET		
https://petfriends.skillfactory.ru/api/pets		
Headers:		
auth key: <api ключ=""></api>		
Query params:		
filter=my pets		
1. DELETE	1. Код ответа 404 (Not Found). Ошибка:	Удаление
https://petfriends.skillfactory.ru/api/pets/in	Питомец не найден.	несуществующего питомца,
valid_pet_id	титомец не наиден.	проверка корректной
Headers:		обработки ошибки.
auth key: <api ключ=""></api>		оораоотки ошиоки.
1. POST	1. Код ответа 200 (ОК). Питомец создан.	Проверка обработки
https://petfriends.skillfactory.ru/api/pets	2. Код ответа 400 (Bad Request).	некорректных данных при
Headers:	Ошибка: Некорректные данные.	обновлении питомца.
auth key: <api ключ=""></api>	отнока. Пекоррективе данные.	обновлении питомца.
Body:		
Souy.		
"name": "Маффин",		
"animal type": "Лабрадор",		
"age": "4",		
"pet photo": "images/001.jpg"		
}		
2. PUT		
https://petfriends.skillfactory.ru/api/pets/{		
pet id}		
Headers:		
auth key: <api ключ=""></api>		
Body:		
Douy.		

	T	1
\{ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \		
"name": "",		
"animal_type": "",		
"age": "invalid_age"		
}		
1. POST	1. Код ответа 200 (ОК). Питомец создан.	Проверка логики
https://petfriends.skillfactory.ru/api/pets	2. Код ответа 200 (ОК). Данные питомца	получения данных питомца
Headers:	совпадают с вводом.	после его создания.
auth_key: <api ключ=""></api>		
Body:		
\{		
"пате": "Маффин",		
"animal type": "Лабрадор",		
"age": "4",		
"pet photo": "images/001.jpg"		
}		
2. GET		
https://petfriends.skillfactory.ru/api/pets/{		
pet id}		
Headers:		
auth key: <api ключ=""></api>		
1. POST	1. Код ответа 200 (ОК). Питомец создан.	Удаление питомца и
https://petfriends.skillfactory.ru/api/pets	2. Код ответа 200 (ОК). Питомец удален.	проверка его
Headers:	3. Код ответа 404 (Not Found). Ошибка:	недоступности в системе.
auth key: <api ключ=""></api>	Питомец не найден.	medecifiniocini b enereme.
Body:	Timromed no nangen.	
{		
"name": "Маффин",		
"animal type": "Лабрадор",		
"age": "4",		
"pet photo": "images/001.jpg"		
pet_photo : mages/001.jpg		
2. DELETE		
https://petfriends.skillfactory.ru/api/pets/{		
pet_id} Headers:		
auth_key: <api ключ=""></api>		
3. GET		
https://pattriands.gkilltootory.ru/oni/nots/1		
https://petfriends.skillfactory.ru/api/pets/{		
pet_id}		