

Postman Server Monitoring

Author	Сергей Погребняк
Last changes	18.12.2019
Swift	5.2
Xcode	11.3
Postman	7.14.0

- Введение
- 1 Добавление теста в запрос
- 2 Создание мониторинга
- 3 Запуск мониторинга
- 4 Просмотр результатов мониторинга
- Вместо заключения
- Фичи которые помогут лучше жить
 - 1 Настройка рассылки
 - 2 Просмотр количества использованных мониторинг запросов

Введение

Postman Server Monitoring - это инструмент приложения Postman который позволяет проводить регулярный мониторинг сервера и прогона тестов в заданный период времени или через заданный интервал, в случае не прохождения сервером тестов Postman автоматически делает рассылку на указанные e-mail-ы, отправит сообщение в Slack или другие соц сети. Это может быть полезным при одновременной разработке серверной и клиентской части, или после релиза приложения для проверки на корректность работы серверной части приложения.

Postman server monitoring не тоже самое что runner в Postman-е, это схоже с тем как мы в Postman-е пишем запрос и тесты для него, и запускаем их нажатием кнопки send, и визуально проверяем на вкладке tests что все тесты пройдены успешно. Postman server monitoring делает аналогичные действия только запустит все запросы в коллекции, если хотя б один тест не выполнится, он сообщит нам об этом.

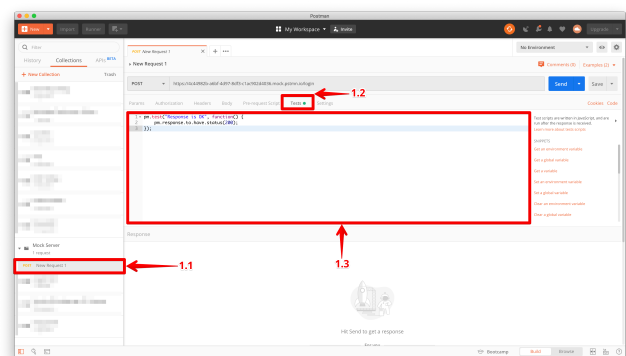
1 Добавление теста в запрос

1.1- Перейдем в наш запрос

1.2- Откроем вкладку «Tests»

1.3- Добавляем тест для проверки на успешность выполненного запроса

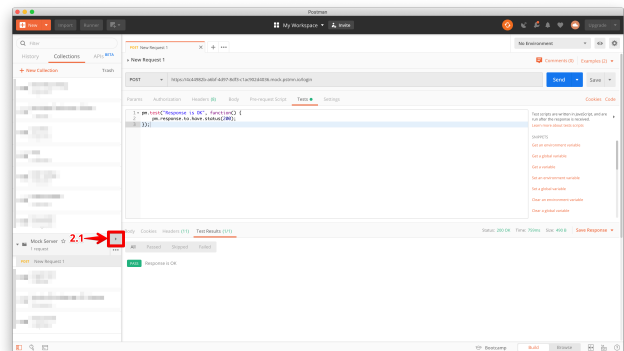
```
pm.test("response is ok", function() {  
  pm.response.to.have.status(200);  
});
```



2 Создание мониторинга

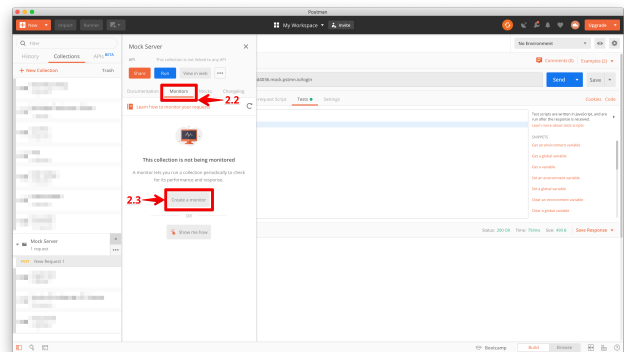
После того как у нас имеется созданный запрос и тест/ы для него, создадим мониторинг для нашей коллекции.

2.1- Разворачиваем список доступных действий для коллекции



2.2- Переходим на вкладку «Monitors»

2.3- И жмем на кнопку «Create monitor»



2.4- Вводим название для мониторинга

2.5- Выбираем версию (если она есть)

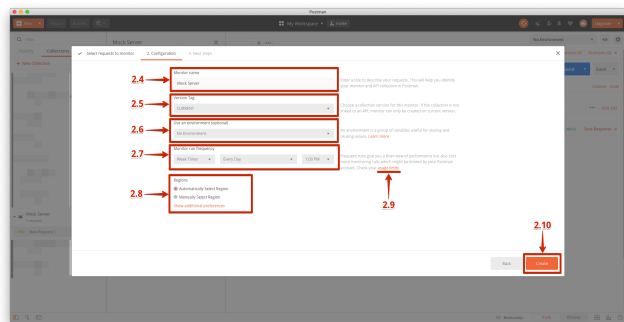
2.6- Выбираем окружение (если оно есть)

2.7- Задаем время или временной интервал, когда Postman должен прогнать запросы

2.8- Выбираем регион

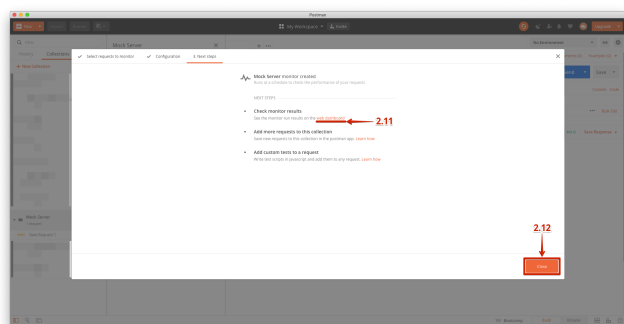
2.9- В бесплатной версии количество проверок ограничено, по данной ссылке можно пройти и посмотреть ограничение и количество оставшихся запросов

2.10- Создаем сам сервер мониторинга



2.11- Ссылка для просмотра результатов

2.12- Закрываем окно создания мониторинга

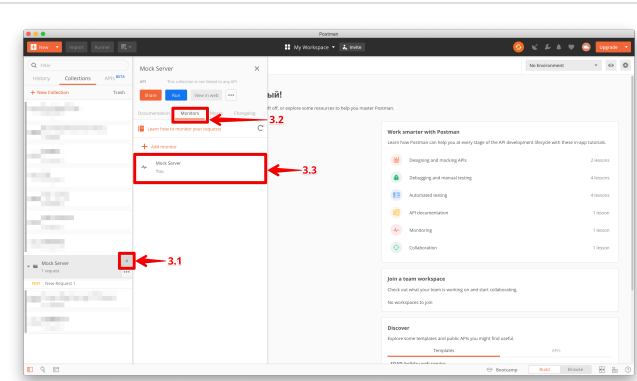


Мониторинг успешно создан, и будет работать пока мы его не удалим или не поставим на паузу. Также мониторинг будет отменен если у нас закончится лимит на запросы.

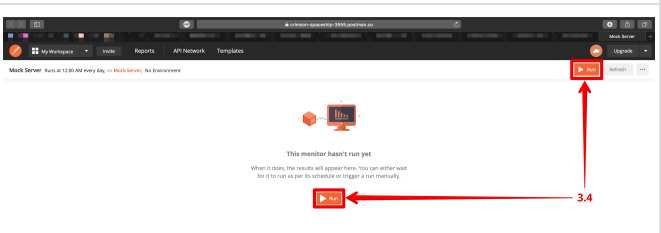
3 Запуск мониторинга

Возникают ситуации когда нужно запустить мониторинг в ручную, в данный момент времени а не ждать часа "X", для этого нужно сделать следующие шаги:

- 3.1- Открываем меню для нашей коллекции
- 3.2- Переходим на вкладку мониторинга
- 3.3- Нажимаем на созданный нами мониторинг, нас перебросит на веб версию нашего профиля

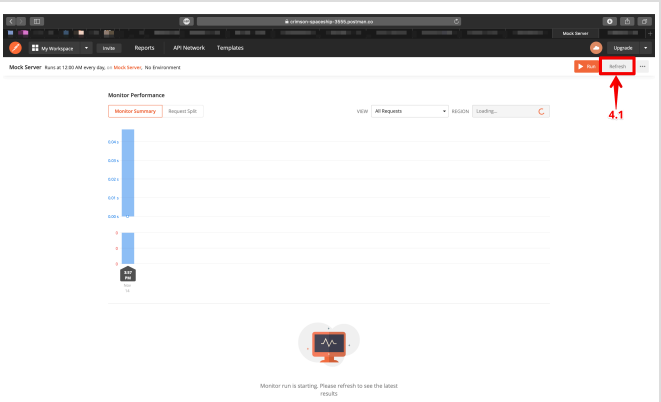


- 3.4- Для принудительно запуска мониторинга нажимаем на кнопку «Run»



4 Просмотр результатов мониторинга

- 4.1- Для того чтобы посмотреть результаты нажимаем кнопку «Refresh» (мониторинг может занимать длительное время)

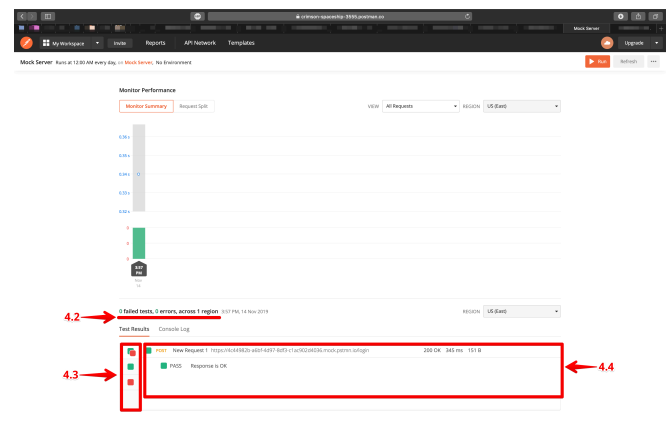


После прогона всех запросов мы увидим результат следующего вида:

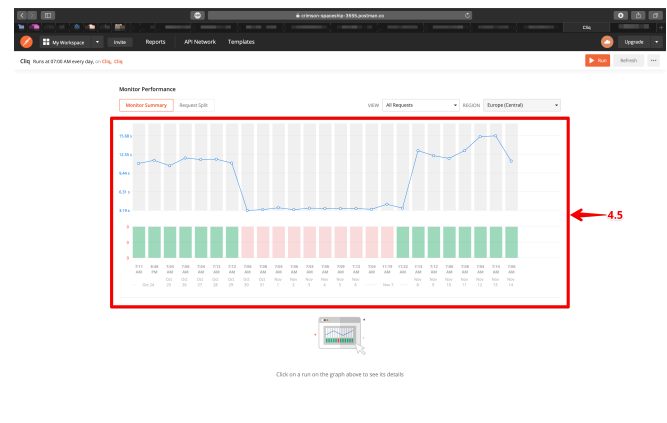
4.2- Количество не прошедших тестов

4.3- Можем отсортировать для того чтобы понять какие тесты у нас упали

4.4- Отображаются детали мониторинга для выбранного дня



4.5- Выбирая определенный день мы можем понять что с нашим сервером было не так и проанализировать – какие тесты были не пройдены



Вместо заключения

В этот момент возникает резонный вопрос – а почему это должен делать разработчик клиента а не сервера? Ответ – в идеальном мире это должен делать он, но частая ситуация когда сервер находится на одной стороне а клиент на другой, и сервер сломали – в результате происходит следующая ситуация – заказчик проснувшись утром открывает свое приложение и видит что оно не работает, и быстро бежит писать письмо менеджеру или даже на прямую деву о том какой ты плохой, что ты сделал все плохо... Разработчик приложения в панике не понимая как так, вить до этого оно работало 2 года, а тут упало. Начинает быстро все проверять и понимает что накосячил не он... а виной всему - сломанный сервер. Для того чтобы себя уберечь от таких вот ситуаций была создана коллекция запросов и запущена на постмене. Чтобы в следующий раз проснувшись утром увидев письмо с постмена, о том что тесты не пройдены, писать письмо серверу с тем что у них что-то снова сломалось...

В такой схеме это даст несколько преимуществ:

- разработчик приложения может добавить жизненно важные запросы для проверки и по приходу на работу проверить все ли с сервером хорошо
- такой подход экономит время на поиски и устранение крэшей и ошибок
- заказчику будет приятнее когда не он словит крэш, а мы увидим что есть проблема с сервером/клиентом, быстро ее устраним и скажем – была ошибочка мы все поправили все супер) Как результат - нам + в карму, заказчику + к настроению.
- ну и пожалуй главное – пользователи приложения всегда будут обеспечены жизненно важными функциями приложения, так как в случае не прохождения тестов мы сразу увидим это и будем принимать меры по восстановлению работоспособности нашего продукта
- довольные пользователи – хорошие рейтинги! В сторе будет меньше негативных отзывов о приложении.

Фичи которые помогут лучше жить

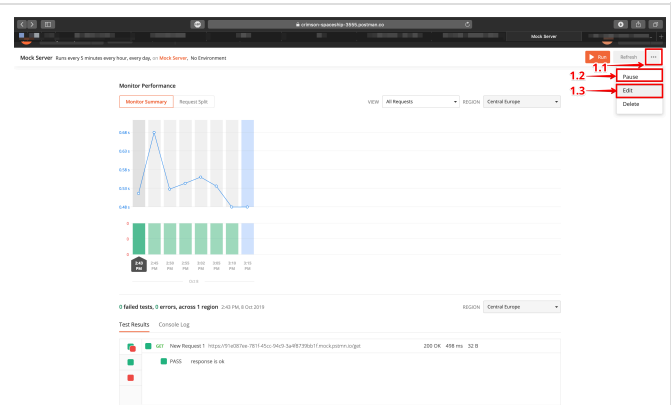
1 Настройка рассылки

Для того чтобы настроить рассылку почты нужно выполнить следующие действия:

1.1- Развернуть меню настроек нашего мониторинга

1.2- При необходимости поставить мониторинг на паузу (в виду того что на бесплатном аккаунте кол-во запросов ограничено нужная штука!)

1.3- Нажимаем кнопку «Edit» для перехода к настройкам нашего мониторинга



1.4- Разворачиваем/сворачиваем меню дополнительных настроек

1.5- Активируем почтовую рассылку в случае ошибки и добавляем почтовые ящики на какие должна делаться рассылка

1.6- Сохраняем внесенные изменения

MONITORS ► EDIT MONITOR

Edit monitor

Monitor Name

Mock Server

Collection

Mock Server

Version

CURRENT

Environment (optional)

No Environment

Schedule

Minute Timer

Every 5 Minutes

Regions

Auto select region

● Select regions manually

☐ US (East)

☐ Asia Pacific

☐ Canada (Central)

☒ Europe (Central)

☐ South America

☐ United Kingdom

☐ US (East, Static IP)

☐ US (West)

US (West, Static IP)

1.4

1.5

Hide Additional Preferences ▴

Receive email notifications for run failures and errors

Notification Recipients

+ Add another recipient

Add

pogrebnyak.ve...

×

You can add up to 5 notification recipients

Request timeout

Delay between requests

Don't follow redirects

Disable SSL validation

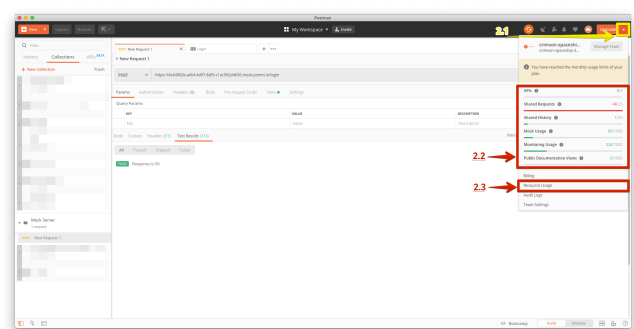
1.6

Cancel

Save Changes

2 Просмотр количества использованных мониторинг запросов

- 2.1- Разворачиваем настройки нашего workspace
- 2.2- Отображаются данные о количестве использованных ресурсов в 30 дневный период
- 2.3- Можем перейти на сайт для смены плана или просмотра данных в веб версии



2.4- Тип текущего плана

2.5- Период использования данных (каждый месяц некоторые данные сбрасываются и доступны снова)

2.6- Переход на новый тарифный план

2.7- Таблица использованных данных

