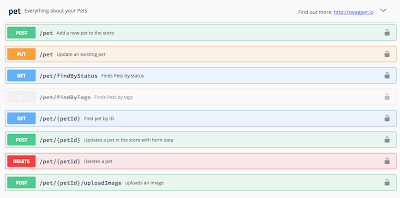
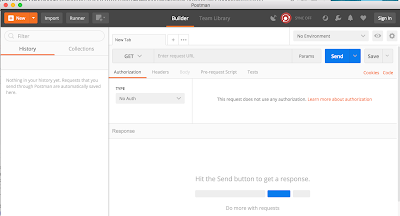
Если у вашего API есть Swagger-файл, то будет легко увидеть, какие типы запросов API позволяет, и какие параметры необходимы этим запросам. Разработчики Swagger также создали образец приложения, которое можно использовать для практики REST-запросов: [http://petstore.swagger.io](http://petstore.swagger.io/). Этот сайт имитирует онлайн-зоомагазин, и пользователи могут добавлять и получать информацию о своих питомцах.



Зайдя на этот сайт, вы увидите, что там доступно множество возможных запросов к Pet. давайте посмотрим на запрос GET к /pet/{petID}. Нажмите на этот запрос, а затем на кнопку "Try it out". Вы увидите, что petId – это необходимый параметр. Введите 1 в поле параметра, и нажмите на "Execute". Проскролльте вниз к секции "Response Body", и вы увидите, что сервер вернул информацию о питомце! Проскролльте немного выше и посмотрите на URL запроса: [**http://petstore.swagger.io/v2/pet/1**](http://petstore.swagger.io/v2/pet/1)**.**Мы будем использовать этот URL, чтобы учиться делать запросы через Postman.

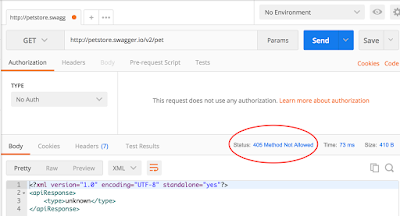
Postman – лучший из известных мне инструмент тестирования REST-запросов. Он бесплатно доступен на сайте [https://www.getpostman.com](https://www.getpostman.com/), а еще у него есть платная командная версия. Если у вас еще нет Postman, скачайте и запустите его.



Postman должен стартовать на открытой вкладке, настроенной на использование GET-запроса. Все, что вам теперь остается сделать – это ввести URL запроса, который мы использовали ранее, и нажать "Send". Вы должны получить ответ, аналогичный ответу на этот запрос в Swagger.

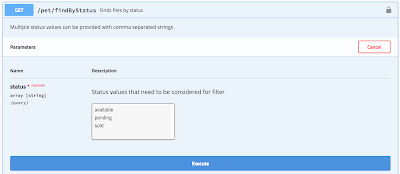
Теперь, когда мы разобрались, как работают GET-запросы, давайте подумаем, как же их тестировать! Очевидно, что мы протестировали "счастливый" сценарий, получив питомца с id 1. Что будет, если мы изменим id-параметр на 2? А если на -1? На 0? На "foo"? Тут стоит упомянуть, что Swagger Pet Store – тестовое приложение, и вы можете увидеть в нем поведение, которое в реальном приложении будет нежелательным. К примеру, я получила результат, тестируя с параметром -1. Как правило, ожидается, что ID объектов данных не могут быть отрицательными, и в реальном приложении это было бы багом.

Давайте посмотрим, что будет, если мы вообще не будем вводить параметр, и наш URL запроса будет просто выглядеть как [**http://petstore.swagger.io/v2/pet**](http://petstore.swagger.io/v2/pet)**.**Мы получим XML-ответ, который гласит "unknown". Можно также заметить, что код ответа – 405 Method Not Allowed.



Коды статуса, полученные после отправки REST-запроса, сообщают вам о поведении вашего приложения. Мы еще будем их обсуждать, а пока давайте запомним, что код 200 – это хорошо, а код, начинающийся с 4, обычно означает, что что-то пошло не так. Скажем, вы можете получить статус 404 для питомца, которого не существует. Этот статус означает, что запись не найдена.

Вернемся к Swagger Pet Store и посмотрим на еще один GET-запрос. В этот раз это запрос pets/findByStatus. Нажмите на запрос, а затем на "Try it out".



Отметим, что для выполнения этого запроса нужно выбрать статус. Давайте выберем статус "sold" и нажмем на "Execute". Если вы проскроллите вниз до окна ответа, вы увидите, что сервер вернул множество питомцев. Теперь проскролльте чуть выше и посмотрите на URL запроса:

[**http://petstore.swagger.io/v2/pet/findByStatus?status=sold**](http://petstore.swagger.io/v2/pet/findByStatus?status=sold).

Вопросительный знак в URL означает, что мы использовали параметр запроса. Это немного отличается от параметра пути, который мы наблюдали, запрашивая GET по id питомца. Параметры запроса всегда начинаются с вопросительного знака. Затем идет имя параметра, знак равенства, и значение параметра, которое мы хотим использовать.

Скопируйте запрос и запустите его в Postman. Просто нажмите на кнопку "+" вверху экрана, чтобы открыть вкладку и создать новый запрос. По умолчанию он всегда GET. Введите URL и нажмите "Send". Вы должны получить результат, аналогичный прогону запроса в Swagger.

Мы протестировали один успешный сценарий для этого запроса. Его также можно прогнать с использованием значений параметра "available" и "pending". Вы можете запустить запрос [**http://petstore.swagger.io/v2/pet/findByStatus?status=pending,sold**](http://petstore.swagger.io/v2/pet/findByStatus?status=pending,sold)и получить всех забронированных и всех проданных питомцев! А что будет, если передать в параметре "foo"? вы получите пустой набор результатов. Это отличается от ответа, который мы получали, посылая "foo" как параметр для GET-запроса питомца по ID, и демонстрирует разницу между параметром запроса и параметром пути. Если параметра пути не существует, то обычно вы получите ошибку 400. Если не существует параметра запроса, то вы получите ответ 200 и пустой набор результатов в теле ответа.

Теперь, когда вы попробовали запустить два разных GET-запроса, можно тестировать дальше, манипулируя URL запросов различным образом. К примеру, посмотреть, что будет, если вы сделаете запрос к https вместо http. Или выяснить, как поведет себя система, если вы поменяете "io" в запросе на "com", или "v2" на "v1". Можно удалить "/pet". Можно попробовать изменить тип запроса на POST или DELETE. Наблюдения за тем, что происходит, помогут вам понять, как работают REST-запросы, и даст вам пищу для идей тестирования вашего API.