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.

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).

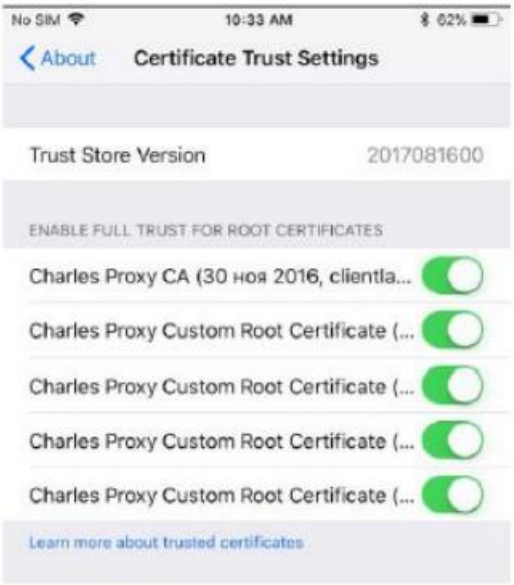
IP-адрес компьютера можно узнать через командную строку (ipconfig) или в самом Charles Proxy (Help -> Local IP Address).

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

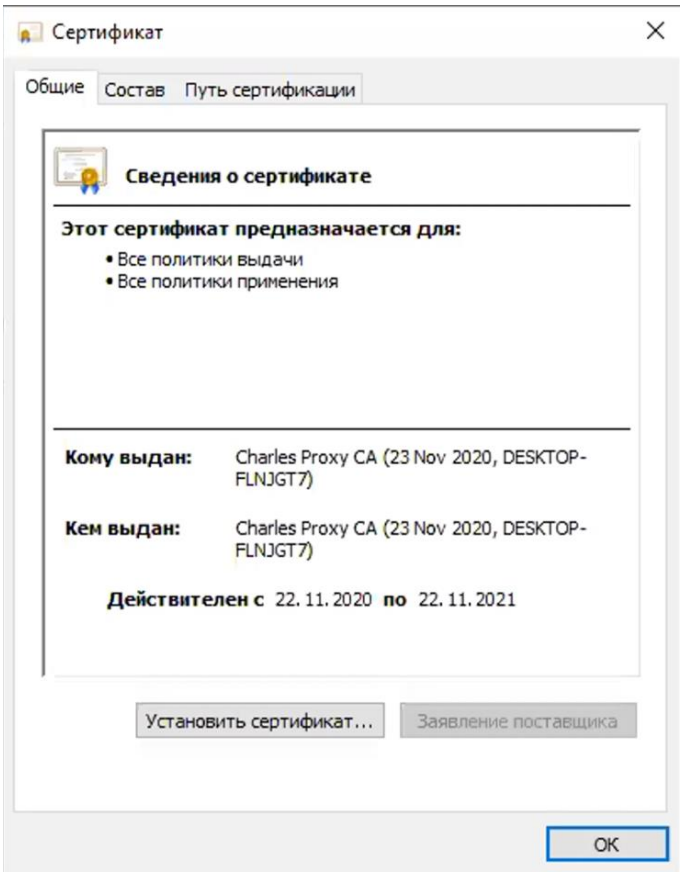
манипуляции - требуется установить SSL-сертификат Charles Proxy в браузере на мобильном устройстве.

Скачать сертификат можно по адресу: chls.pro/ssl (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.

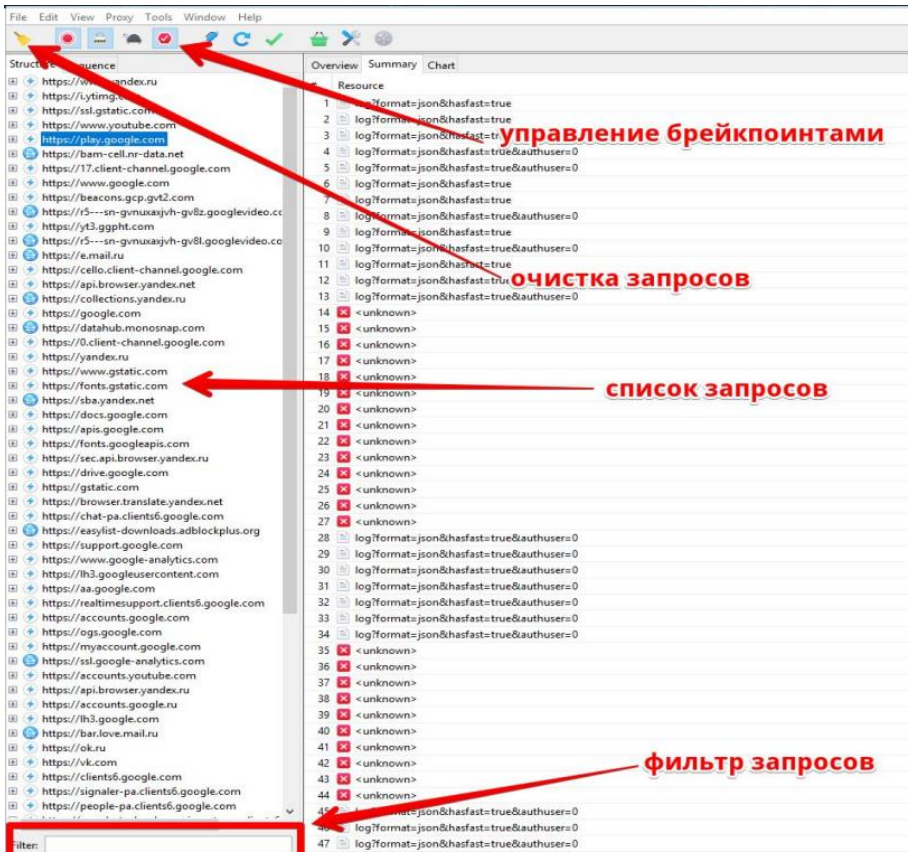
Главное отличие в настройках для этой ОС в том, что с Android 6.0 и выше в Androidmanifest надо добавить специальную конфигурацию, позволяющую просматривать защищенный трафик. На продакшене эта конфигурация убирается из соображений безопасности.



При тестировании приложения на ПК 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 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). 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.

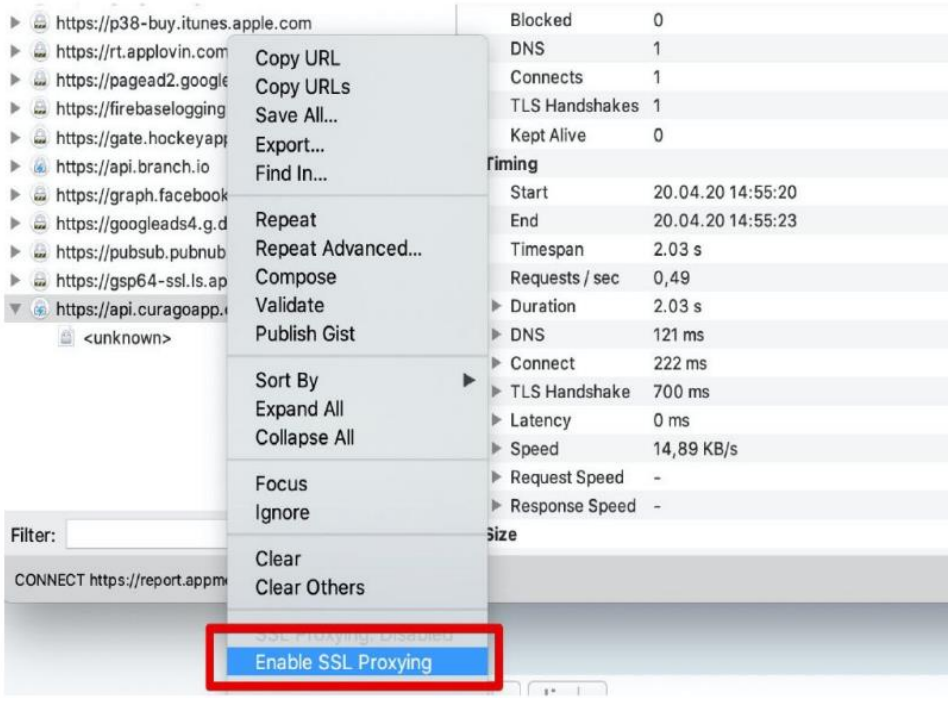
Например, вот так

`^\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.

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

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.

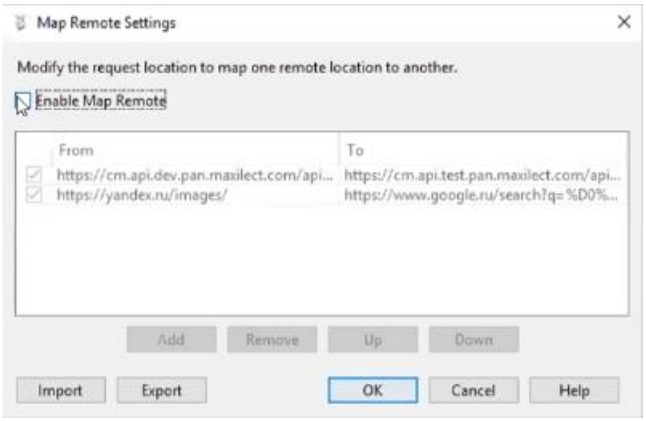
Чтобы проверить, как это работает можно использовать повтор запроса (Repeat из того же контекстного меню). Запрос перехватывается, его можно редактировать.

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.

Перейдите в Tools → Map Local. Далее в окошке «Map Local Settings» нажмите Add → Хост: https://hostname → Local path: путь на компьютере до файла. Можно использовать готовые медиа-файлы, HTML, CSS, JSON, XML. Больше подходит, конечно, разработчикам, чтобы не загружать данные на сервер для его последующего тестирования, но и тестировщик может найти грамотное применение.

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.

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

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.

Proxy → Throttle Settings → галочка **Enable Throttling**. Если не разбираетесь во всех перечисленных пунктах, то можете использовать Throttle preset и там выбрать подходящую для теста скорость, а система автоматически заполнит остальные поля.

Если выбрать «Only for selected hosts», то можно задать определенный хост, к которому будут применяться ваши настройки. Здесь можно использовать готовые пресеты с настройками для различных типов (4G, 3G и т. д.). А также можно задать различные параметры, коротко о некоторых из них:

Bandwidth — максимальный объем данных, который может быть передан с течением времени.
Utilisation — доля общей пропускной способности, которая может быть предоставлена пользователю в любой момент времени.
Latency — задержка в миллисекундах по запросу firts между клиентом и удаленным сервером.
MTU — максимальное передающее устройство для текущего пресета.
Reliability — мера вероятности, что соединение не удастся. Используется для имитации ненадежных сетевых условий.
Stability — мера вероятности, что соединение будет нестабильным и, следовательно,

снизится качество. Полезно для моделирования сетей, в которых периодических падает качество связи, например, мобильных.

Reverse Proxies

Reverse proxy — обратный прокси-сервер. Обычно используется для того, чтобы принимать запросы из Интернета и перенаправлять их на один из веб-серверов.

Port Forwarding

Port Forwarding — проброс портов, который иногда называют перенаправлением портов, или туннелированием — процесс пересылки трафика, адресованного конкретному сетевому порту, с одного сетевого узла на другой. Этот метод позволяет внешнему пользователю достичь порта внутри локальной сети.

Tools.

Инструмент **No Caching** предотвращает кэширование, манипулируя заголовками HTTP, которые управляют кэшированием ответов.

Block Cookies — заголовок файла Cookie удаляется из запросов, предотвращая отправку значений файла из клиентского приложения (например веб-браузер) на удаленный сервер. А также из ответов удаляется заголовок Set-Cookie, предотвращая получение клиентским приложением запросов на установку файлов cookie с удаленного сервера. В настройках можно включить удаление Cookie как для всех хостов, так и для выбранных.

Block List — позволяет блокировать определённые доменные имена. Когда веб-браузер попытается запросить любую страницу из заблокированного доменного имени, она заблокируется. Можно выбрать либо «Drop connection», либо возврат 403 ошибки.

DNS Spoofing

Виртуальный хостинг — это когда у вас есть несколько сайтов на одном IP-адресе, и веб-сервер определяет, какой сайт вы запрашиваете, основываясь на имени, введённом в браузере. Точнее, сервер смотрит на заголовок хоста, отправленный в запросе. Например, когда нужно подменить хосты, чтобы при вводе какого-либо адреса в браузере запросы уходили по другому адресу (допустим, на тестовую площадку).

Mirror — эта функция позволяет автоматически сохранять все ответы, возвращаемые в Charles Proxy. Они раскладываются локально в такой же иерархии, как на сервере. Если внезапно случился даунтайм на бэкенде, отвалилась тестовая среда и т. д., у вас уже есть готовые моки для Map Local.

Активировать функцию можно так: **Tools** → **Mirror** или **Tools** → **Auto Save**.

Compose — функция редактирования запросов, которые вы поймали. Например вы добавляете в избранное какой-то товар, но почему-то он не добавляется. Вы можете отредактировать уже отправленный запрос и отправить его еще раз. Для этого необходимо выбрать нужный запрос из списка, нажать на нём правой кнопкой и выбрать **Compose**. Иконка у запроса поменяется, и теперь можно смело его редактировать. После того, как вы изменили нужные значения в запросе, нажмите внизу

«Execute», чтобы отправить запрос на сервер.

Recording Settings — настройки отображения списков разрешенных и запрещенных доменов.

Во вкладке «Options» можно настроить лимит, то есть количество запросов, которое Charles Proxy может записать.

Во вкладке «Include» можно выбрать конкретный домен для отображения пакетов.

Во вкладке «Exclude» можем выбрать те домены, которые необходимо спрятать при сниффинге.

Focus — эта функция перемещает домен на первые позиции в списке.
Repeat — отправляет на сервер запрос, идентичный выбранному.
Repeat Advanced — идентично Repeat, только можно выбрать количество отправляемых запросов и задержку между ними. Эта функция пригодится при проверке реакции сервера на флуд.
Здесь Concurrency — количество пользователей, а Iterations — количество повторений каждого запроса. Также можно поставить галочку «Show results in new Session», в таком случае откроется новое окно, где будут выполняться запросы.

Список функций Charles Proxy этим, конечно, не ограничивается.

Следует иметь ввиду, 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.