**Начинающему QA: полезные функции снифферов на примере Charles Proxy**

**Снифферы** - это инструменты, позволяющие перехватывать, анализировать и модернизировать все запросы, которые через них проходят. Они полезны, когда из потока нужно извлечь какие-либо сведения или создать нужный ответ сервера. Так можно проводить модульное тестирование продукта, в котором есть и бэк, и фронт, и разные команды со своей версионностью.

В этой статье я расскажу об основных функциях снифферов, которые могут быть полезны QA. Попробую не вдаваться в теорию, а сфокусироваться на практике. Наиболее популярными представителями анализаторов трафика сейчас являются WhireShark, Fiddler и Charles Proxy. Об удобстве интерфейсов и функционале каждого из них можно рассуждать долго, учитывая все плюсы и минусы. Но здесь я отдал предпочтение Charles, поскольку сам им активно пользуюсь. Буду рассказывать на его примере.



Что собой представляет Charles Proxy

Charles Web Debugging Proxy - это инструмент мониторинга HTTP и HTTPS трафика. Он выступает в роли прокси-сервера (промежуточного звена) между тестируемым приложением и сервером на бэкенде, позволяя не только видеть, но также перехватывать и редактировать запросы.

Главное преимущество Charles Proxy и снифферов в целом - возможность просмотра трафика, в том числе с мобильных устройств, что значительно облегчает работу тестировщика клиент-серверных мобильных приложений.

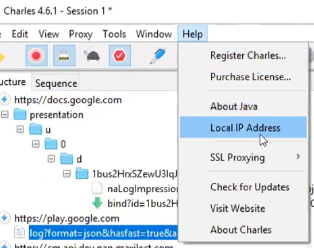
**Первичная настройка**

При тестировании мобильного приложения

Charles Proxy необходимо запустить на компьютере, который находится в той же локальной подсети, что и мобильное устройство с тестируемым приложением.

Как правило, соединение настраивается по Wi-Fi. В настройках Wi-Fi мобильного устройства в качестве proxy-сервера надо указать IP-адрес компьютера и стандартный порт инструмента 8888 (пароль остается пустым).

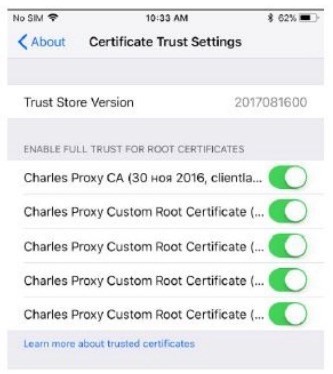
IP-адрес компьютера можно узнать через командную строку (ipconfig) или в самом Charles Proxy (Help -> Local IP Address).



Этот же адрес есть в инструкции по подключению, доступной в Help -> SSL Proxying -> Install Charles Root Certificate on mobile device remote browser.

После сохранения настроек Charles Proxy сможет читать HTTP-трафик мобильного устройства. Но чтобы смотреть расшифрованный трафик HTTPS, нужны дополнительные манипуляции - требуется установить SSL-сертификат Charles Proxy в браузере на мобильном устройстве.

Скачать сертификат можно по адресу: [chls.pro/ssl](http://chls.pro/ssl) (адрес, по которому скачивается сертификат, также можно найти в инструкции Help -> SSL Proxying -> Install Charles Root Certificate on mobile device remote browser). Далее в iOS его необходимо сделать доверенным (в Настройки -> Основные -> Профили).

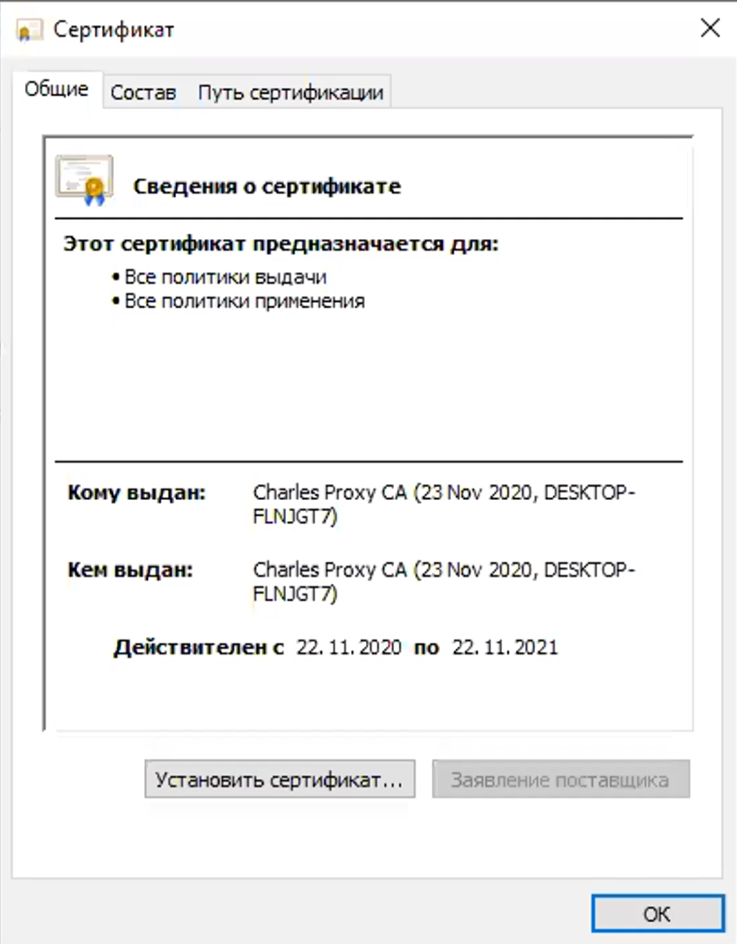


В Android установленные сертификаты верифицируются в Settings -> Trusted Credentials на вкладке User.

Главное отличие в настройках для этой ОС в том, что с Android 6.0 и выше в Androidmanifest надо добавить специальную конфигурацию, позволяющую просматривать защищенный трафик. На продакшене эта конфигурация убирается из соображений безопасности.

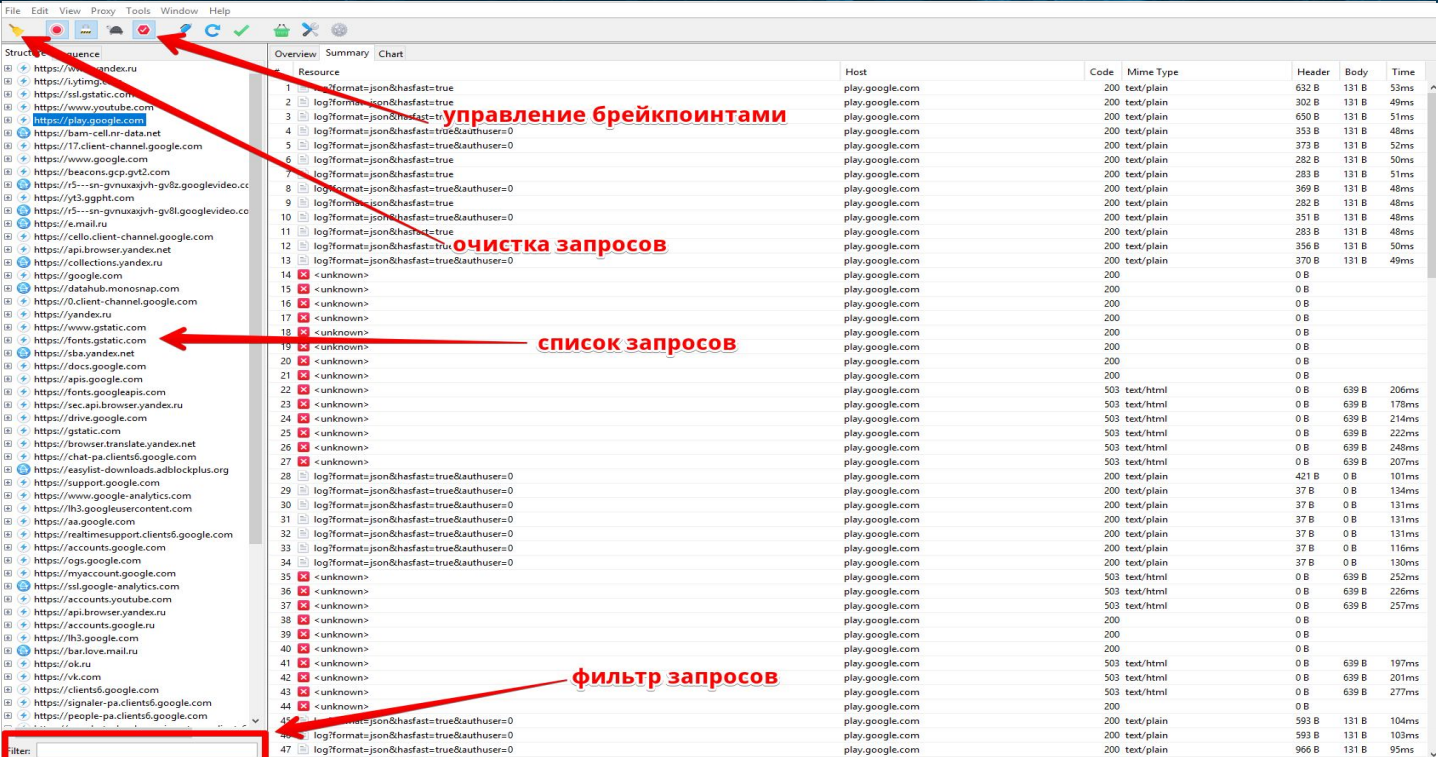
**При тестировании приложения на ПК**

В этом случае дополнительные сертификаты нужно установить на сам ПК. Для скачивания и установки нужна ссылка из Help -> SSL Proxying -> Install Charles Root Certificate.



Сертификат устанавливается в доверенные корневые центры.

Два слова об интерфейсе



Интерфейс Charles Proxy прост. Слева - список перехваченных запросов, справа - детали.

В списке запросов есть две основные вкладки - **Structure и Sequence**.

В первом случае запросы рассортированы по хостам-папкам. Наведя на любой из них, можно получить всю информацию о количестве запросов к этому корневому хосту, доле удачных, таймингах, размерах и т.п. Фактически, здесь представлена вся та же информация, которую можно получить из панели разработчика в браузере. Выбрав конкретный URL, можно увидеть код ответа, версии протоколов, контент и т.п. Тело запроса, заголовки, cookie (если есть) можно посмотреть в разных форматах - даже в HEX.

С помощью контекстного меню запроса можно настраивать блокировки, повторять и изменять запросы. На этом мы еще остановимся.

На вкладке Sequence запросы выведены по времени в виде настраиваемой таблицы. Видно, когда начался запрос, сколько он длился, его размер, статус и т.п. Наведя на конкретную строку, мы получим ту же информацию о теле, заголовках и т.п.

Если запросов на экране слишком много, с помощью панели инструментов их можно очистить или вообще остановить перехват. Там же есть возможность включить и выключить троттлинг (подробнее об этих функциях расскажу позже). Базовая настройка каждой из функций осуществляется через меню, а кнопки панели управления выступают своего рода тумблерами On / Off.

**Фильтрация**

В Charles Proxy очень много вариантов фильтрации запросов.

Начнем с вкладки Structure. Самое примитивное - скопировать хост и вставить в поле Filter. Так мы увидим только запросы с этого хоста. Примерно того же результата можно добиться, если в контекстном меню хоста выбрать Focus. Остальные запросы будут собраны в Other Hosts. Если при этом перейти на закладку Sequence и отметить настройку Focused, то в списке окажется информация только о тех запросах, которые были выбраны на вкладке Structure.

На вкладке Sequence есть аналогичный фильтр.

Charles Proxy умеет работать с регулярными выражениями. Для этого на вкладке Sequence выбираем Settings и отмечаем пункт Filter uses regex. И вписываем в поле поиска элементарную регулярку.

Например, вот так

^\w{4}\.

можно выбрать все запросы, в которых в начале имени хоста находится четыре буквы, а потом идет точка.

Там же можно включить Autoscroll списка запросов или указать максимальное количество строк.

В Charles Proxy можно фильтровать не только отображение, но и запись запросов. Для этого надо зайти в Proxy -> Record settings и задать условия на одной из вкладок - Include или Exclude - так мы включаем или выключаем запись запросов данного хоста.

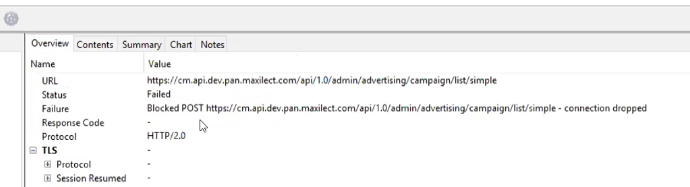
Похожего результата можно добиться, используя блок-листы. Включить хост в блок лист можно из контекстного меню (команда Block list) или через добавление в Tools -> Block list, где следует отметить Enable Block list.

Блокируемый запрос можно прервать двумя способами (подходящий надо выбрать в настройках):

* сбросить соединение;
* вернуть ошибку 403.

От выбранного способа блокировки зависит, какой ответ мы получим.

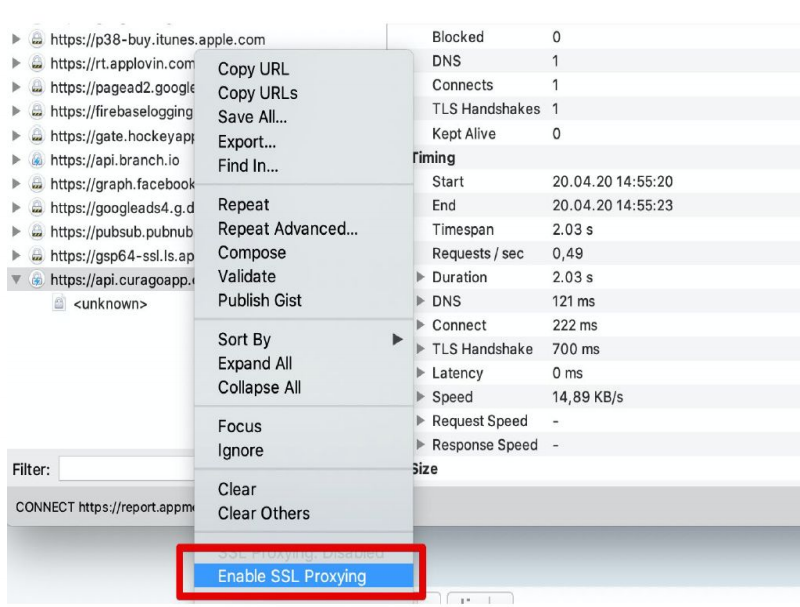
Можно провести эксперимент. Включим наш тестовый стенд в Block list, выбрав простой сброс соединения. С помощью контекстного меню запроса повторим его (команда Repeat) и получим такой ответ:



Статус запроса - Failed, а в описании ошибки указано, что Connection dropped.

Просмотр SSL-трафика

Если ранее мы успешно установили SSL-сертификат, для просмотра зашифрованного трафика остается только включить SSL proxying для нужного хоста в самом Charles Proxy. Это можно сделать через контекстное меню конкретного хоста.



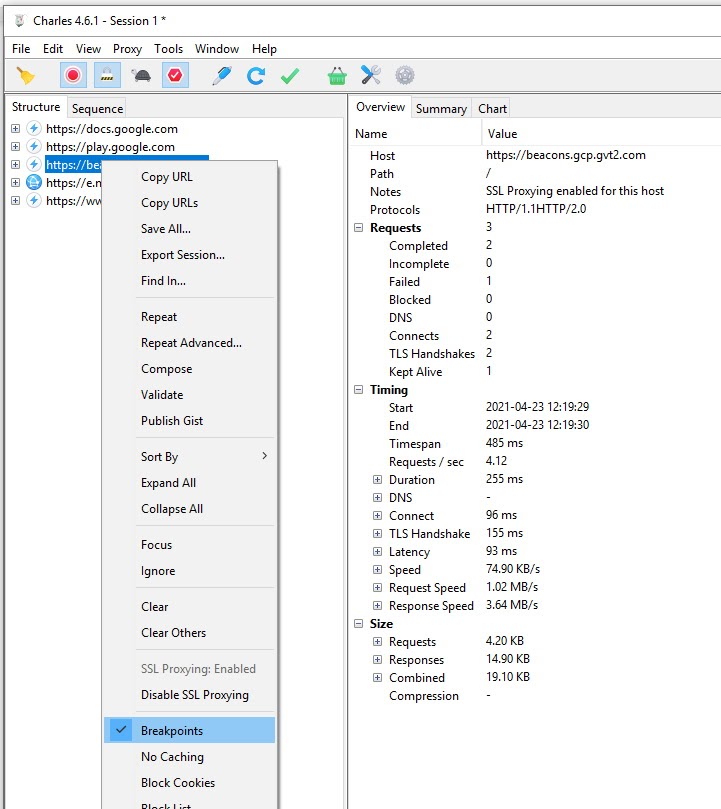
Чтобы не включать каждый хост, можно зайти в Proxy -> SSL Proxying settings и на первой вкладке SSL Proxying включить Enable SSL Proxying.

Аналогично настройке фильтров на вкладках Include и Exclude можно добавить или исключить конкретные хосты. Если списки на этих вкладках не заполнять, по умолчанию мы будем читать трафик со всех хостов.

**Брейкпоинты**

Наверное, одна из самых популярных функций Charles Proxy - это настройка точек остановки, которые позволяют перехватывать запросы.

Установить Breakpoint можно из контекстного меню запроса. После этого все аналогичные запросы будут перехвачены. Их можно будет просматривать и редактировать.



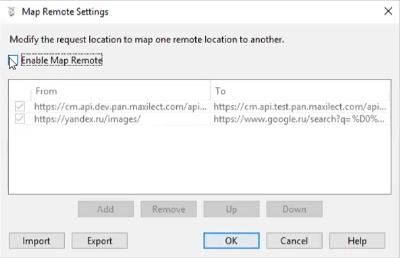
Чтобы проверить, как это работает можно использовать повтор запроса (Repeat из того же контекстного меню). Запрос перехватывается, его можно редактировать.

В принципе, изменить можно все - от header до авторизационного токена. Когда редактирование будет закончено, можно выбрать Execute и в Charles Proxy появится повторный запрос, который и отправится на сервер, а потом вернется с ответом. В этот момент можно будет посмотреть и отредактировать ответ, который получит приложение - появится поле Edit response.

Редактируя запрос, можно ввести заведомо некорректные данные и посмотреть, как ответит сервер. Также можно отредактировать ответ (внеся некорректные данные) и использовать его для тестирования фронта. Можно оставить корректные данные, но изменить код - посмотреть, как фронт воспринимает информацию, переданную через API.

**Map remote**

Еще одна популярная функция Charles Proxy - подмена ответа сервера. Так мы можем ответ одного хоста подменить на ответ другого. Настраивается это через Tools -> Map Remote.



Обратите внимание, в левом нижнем углу есть кнопки Import и Export. Они позволяют обмениваться настройками - переносить их с одного рабочего места на другое.

Например, мы можем подменить контура. Я буду посылать запрос на dev-контур, но ответ хочу получить с тестового стенда. Для этого создаем новый пункт в списке Map Remote Settings. Map From - куда изначально был запрос; Map to - откуда берем ответ.

Эта функция может быть полезна, если мы поймали ошибку на продакшене, например при загрузке таблицы, а информации в dev-контуре для полноценного тестирования не хватает. Вместо того, чтобы перезаполнять информацию в каждом контуре, мы можем скопировать ответ сервера в нужный контур и посмотреть, как он ведет себя после исправления ошибки.

**Map Local**

Главное отличие Map Local от предыдущей функции в том, что замена осуществляется не на ответ другого сервера, а на содержимое локального файла. Настройки выглядят точно также, но вместо второго сервера мы указываем локальный путь к файлу с ответом.

**Rewrite**

Функция Rewrite может быть полезна, если вам нужно переписать данные, которые отправляются в Charles Proxy. В отличие от простого редактирования Rewrite позволяет задать правила изменения и работать в автоматическом режиме. Можно изменять и добавлять заголовки, искать и заменять текст в теле запроса или ответа. Можно даже менять статус ответа.

Rewrite удобно использовать, когда нужен готовый ответ, но мы не хотим каждый раз ставить брейкпоинт и вписывать его в ручную. Редактируя таким образом ответ, фактически, мы ставим заглушку - можем имитировать работу сторонних партнеров. Честно говоря, в Postman-е эта функция реализована более интуитивно. Но раз мы говорим про Charles, не будем отходить от его примера.

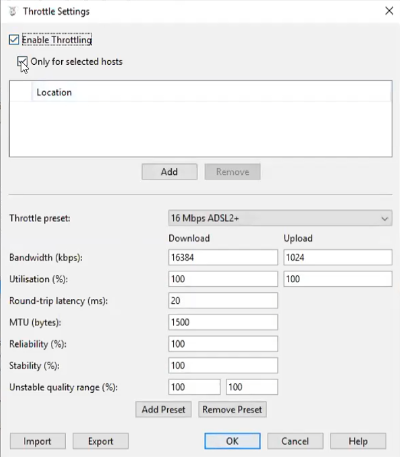
Настроить это можно через Rewrite settings, доступные в контекстном меню. Единственный недостаток инструмента в том, что каждое правило замены прописывается отдельно.

Помимо изменения запросов и ответов мы можем запретить кэширование или cookie (функции No caching и Block cookies). Эти опции повторяют аналогичные инструменты панели разработчика в браузере. В обоих случаях настраивается список хостов, для которых действует настройка. Если же список пуст, то кэширование и cookie отключаются на всех перехваченных хостах.

**Throttling**

Charles Proxy помогает тестировать сервис на плохой связи, искусственно ограничив через настройки пропускную способность канала. Эта функция полезнее всего для тестирования десктопных приложений, поскольку на мобильных устройствах качеством связи можно управлять через панель разработчика.

Настраивается функция через Proxy ->Throttling settings.



Ограничения можно включить либо для хостов из списка, либо для всех перехваченных запросов. Из коробки есть несколько пресетов, но все параметры связи можно корректировать вручную. Мне по большей части хватает базовых настроек.

**Repeat Advanced**

Хотя полноценное нагрузочное тестирование лучше проводить в специальных инструментах, Charles Proxy имеет одну базовую настройку, которая помогает закрыть минимальные потребности. Функция Repeat Advanced (доступная через контекстное меню перехваченного запроса) позволяет нужное количество раз повторить тот же запрос. После настройки откроется отдельная сессия, где будут видны детали каждого из запросов.

Конечно, список функций Charles Proxy этим не ограничивается. Есть еще много полезного - от перенаправления доменного имени на другой IP-адрес, до автоматического сохранения полученных ответов.

Отмечу, что Charles Proxy платный. Можно использовать триальную версию. Но раз в 5-7 минут поверх него будет отображаться всплывающее окно с версией, а раз в 30 минут он будет выключаться, при этом сессии не сохраняются. Решайте сами, помешает ли это вашей работе.

В целом Charles Proxy - это хороший инструмент. Не сказал бы, что он сильно отличается от того же Fiddler-а, но на мой взгляд он больше заточен под MacOS, поскольку пришел на эту платформу раньше. Хотя сейчас уже нельзя сказать, что какой-то сниффер принципиально лучше или хуже. Все они решают примерно одни и те же задачи сходным образом.