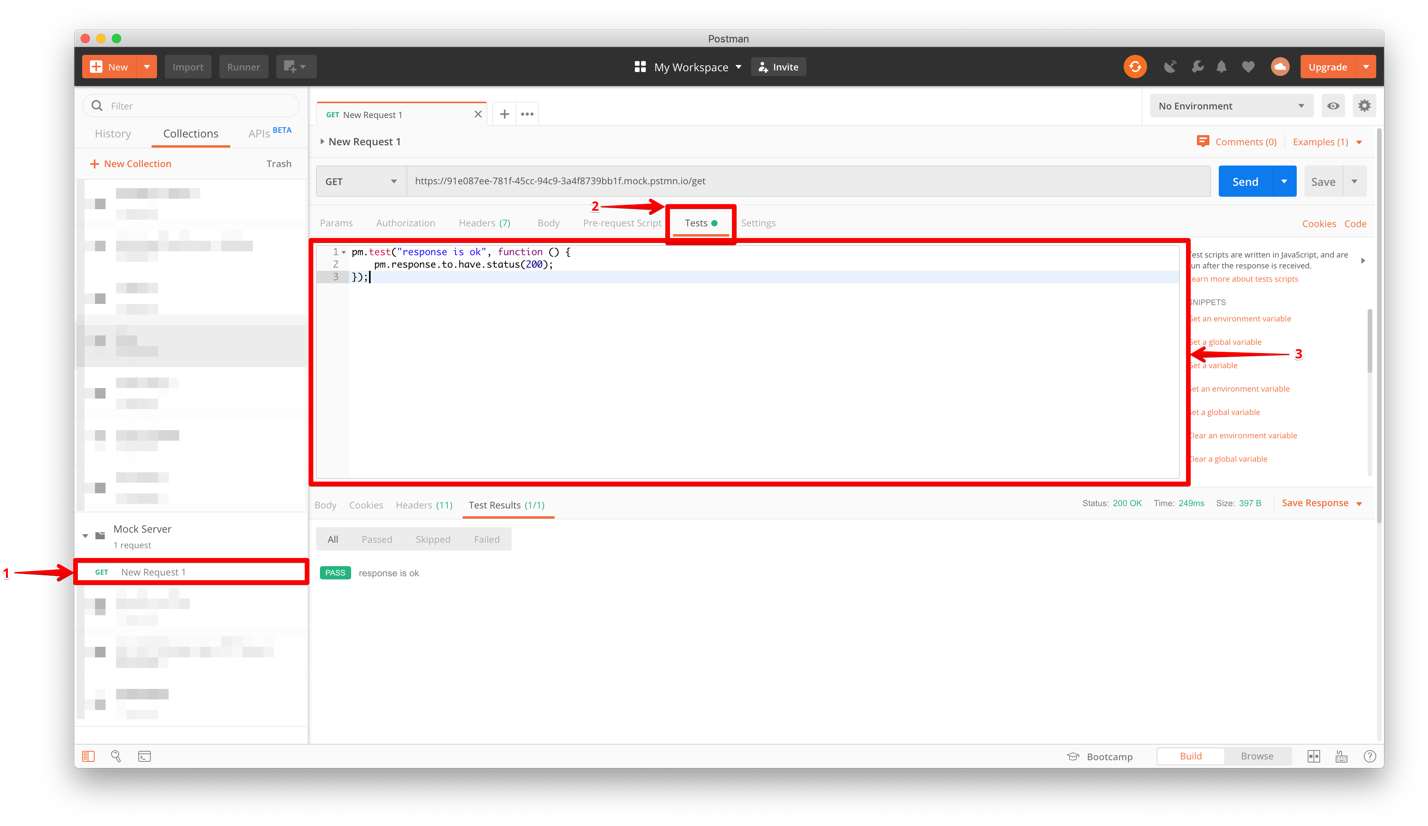
Postman Server Monitoring

Postman Server Monitoring - это интрумент 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 сделает 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 сделает аналогичные действия только запустит все запросы в коллекции, если хотя б один тест не выполнится, он сообщит нам об этом.

Добавим тест в запрос:

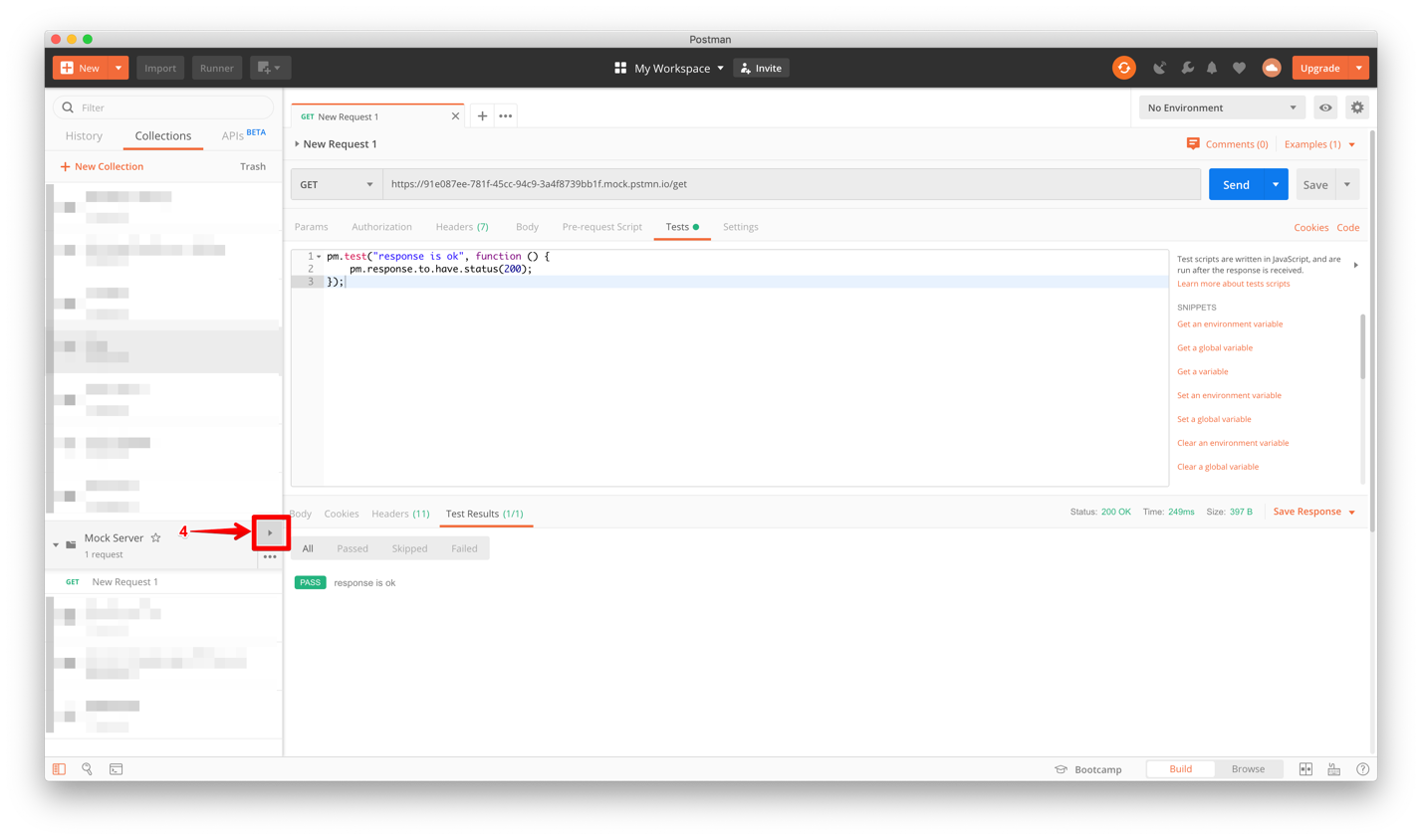
­

1. Перейдем в наш запрос
2. Откроем вкладку «Tests»
3. Добавляем тест для проверки на успешность выполненного запроса

pm.test("response is ok", function() {

pm.response.to.have.status(200);

});

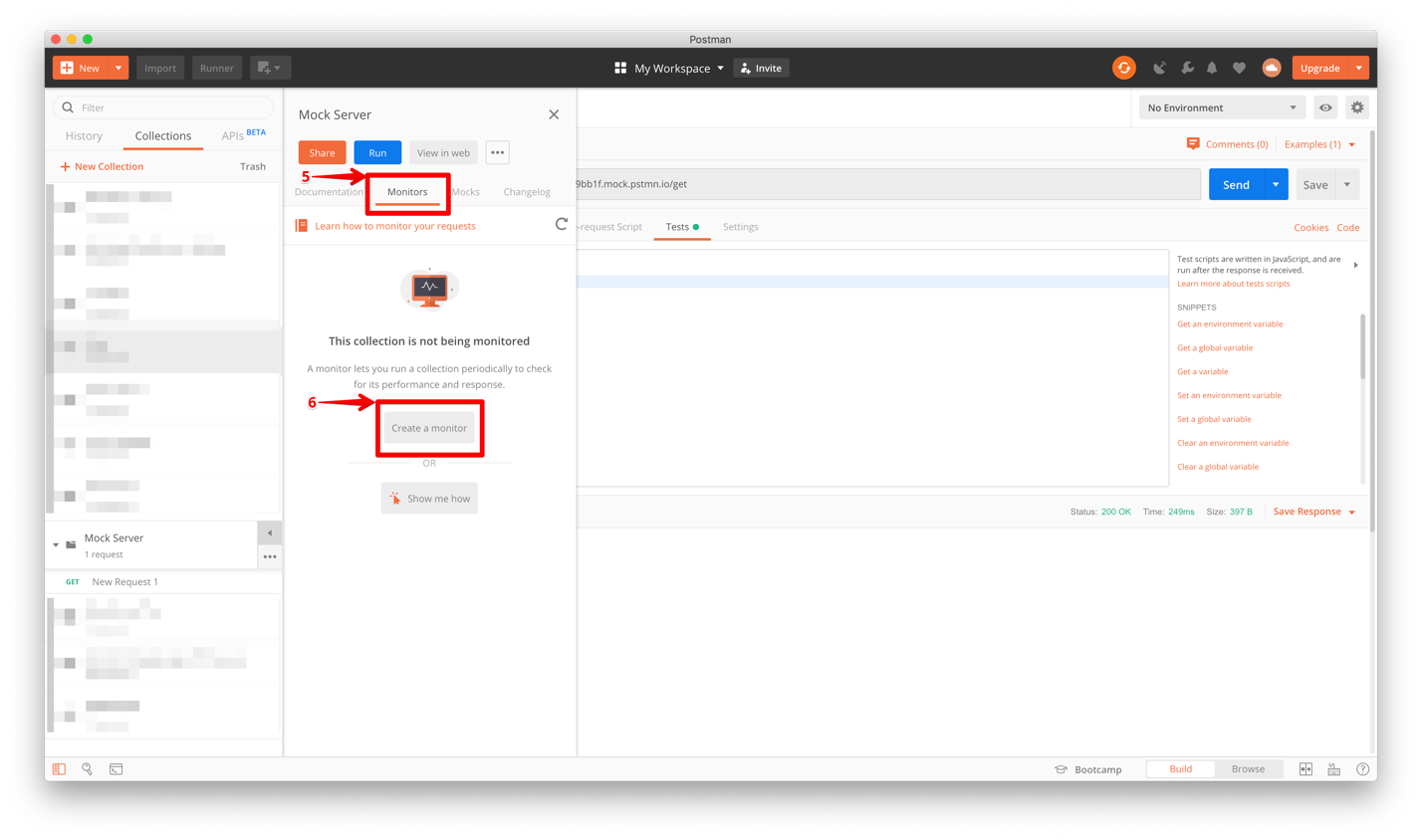


После того как у нас имеется созданный запрос и тест/ы для него:

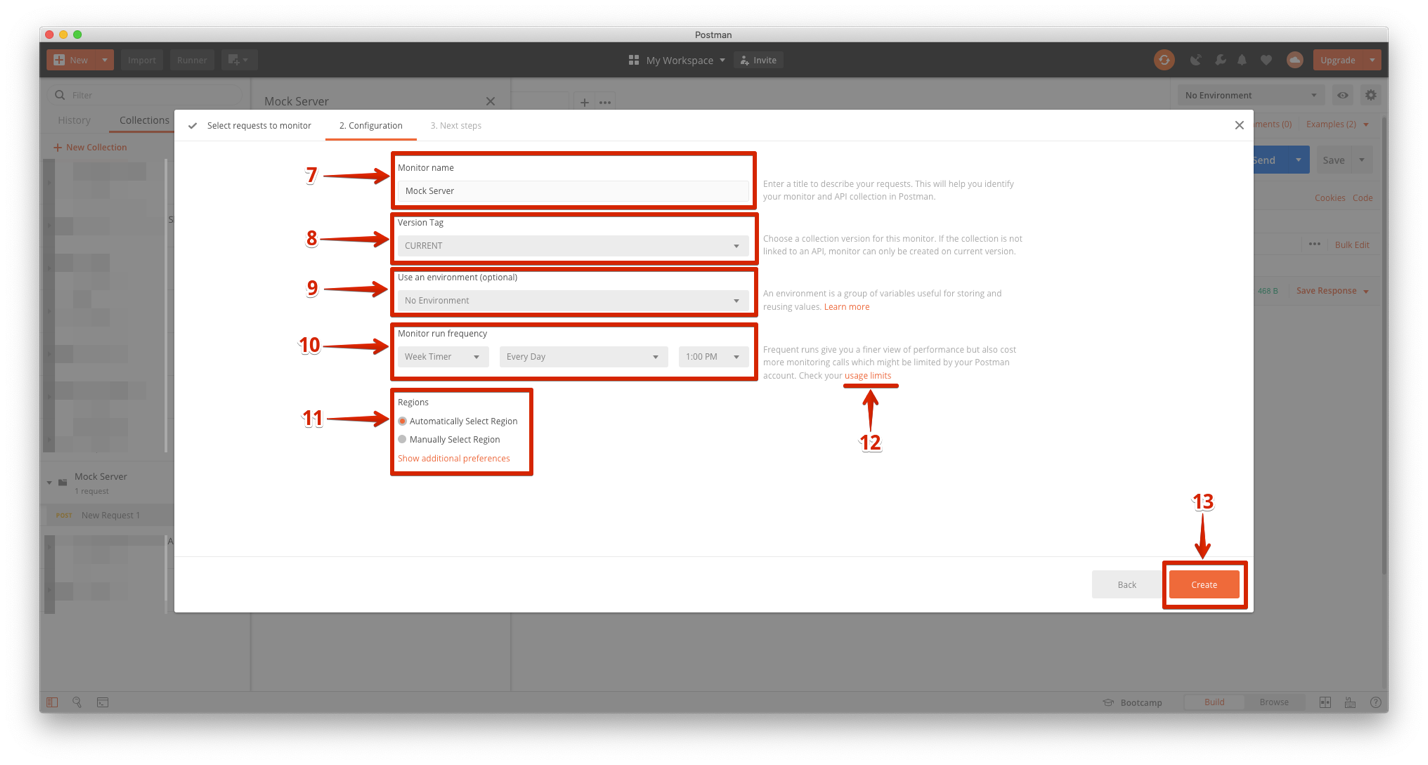
1. Разворачиваем список доступных действий для коллекции

Создадим мониторинг для нашей коллекции.

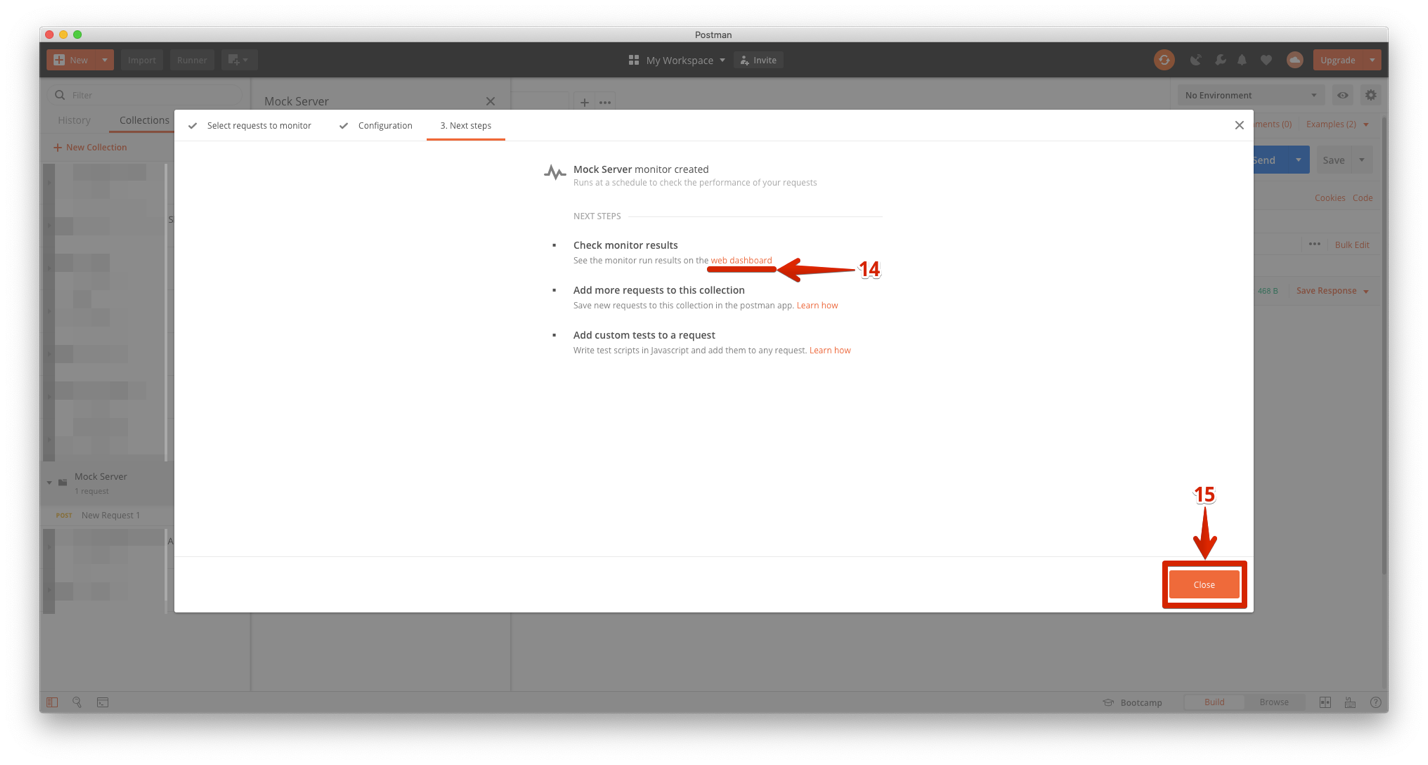
Для этого в открывшемся меню:



1. Переходим на вкладку «Monitors»
2. И жмем на кнопку «Create monitor»



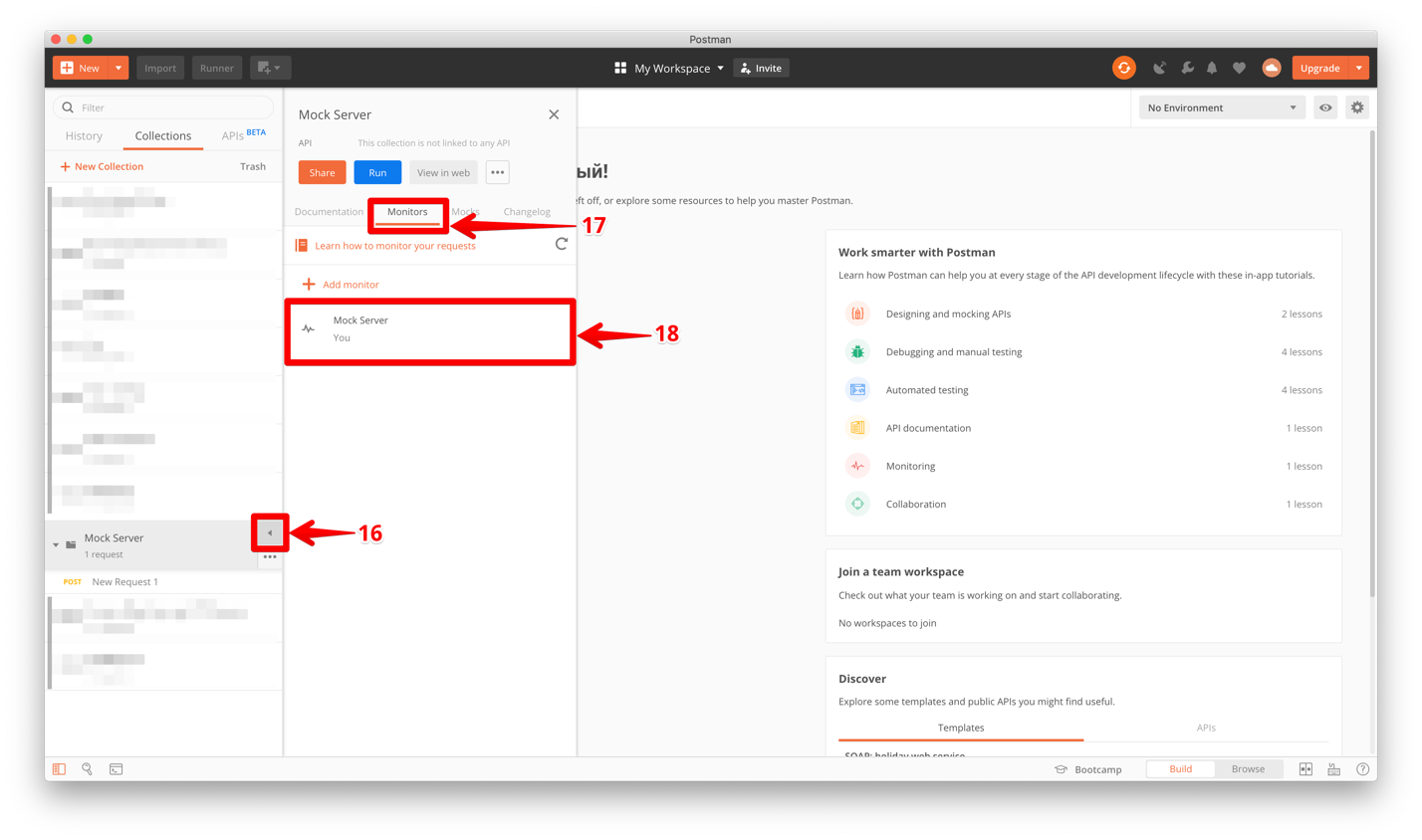
1. Вводим название для мониторинга
2. Выбираем версию (если она есть)
3. Выбираем окружение (если оно есть)
4. Задаем время или временной интервал, когда Postman должен прогнать запросы
5. Выбираем регион
6. В бесплатной версии количество проверок ограничено, по данной ссылке можно пройти и посмотреть ограничение и количество оставшихся запросов
7. Создаем сам сервер мониторинг



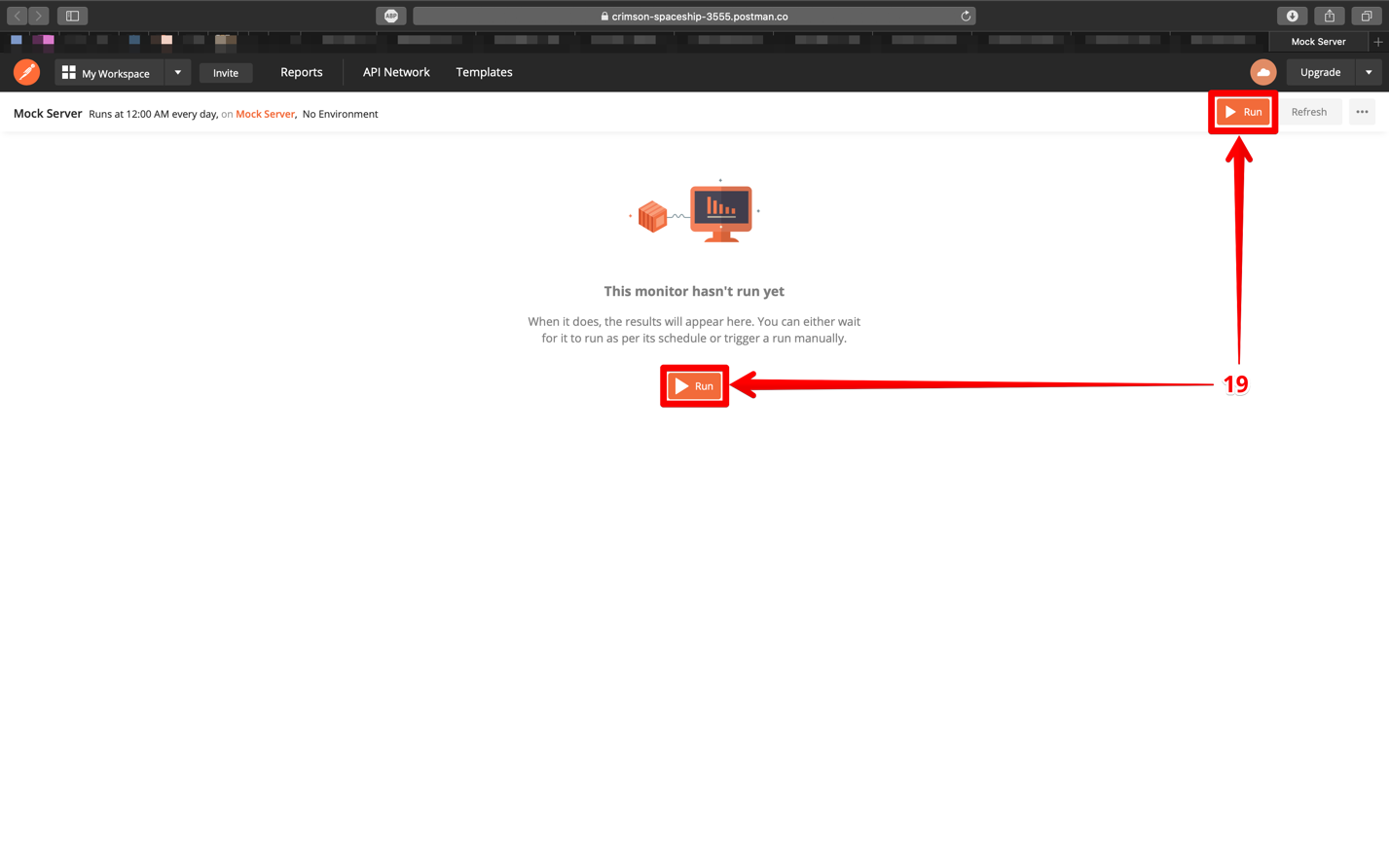
1. Ссылка для просмотра результатов
2. Закрываем окно создания мониторинга

 Мониторинг успешно создан, но работать сейчас ничего не будет. Мониторинг нужно запустить.

Запуск мониторинга, производится следующим образом:



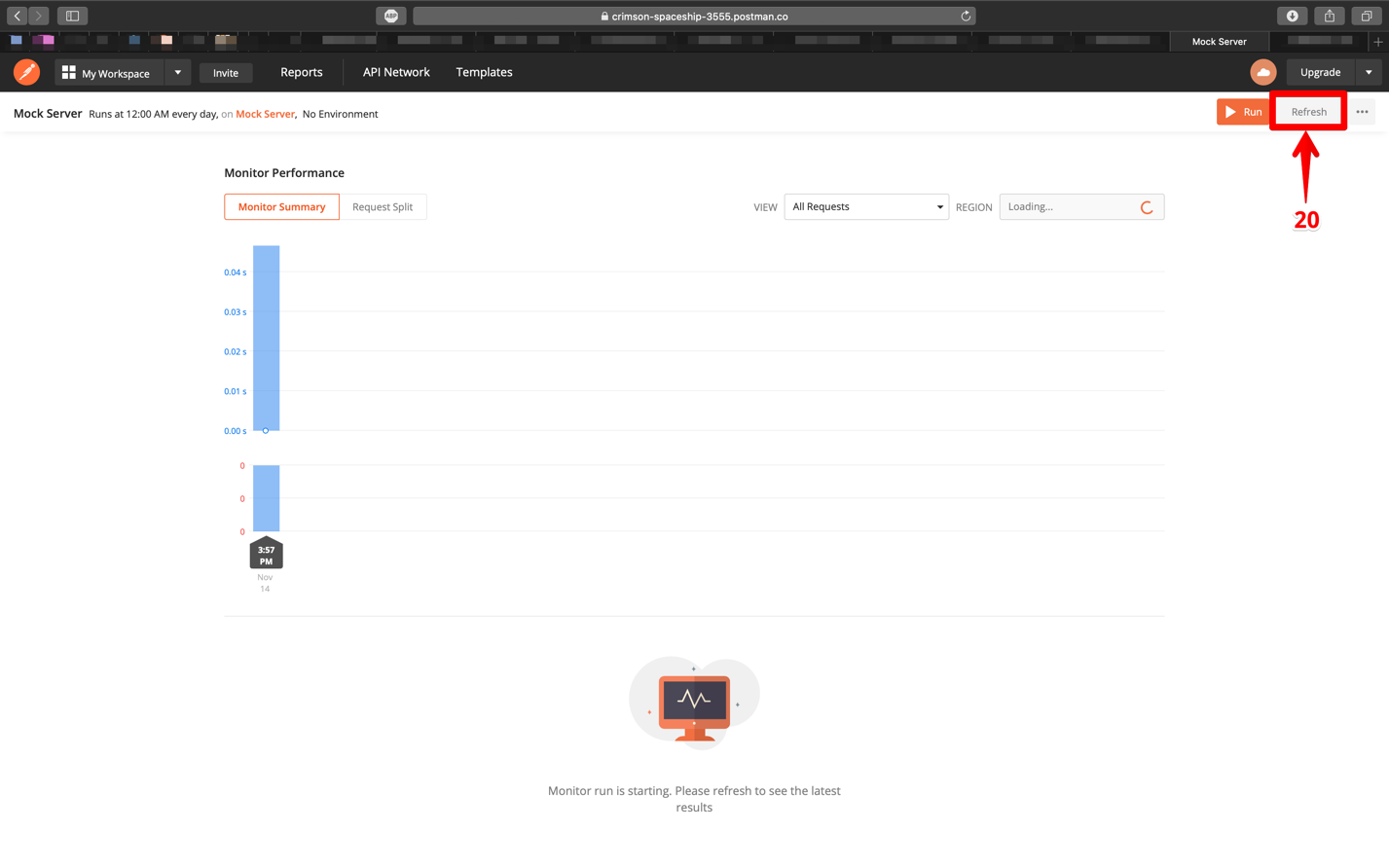
1. Открываем меню для нашей коллекции
2. Переходим на вкладку мониторинга
3. Нажимаем на созданный нами мониторинг, нас перебросит на веб версию нашего профиля



1. Для запуска мониторинга нажимаем на кнопку «Run»

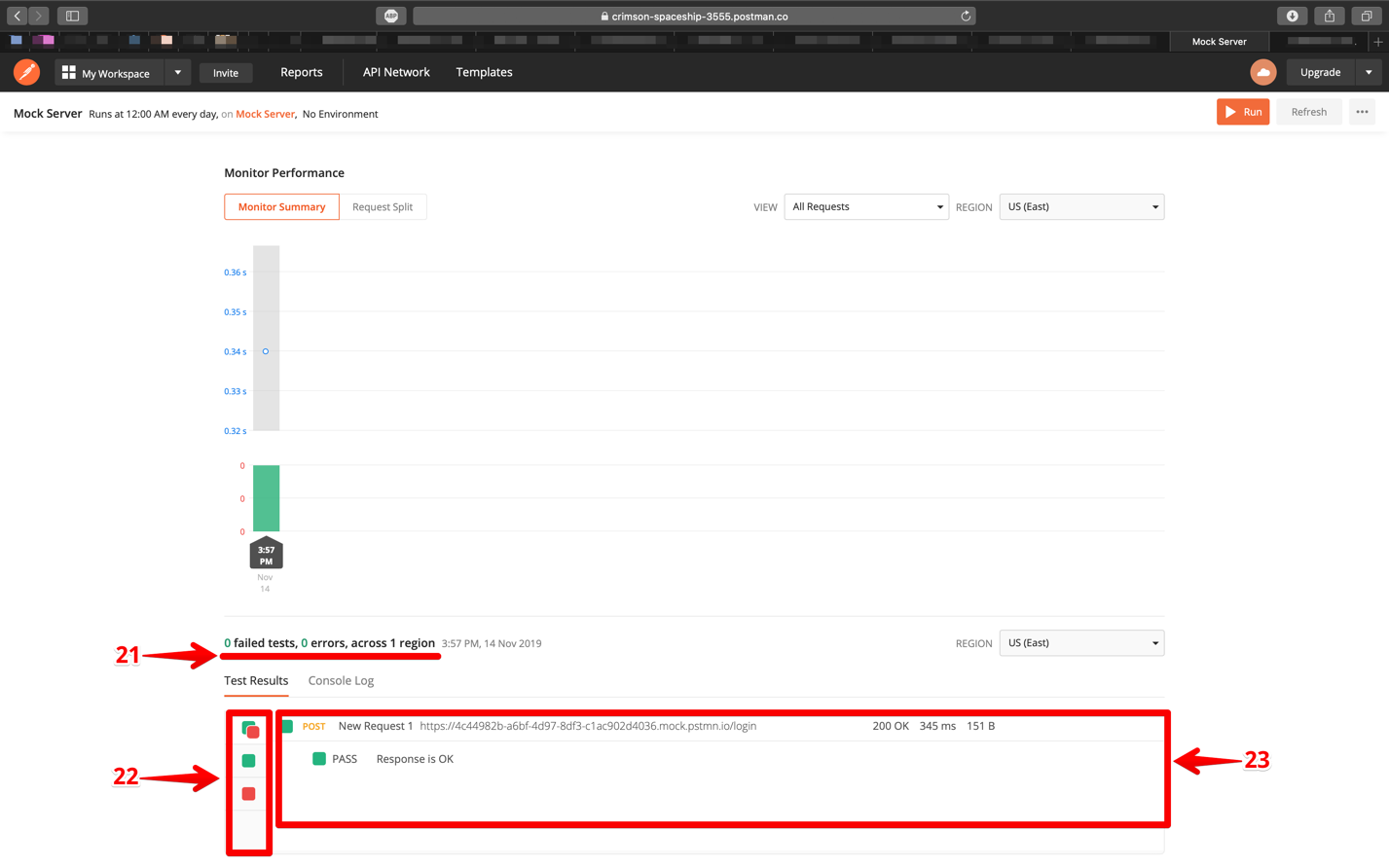
После этого мониторинг успешно запущен и будет работать пока мы его не удалим или не поставим на паузу. Также мониторинг будет отменен если у нас закончится лимит на запросы.

Просмотр результатов мониторинга, производится следующим образом:



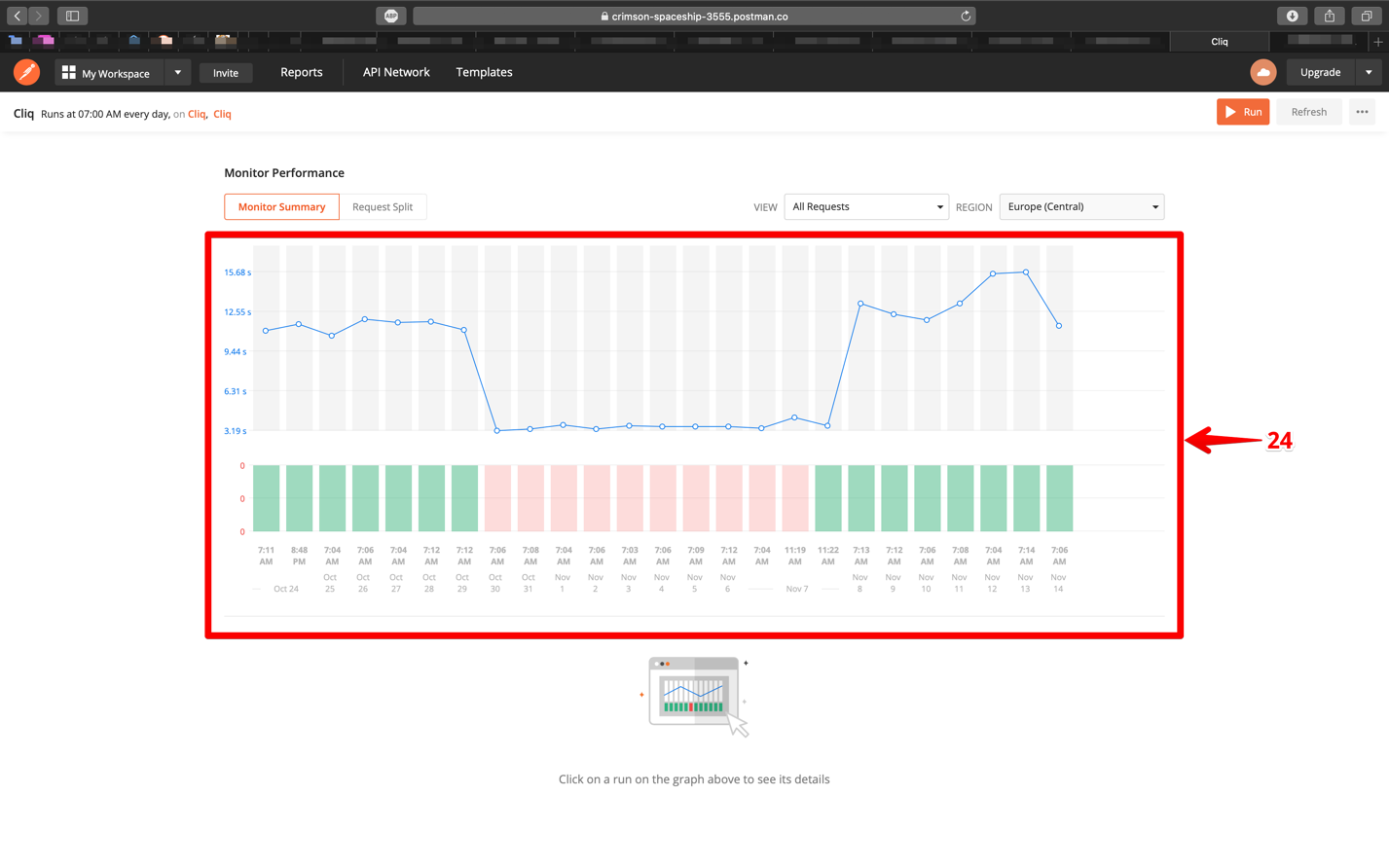
1. Для того чтобы посмотреть результаты нажимаем кнопку «Refresh» (мониторинг может занимать длительное время)

После прогона всех запросов мы увидим результат следующего вида:



1. Количество не прошедших тестов
2. Можем отсортировать для того чтобы понять какие тесты у нас упали
3. Отображаются детали мониторинга для выбранного дня

Через время наш мониторинг будет выглядеть следующим образом:



1. 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 для того 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

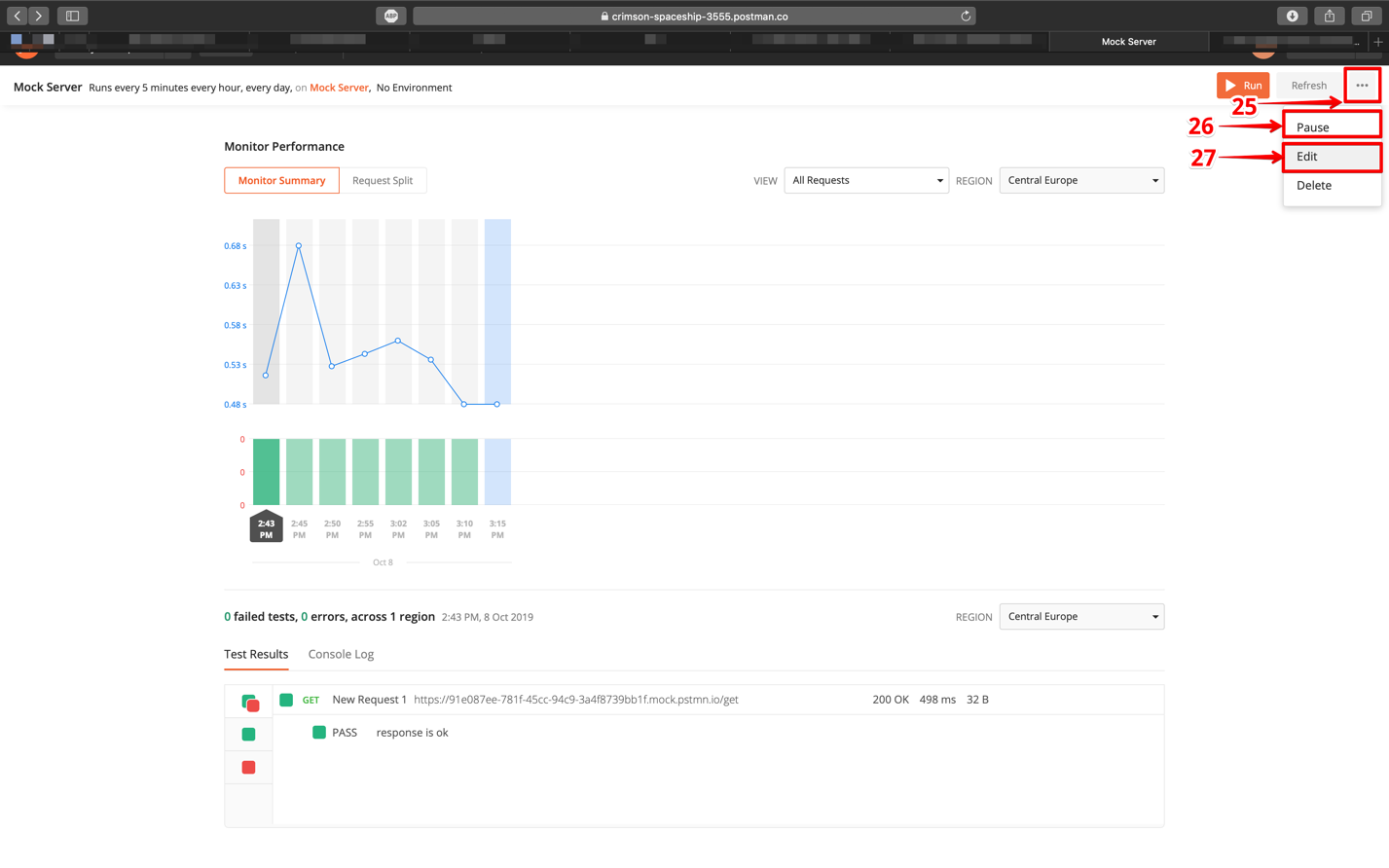
- Такой подход экономит время на поиски и устранение крэшэй и ошибок

- 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.

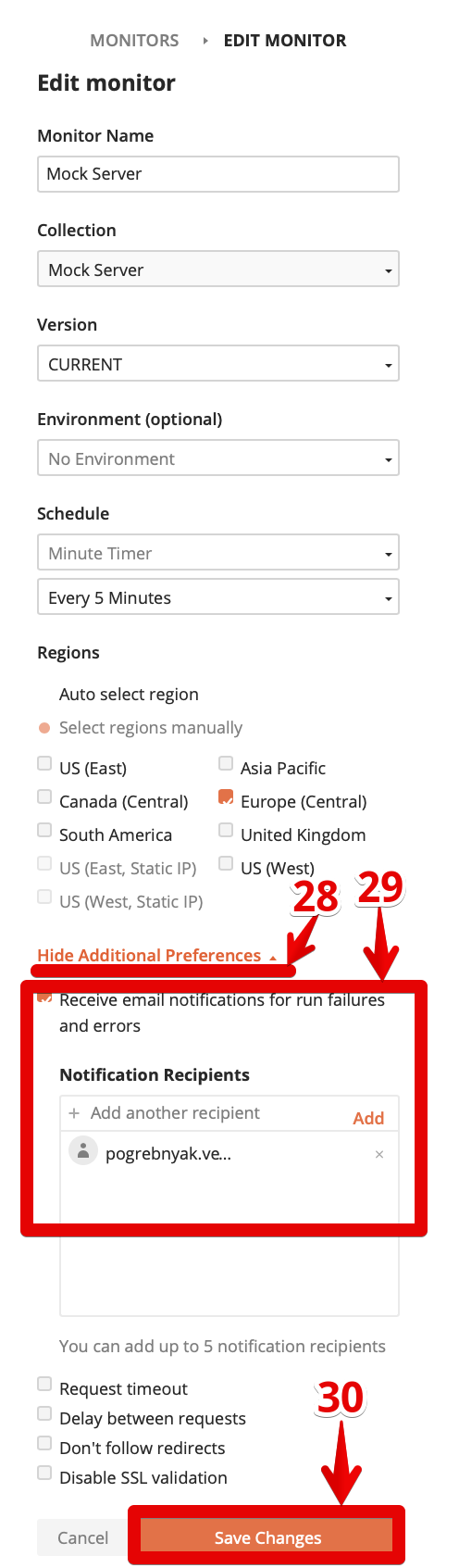
- ну и пожалуй главное – пользователи приложения всегда будут обеспечены жизненно важными функциями приложения, так как в случае не прохождения тестов мы сразу увидим это и будем принимать меры по восстановлению работоспособности нашего продукта

- довольные пользователи – хорошие рейтинги! В сторе будет меньше негативных отзывов о приложении.

Фичи которые помогут лучше жить:

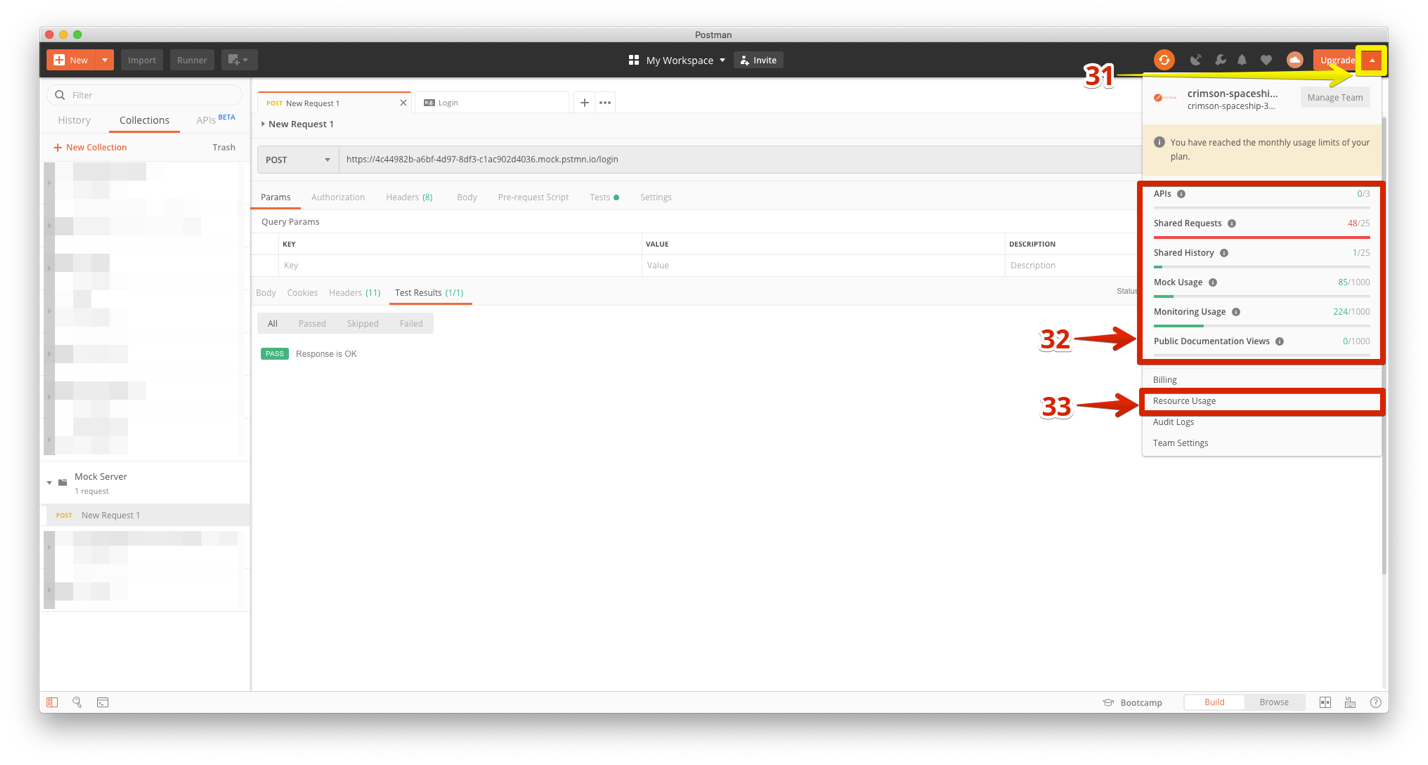
Настроить рассылку почты нужно выполнить следующие действия:

1. Развернуть меню настроек нашего мониторинга
2. При необходимости поставить мониторинг на паузу (в виду того что на бесплатном аккаунте кол-во запросов ограничено нужная штука!)
3. Нажимаем кнопку «Edit» для перехода к настройкам нашего мониторинга

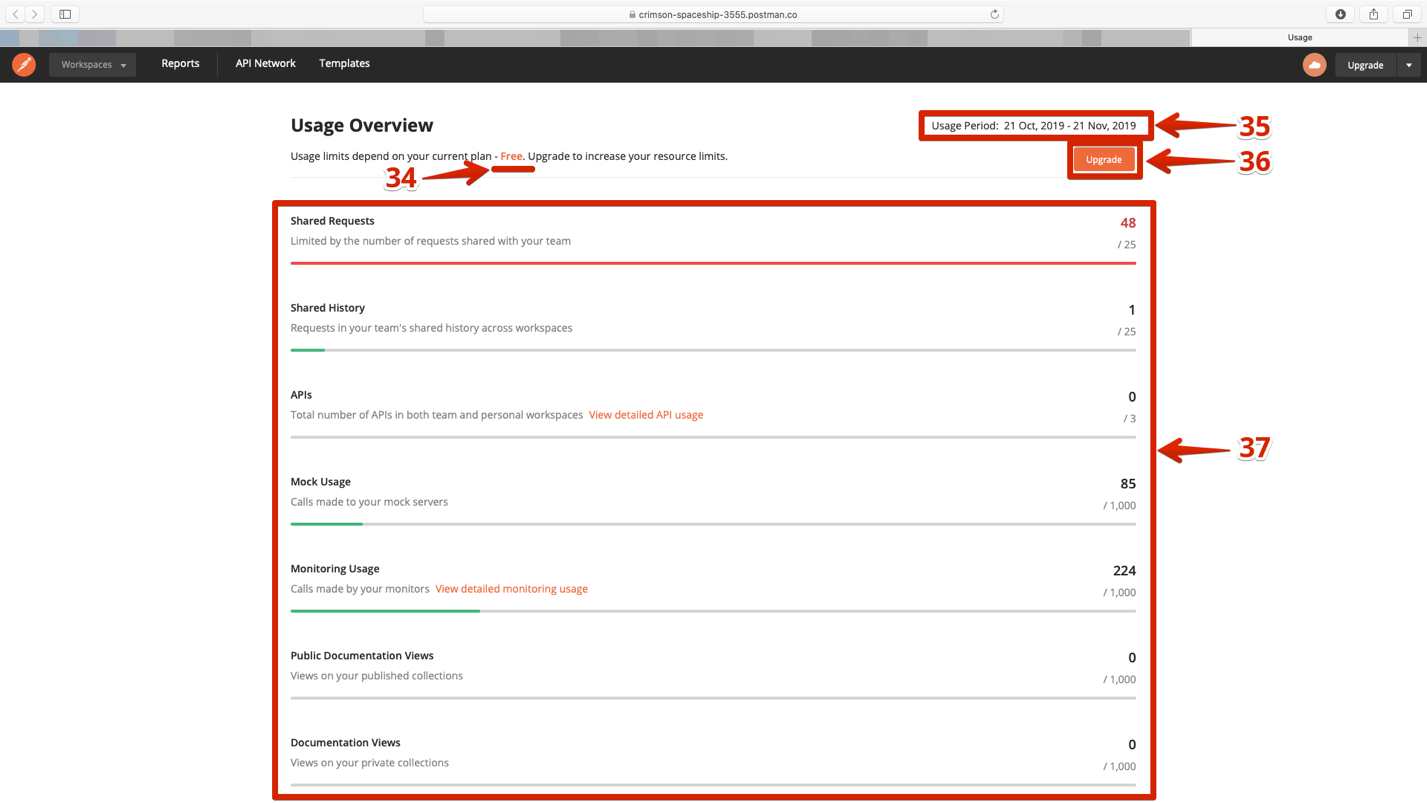


1. Разворачиваем/сворачиваем меню дополнительных настроек
2. Активируем почтовую рассылку в случае ошибки и добавляем почтовые ящики на какие должна делаться рассылка
3. Сохраняем внесенные изменения

Посмотреть количество использованных мониторинг запросов можно следующим образом:



1. Разворачиваем настройки нашего workspace
2. Отображаются данные о количестве использованных ресурсов в 30 дневный период
3. Можем перейти на сайт для смены плана или просмотра данных в веб версии



1. Тип текущего плана
2. Период использования данных (каждый месяц некоторые данные сбрасываются и доступны снова)
3. Переход на новый тарифный план
4. Таблица использованных данных