Charles Proxy Tutorial

Charles — инструмент для мониторинга HTTP/HTTPS трафика. Программа работает как прокси-сервер между мобильным приложением и сервером этого приложения. Charles записывает и сохраняет все запросы, которые проходят через подключенный к нему телефон и позволяет их редактировать.

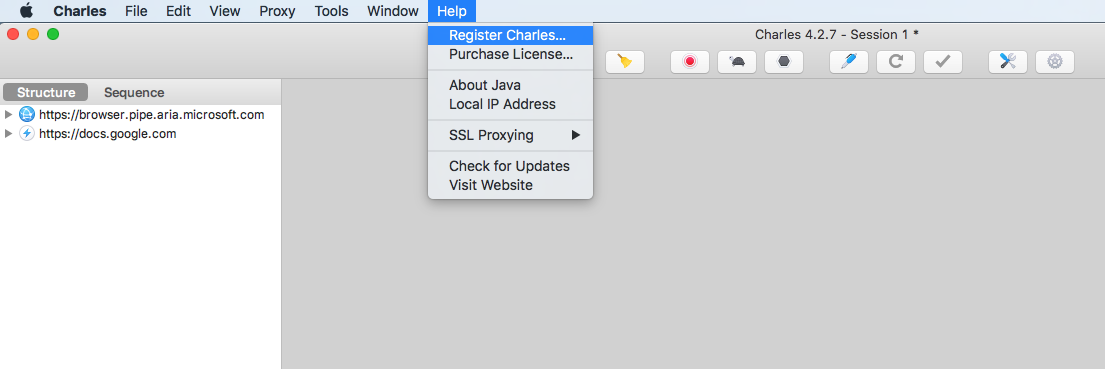
## 1. Установите Charles на компьютер

Для начала вам нужно скачать приложение и установить его на ПК

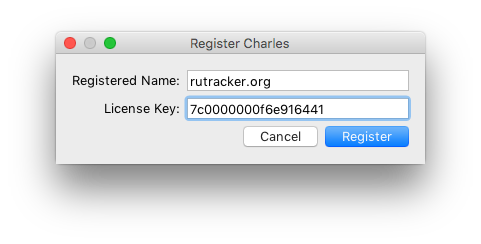
(<https://www.charlesproxy.com/download/>).

Так как вы скачиваете бесплатную пробную версию, Charles будет выключаться каждые 30 минут. Чтобы такого не происходило, необходимо ввести ключ.

В меню выбираем Help > Register Charles:



Появится окно такого вида:



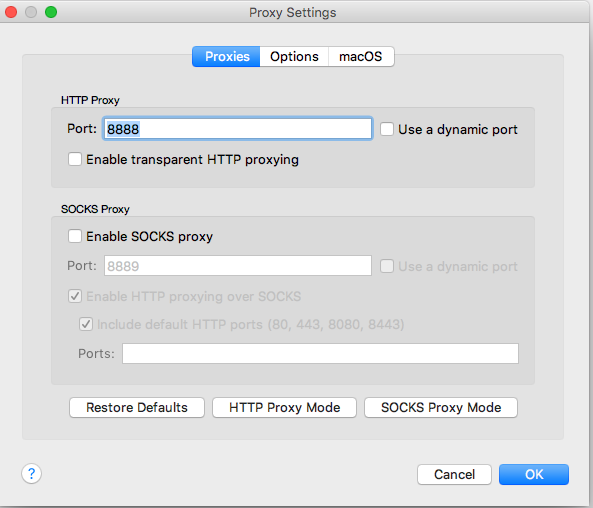
Вводим Registered Name: **rutracker.org** и License Key: **7c0000000f6e916441**, затем нажимаем Register.

После этого вам необходимо перезапустить Charles.

## 2. Откройте Charles Proxy для начала сессии

1. При первом запуске Charles предложит автоматически настроить параметры сети — соглашаемся (кнопка Allow).
2. Перейдите в меню в раздел Proxy > Proxy Settings и выставьте порт 8888, если он еще не прописан.

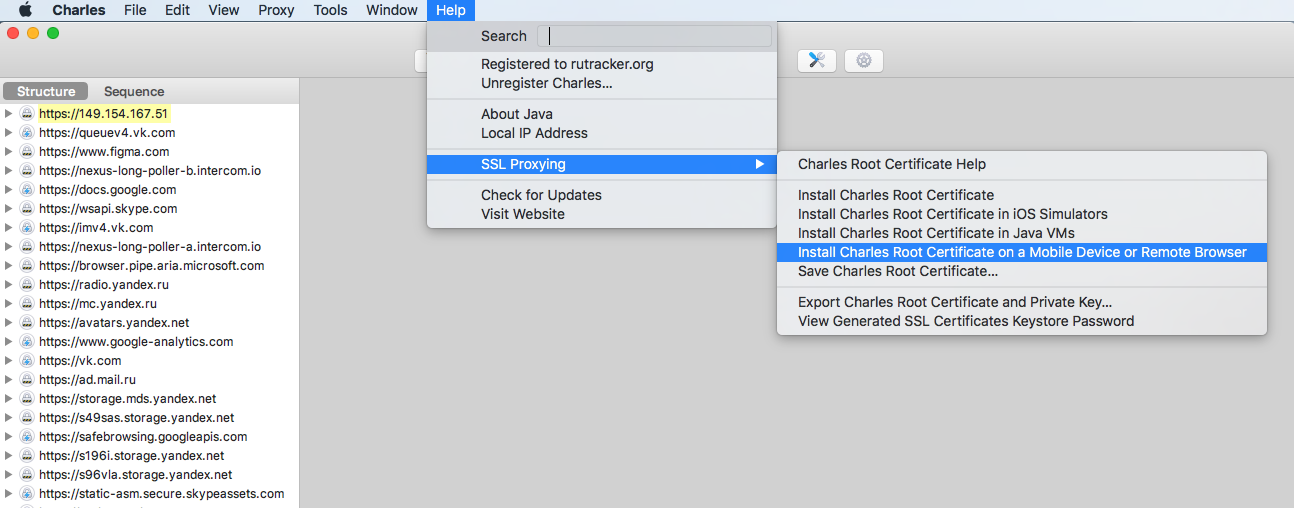
Если он занят другими программами (Charles оповестит), его можно изменить. Порты могут быть от 1 до 65535, причём от 1 до 1024 (или 4096) - системные порты. Ставьте стандартные, например, 8080, 1080, 8888 и т.п. Принцип состоит в том, чтобы компьютер и мобильный телефон согласовывались.



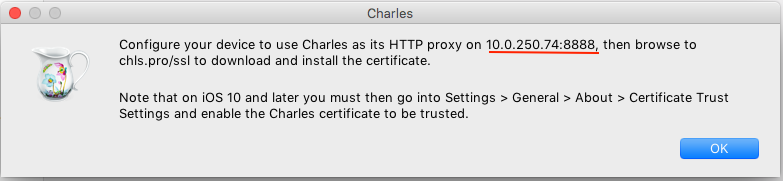
## 3. Найдите IP-адрес и номер порта вашего компьютера

Мы хотим проверить трафик, поступающий со смартфона, поэтому мы начнем с настройки iPhone, чтобы он использовал наш компьютер с запущенным Charles Proxy, как прокси-сервер.

Сначала на компьютере, на котором работает Charles, выберите Help > SSL Proxying > Install Charles Root Certificate on a Mobile Device or Remote Browser

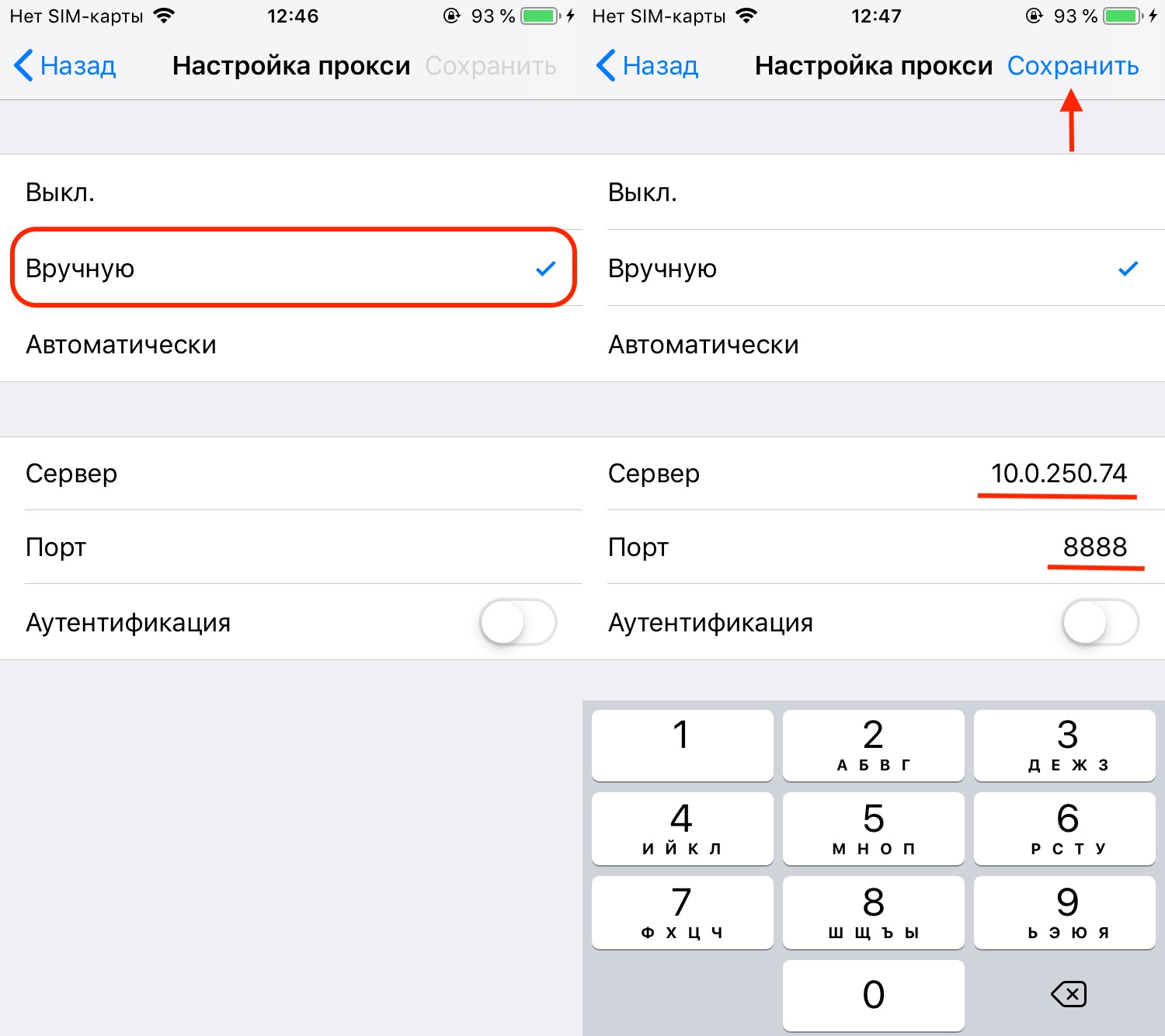
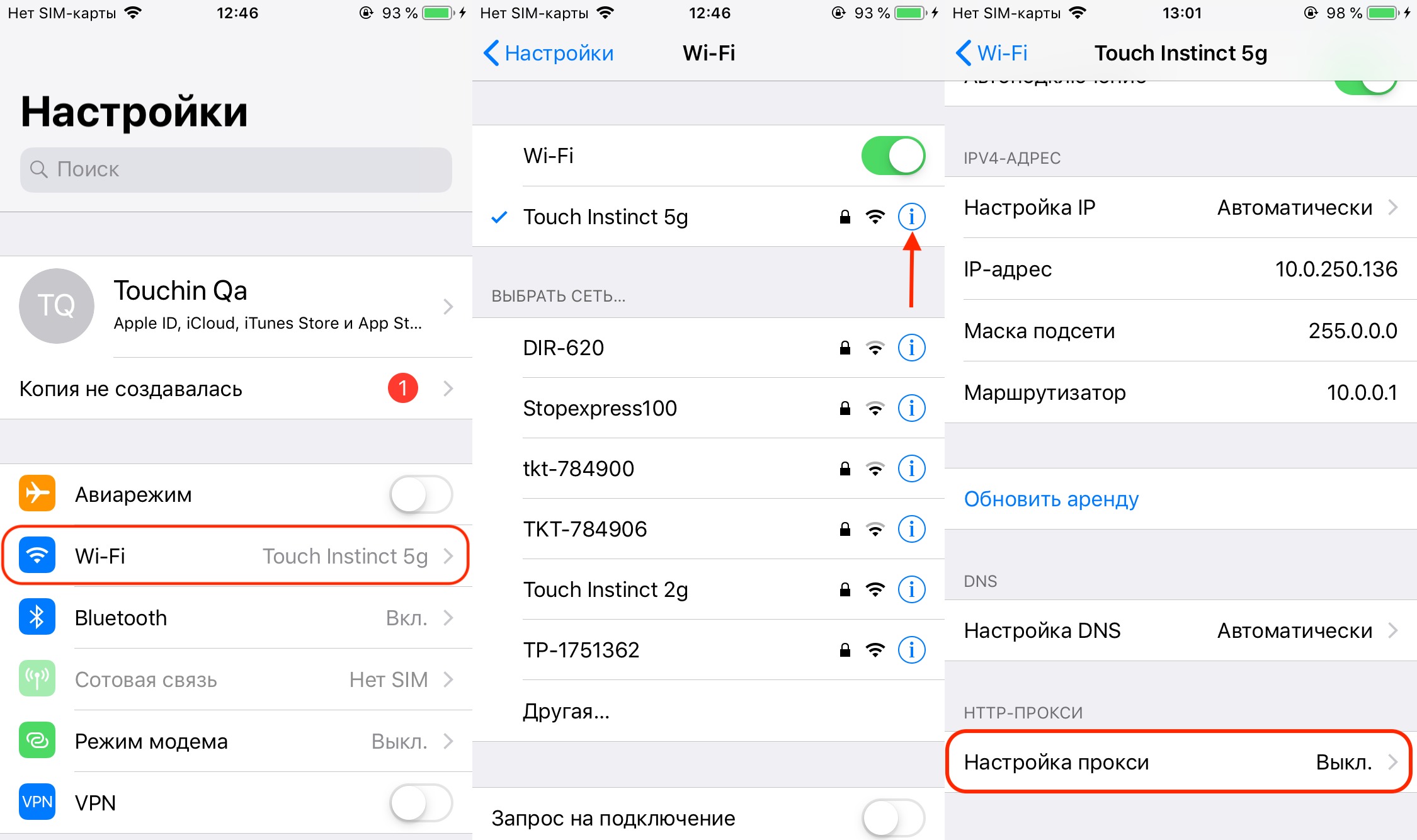


Вы увидите диалоговое окно, подобное указанному на скриншоте ниже, которое предоставляет IP-адрес и порт для машины, на которой работает Charles:



## 4. Настройка Вашего телефона для использования Charles Proxy

На iPhone перейдите к меню “Wi-Fi” в Настройках, затем убедитесь, что он находится в той же сети, что и Charles. Откройте системные настройки > Сеть > Дополнительные настройки > Прокси > **Вручную**. Здесь введите адрес сервера и номер порта из последнего диалогового окна, которое выдал Charles. Нажмите “Сохранить”, чтобы закончить:



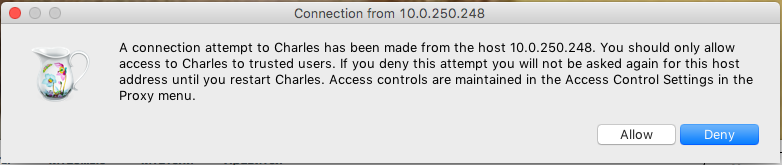
На Андроиде шаги аналогичны.

Теперь ваш смартфон настроен на маршрутизацию HTTP-трафика через Charles Proxy.

## 5. Разрешите соединения к Чарльзу

Когда ваш смартфон будет подключен к Charles, то на вашем компьютере вы увидите диалоговое окно, предупреждающее о том, что Charles получил входящее соединение. Выберите “Allow”. Если вы вдруг случайно запретите входящее соединение, то для того, чтобы еще раз получить это приглашение, вам придется перезапустить Charles.

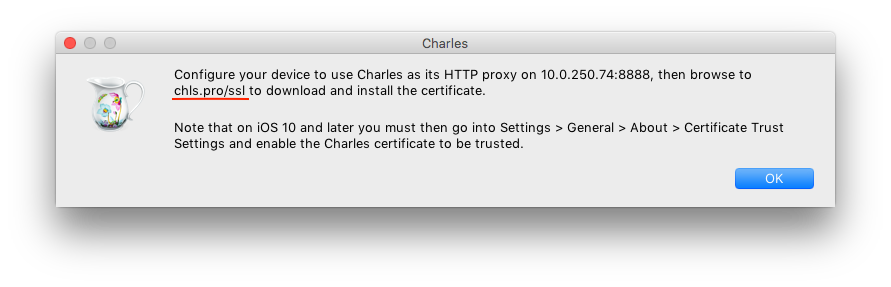
Если диалоговое окно не появилось, попробуй выйти в интернет с девайса (любая страница в гугле, например).



## 6. Установите на ваш телефон корневой сертификат

Информация недоступна для хостов с использованием HTTPS. Это связано с тем, что HTTPS-трафик использует криптографический протокол под названием Transport Layer Security для шифрования данных между серверами и пользователями. Charles предоставляет нам возможность устанавливать и доверять корневому SSL-сертификату, уникальному для каждой установки. Он позволяет видеть HTTPS-данные в виде простого текста.

Для этого вы можете перейти в меню Help > SSL Proxying > Install Charles Root Certificate on a Mobile Device or Remote Browser.



Через браузер телефона переходим по ссылке, указанной в диалоговом окне выше ([chls.pro/ssl](http://chls.pro/ssl)).

На iPhone вы сразу же получите запрос с просьбой разрешить сайту показывать вам профиль конфигурации. Нажмите “Разрешить”. После этого вы попадете на экран Install Profile для Charles Proxy CA. Нажмите “Установить”, вас перенаправят на страницу предупреждения. Нажмите “Установить” еще раз, и еще раз на всплывающем окне. Наконец, нажмите “Готово”, чтобы закончить.

Если ваш девайс работает на iOS 10 или выше, то вам также придется посетить Настройки > Основные > Об этом устройстве > Доверие сертификатов, свитч напротив вашего установленного сертификата должен быть включен.

А на Android-устройстве файл попытается немедленно загрузиться на ваше устройство. Если этого не произошло, то вы можете получить приглашение для его загрузки, и тогда вам нужно нажать на “Скачать”. После загрузки введите ваш PIN-код или отпечаток пальца. После этого откроется окно сертификата, в котором вам нужно назвать сертификат чем-то вроде “Charles Proxy CA”. Нажмите “ОК”, когда закончите.

На некоторых девайсах сертификат не может предложить установку, попробуй найти его в загрузках.

Иногда установка после автоматического скачивания и запуска сертификата в android он устанавливается, но не так, как надо. В Charles данные остаются зашифрованными. Например на Samsung, android 11 (в момент установки, к тому же, не происходит запрос PIN-кода или отпечатка пальца). В этом случае надо зайти в Настройки > Биометрия и безопасность > Другие параметры безопасности > Установить из памяти (установить сертификаты с носителя) > Сертификат CA. Далее появляется окно с предупреждением, где надо нажать “Установить в любом случае”, ввести PIN-код (или отпечаток пальца) и найти сертификат на устройстве. После выбора сертификата он устанавливается в нужное место (имя сертификата вводить нигде не надо).

Пример на iPhone:

|  |  |  |
| --- | --- | --- |

Теперь, когда у вас установлен сертификат, вы сможете в Charles увидеть зашифрованные HTTPS-данные в виде простого текста, а не только данные HTTP, просматриваемые без сертификата.

*\*Если не получается установить сертификат и вызвать диалоговое окно, необходимо проверить настройки брандмауэра:*

***Для Windows:***

1. *Нажмите кнопку Пуск*
2. *Выберите Параметры – Обновление и безопасность – Безопасность Windows – Брандмауэр и защита сети*
3. *Открыть параметры службы «Безопасность Windows»*
4. *В разделе Брандмауэр Microsoft Defender установите для параметра значение Выкл.*

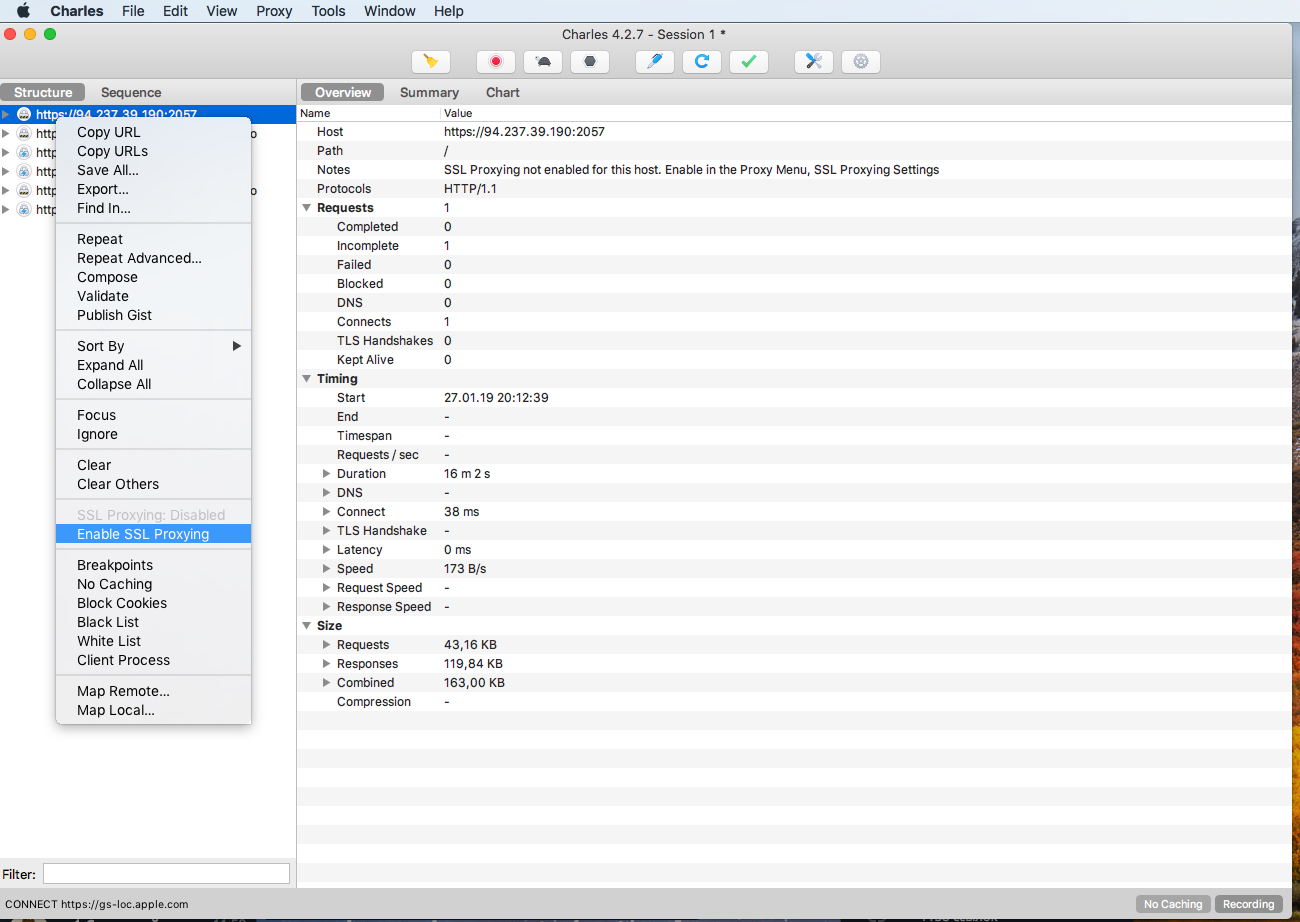
***Для Mac:***

1. *Выберете пункт меню Apple https://help.apple.com/assets/62A8ED2F99A5D0045E612A42/62A8ED3DCA32D1047032C8FF/ru_RU/2f77cc85238452e25cb517130188bf99.png - «Системные настройки»*
2. *Нажмите «Защита и безопасность» https://help.apple.com/assets/62A8ED2F99A5D0045E612A42/62A8ED3DCA32D1047032C8FF/ru_RU/106ba2d7107ef06182690a344e00b1ce.png, затем нажмите «Брандмауэр».*
3. *Если слева внизу отображается запертый замок, нажмите его, чтобы разблокировать панель настроек*
4. *Нажмите «включить брандмауэр»*

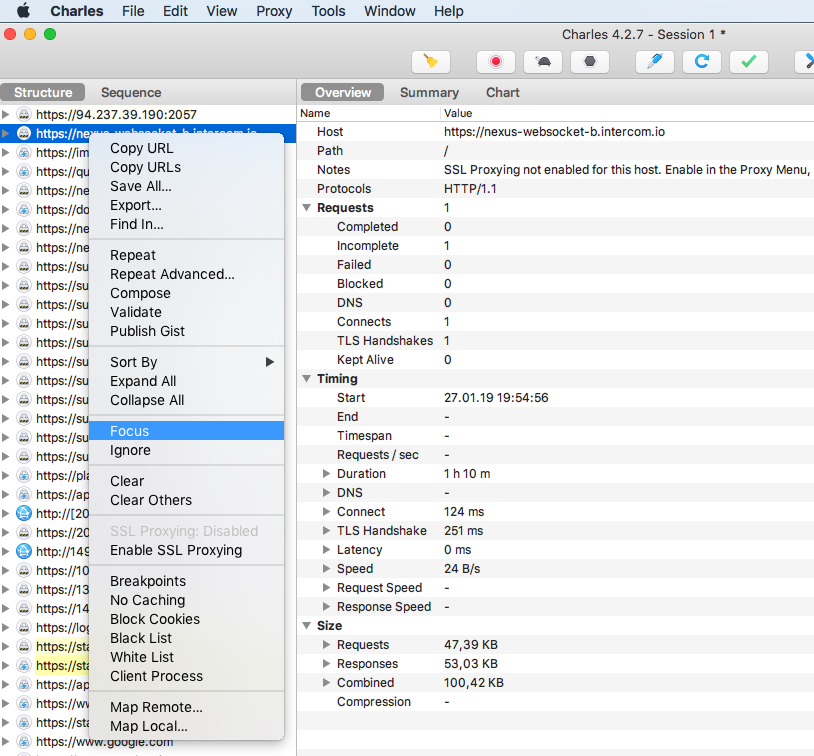
## 7. Мониторинг трафика

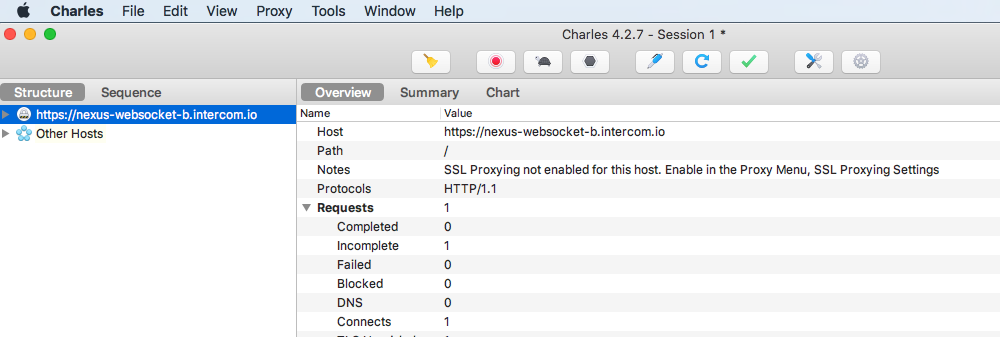
Если приложение ранее было открыто, то его следует перезапустить.

Далее снова обращаемся к Charles’у: в списке появившихся хостов после перезапуска приложения выбираем желаемый хост с HTTPS, кликаем правой кнопкой мыши по нему и ставим галочку Enable SSL Proxying.



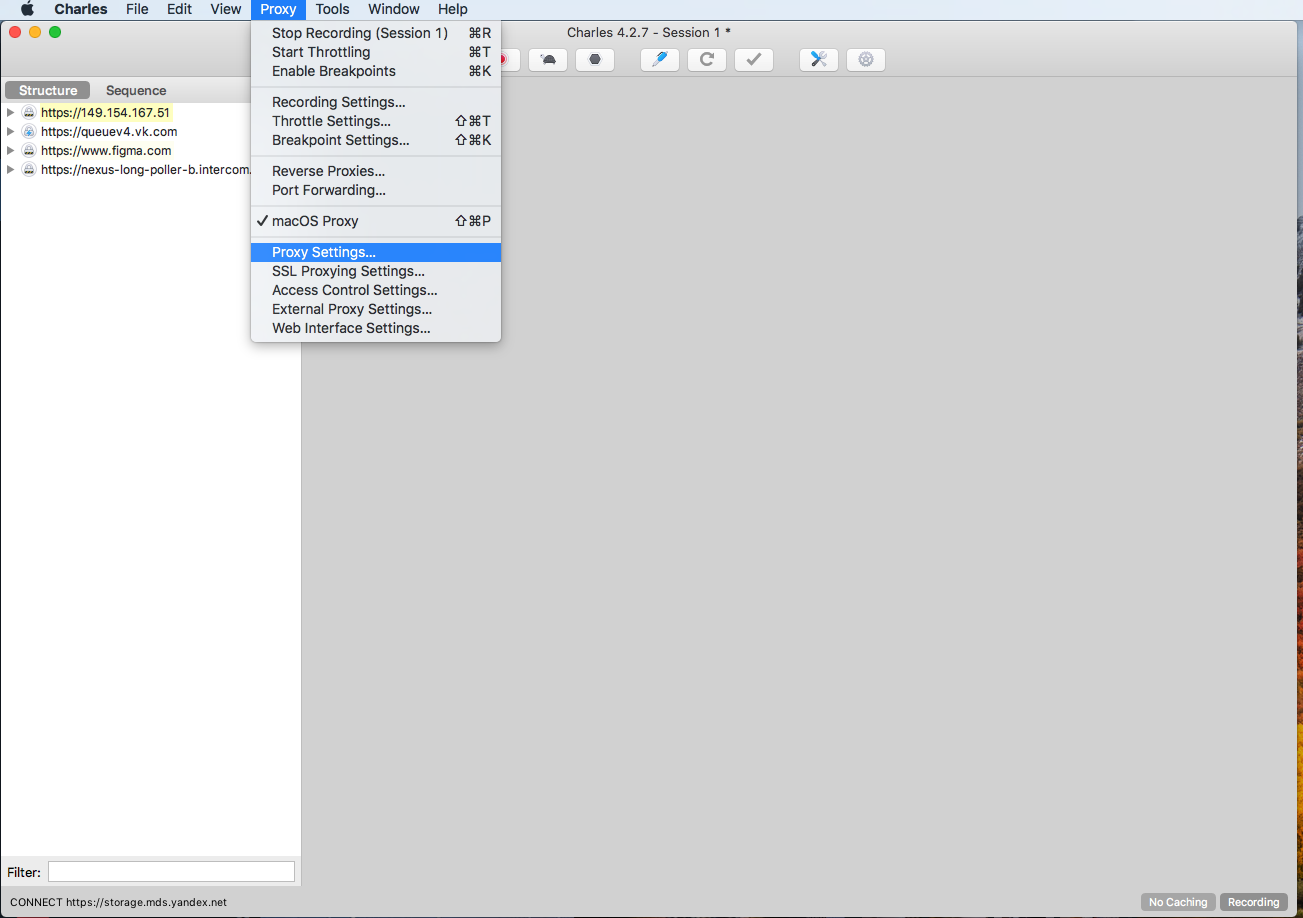
Тут же можно поставить галочку **Focus** на хосте с которым мы хотим работать - таким образом, весь не интересующий нас трафик уйдет в папку **Other hosts.**





Есть одна удобная вещь: проксирование всех хостов сразу, чтобы не приходилось вручную каждому хосту включать “Enable SSL Proxying”:

Для этого нужно перейти в меню “Proxy”, затем выбрать “SSL Proxying Settings”.



Когда параметры появятся, поставьте галочку в чекбокс “Enable SSL Proxying”, нажмите кнопку “Add” на вкладке “SSL Proxying”. Когда появится окно “Edit Location”, просто нажмите “ОК” или вручную введите \* (одну звездочку) в поле “Host” и нажмите “ОК”. Это подстановочный знак, который означает, что настройки применяются ко всем хостам. Если оставить поле Host пустым, то поле само создаст подстановочный знак. Для сохранения изменений в окне настроек SSL Proxying обязательно нажмите “ОК”.

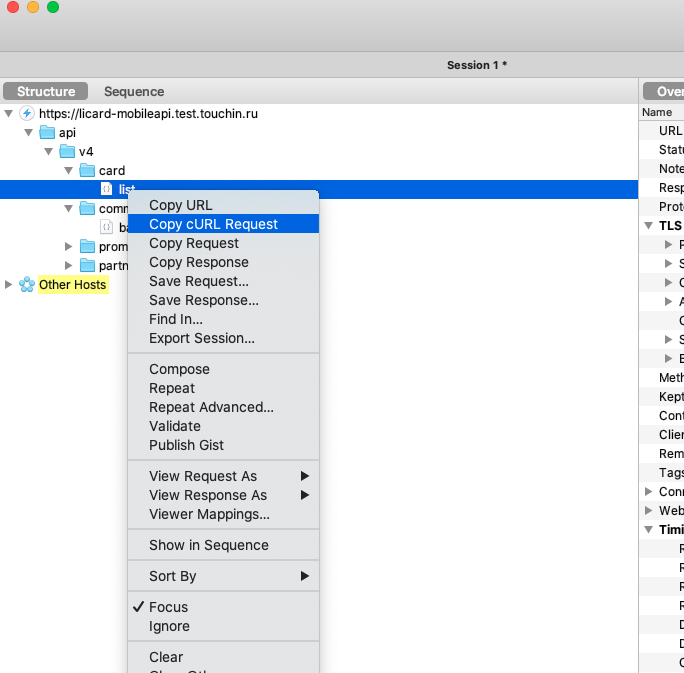


Теперь, все данные, перехваченные Charles Proxy, отображаются!

## 8. Копирование cURL запроса

Для быстрой передачи запросов сделанным приложением есть возможность скопировать запрос со всей нужной информацией в удобоваримом виде (со всеми заголовками и телом запроса).

Для этого нужно выбрать нужный запрос и в контекстном меню выбрать **Copy cURL request.**

****

Вы скопировали curl и теперь можете отправить его в текстовом виде.

Из примера должно получится что то похожее:

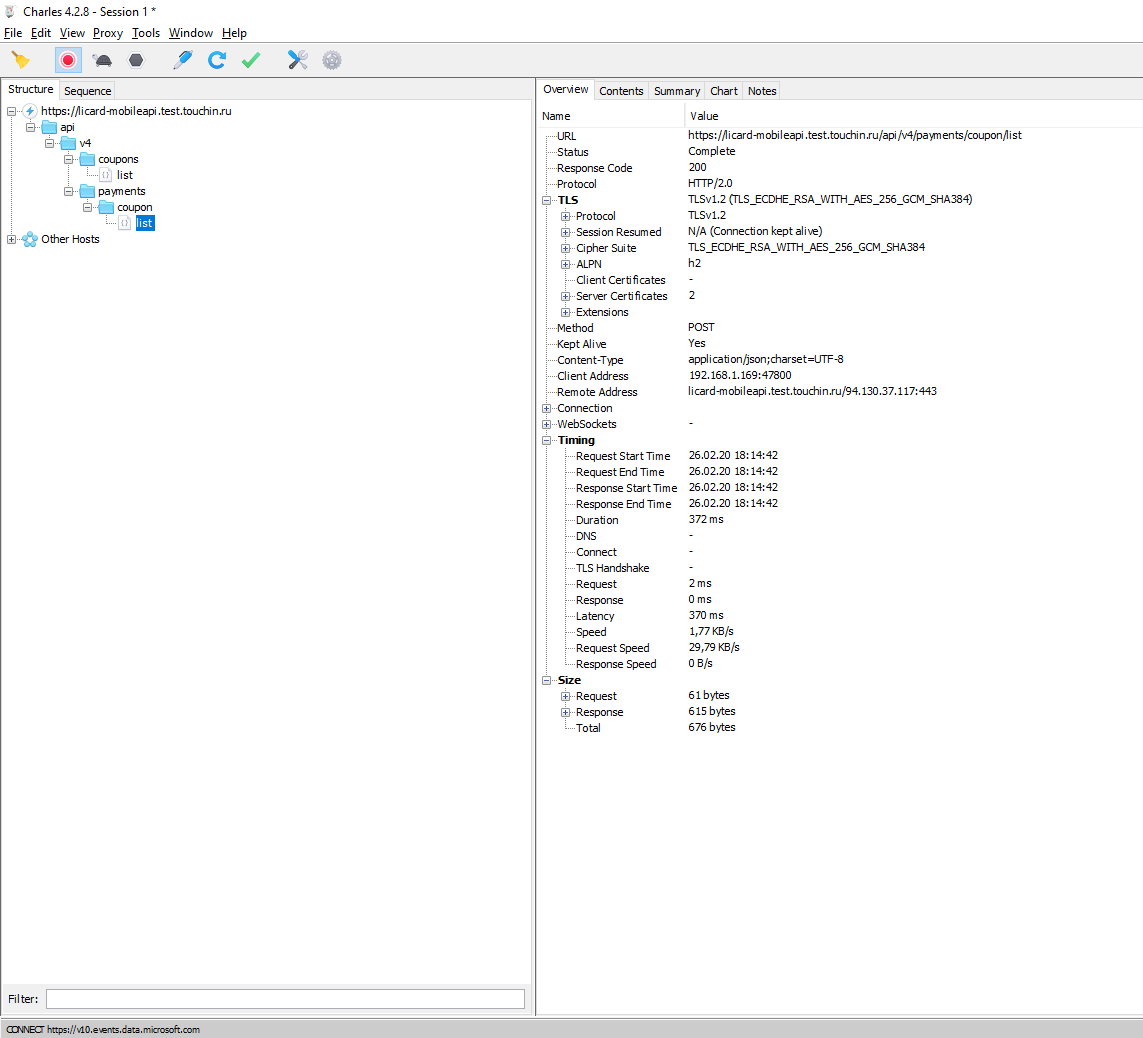
curl -H 'Host: licard-mobileapi.test.touchin.ru' -H 'x-api-token: mcHySTn5vmPvMLWrYMfG3xgC9rV2moJ6' -H 'user-agent: WAS-LX1A, 8.0.0' -H 'content-type: application/json; charset=UTF-8' --data-binary '{"sessionId":"072c1463-8ca5-4901-b574-beb605661c3f"}' --compressed 'https://licard-mobileapi.test.touchin.ru/api/v4/card/list'

# Некоторые техники работы с Charles Proxy

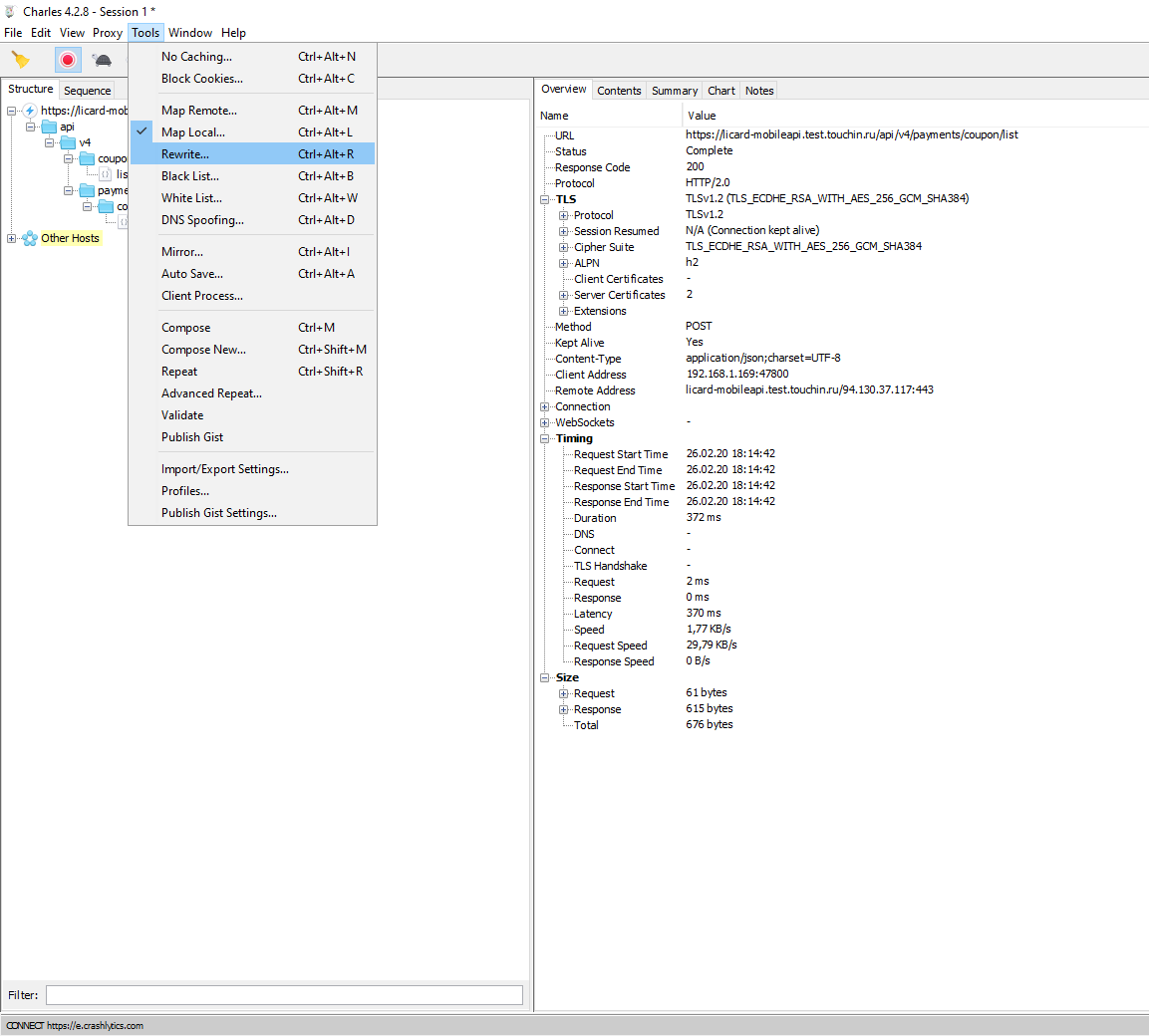
## 1. Rewrite (Подмена данных)

Rewrite позволяет создавать правила, изменяющие запросы и ответы при прохождении через Charles. Такие правила, как добавление или изменение Header’ов, поиск и замену текста в теле ответа (Body) или части URL-адреса запроса.

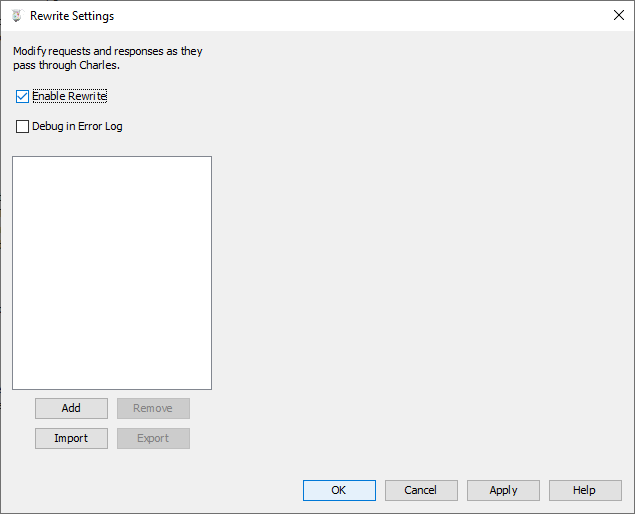
Чтобы начать работать с подменой данных с помощью данного инструмента, вам необходимо выделить мышкой запрос, в котором вы хотите что-то подменить, кликнув по нему левой кнопкой мыши.



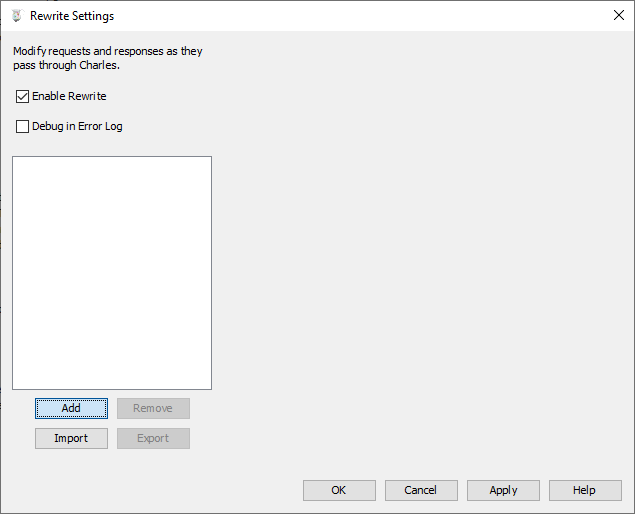
Далее вам необходимо зайти в меню в раздел “Tools” > “Rewrite”:



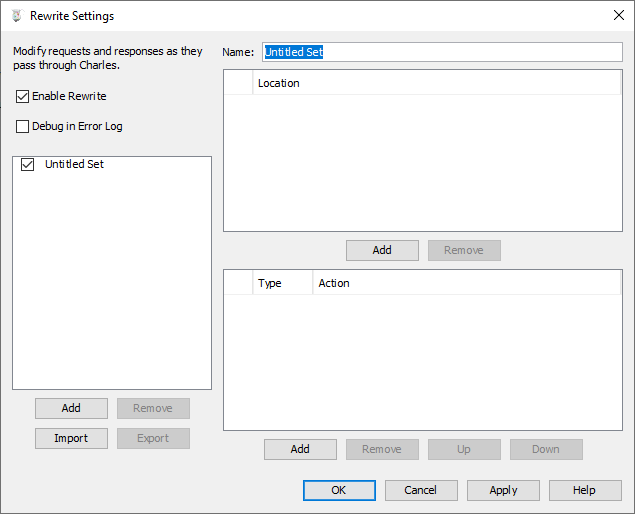
Появится окно для настроек подмены. И для начала работы с ними, вам нужно поставить галочку в чекбокс напротив “Enable Rewrite”:



Затем нажимаем кнопку “Add”:

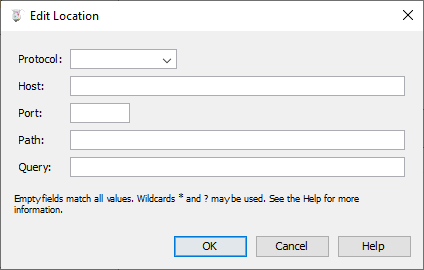


В поле “Name” вы можете поменять название правила, чтобы различать их друг от друга.



После этого нужно нажать следующую кнопку “Add” во вкладке “Location”.

После чего появится данное окно:

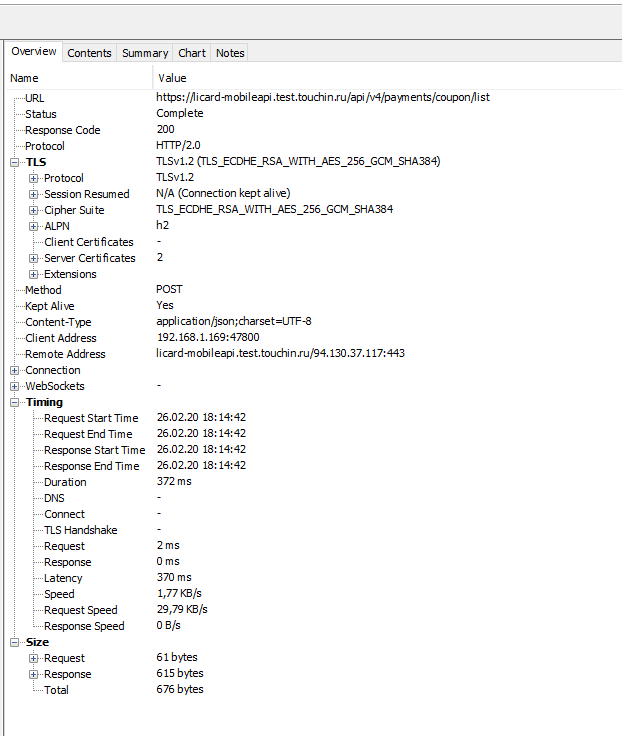


Вам нужно заполнить поля Protocol, Host, Port и Path той же информацией что и у целевого запроса. Для этого вы можете открыть выбранный вами запрос, но во вкладке “Overview”. На примере запроса payments/coupon/list.

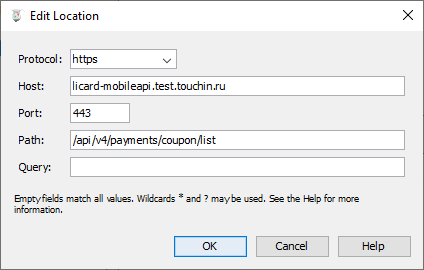
В поле URL на окне Overview мы можем увидеть

1. Protocol - в нашем случае https
2. Host - Все что после протокола и до пути, в нашем случае licard-mobileapi.test.touchin.ru
3. Path - Все что после домена. В нашем случае /api/v4/payments/coupon/list

Port берется из поля Remote Address. В нашем случае 443

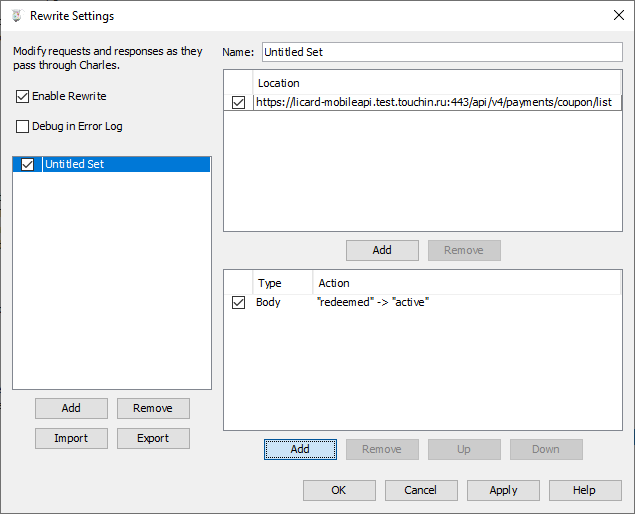


И получатся такие заполненные настройки:

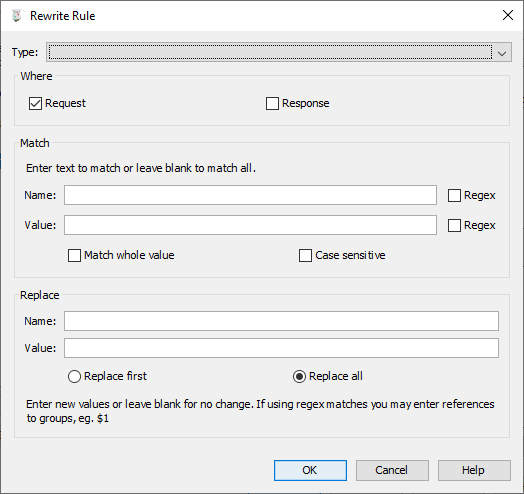


Нажать ОК.

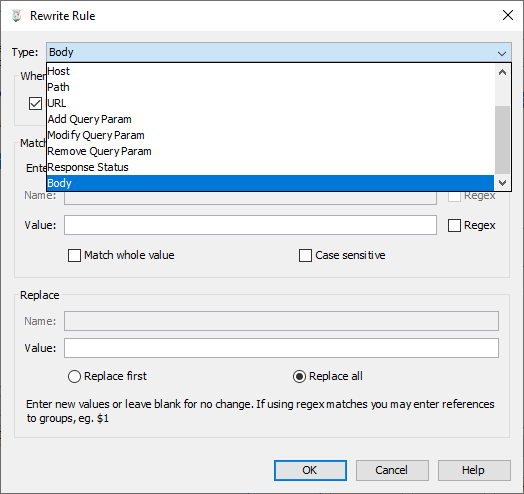
После этого вы вернетесь к основному экрану настроек подмены, и на нем уже нужно будет нажать третью кнопку “Add”:



Появится окно правила подмены, которое тоже нужно заполнить:

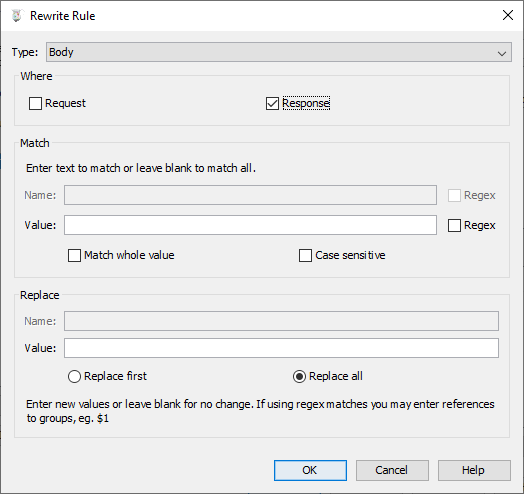


* Для начала вам нужно выбрать Type того, что хотите подменить. Для это вам нужно нажать на поле Type - появится выпадающий список типов подмен:



Покажу на примере подмены тела ответа (“Body”), соответственно, в типе подмены выбираем “Body”

* Затем вам необходимо выбрать, что вы хотите подменить: запрос, ответ или же всё сразу. То есть во вкладке “Where” необходимо поставить галочку(и) в чекбоксы напротив “Request” или “Response” в зависимости от ваших нужд. В примере выберем тело ответа .

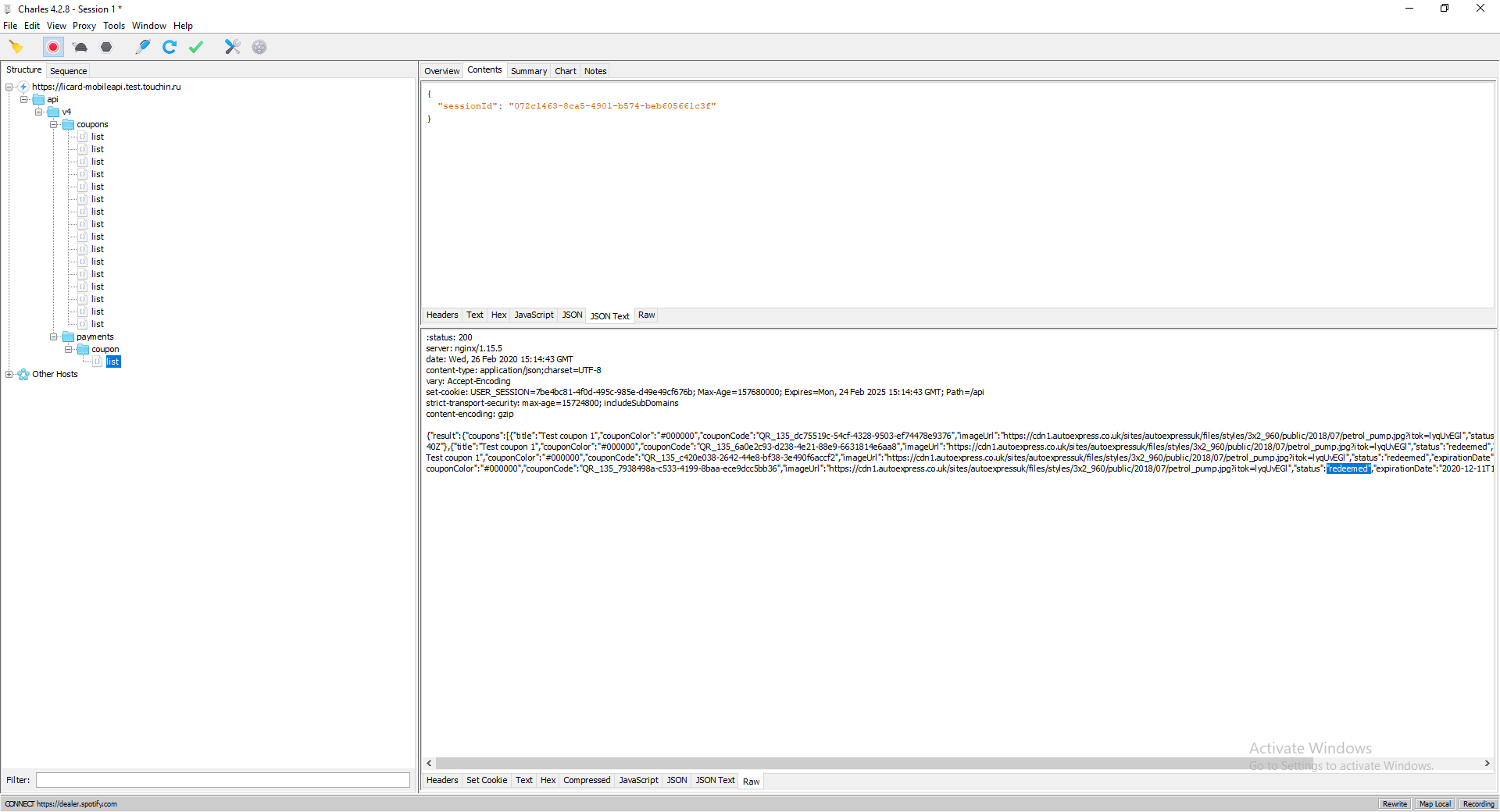


* Далее вам нужно во вкладке “Replace” в поле “Value” ввести в raw-формате данные, пришедшие от сервера, которые вы хотите подменить.

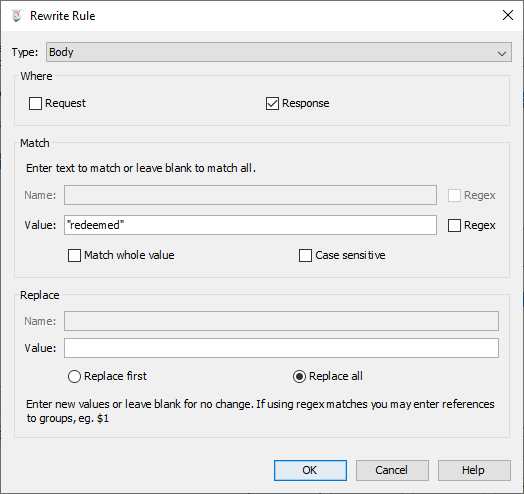
Для этого можно вернуться обратно к выбранному запросу и найти во вкладке “Contents” ответ от сервера, который мы хотим подменить, и выбрать формат его отображения в “Raw” (этот формат выбирать лучше, т.к. в отличие от приятного глазу json-text вы не скопируете лишние пробелы, из-за которых правило подмены может не сработать).

На примере выделим значение поля “status” в котором приходит статус купонов в нашем приложении и сделаем так чтобы они всегда приходили активными.

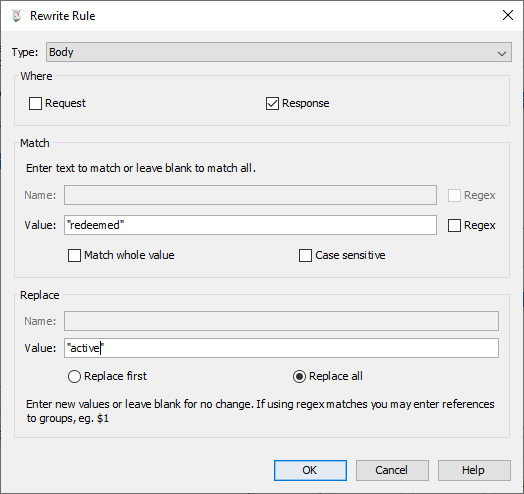
Находим нужное поле и копируем его:



* После этого вам необходимо заполнить поле “Value” во вкладке “Match” - тут заполняется то, что вы хотите подменить в самом запросе. Для этого можно вставить скопированный текст ответа от сервера в предыдущем шаге:



Далее в поле Value во вкладке Replace нужно внести то на что должно заменяться значение из вкладки Match. В нашем случае мы хотим чтобы все купоны были активными, поэтому введем туда “active” и поставим радиобаттон Replace all

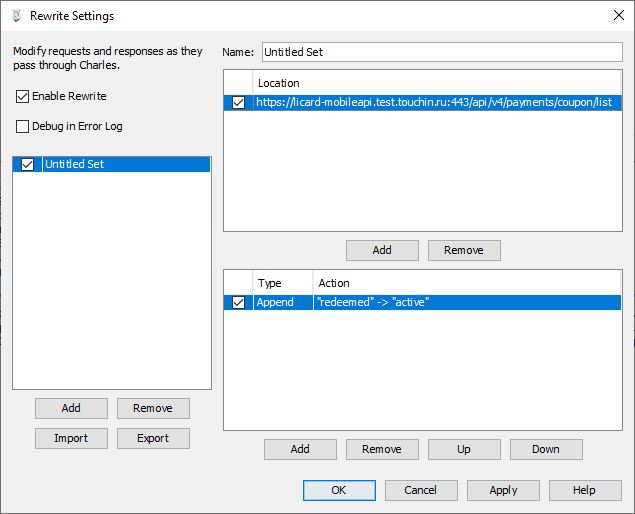


* Замена поддерживает режимы “Replace first” или “Replace all”. Заменяет первое найденное значение либо все соответственно.

Также есть возможность не вводить точные значения полей Name и Value а использовать регулярные выражения для более эффективной замены значений. Процесс замены остается таким же, только в поля Name/Value нужно ввести регулярное выражение и поставить чекбокс Regex справа от поля.

Нажимаем “ОК”.

Так выглядит наша настройка подмены целиком:



В конце нужно нажать “Apply”, а затем “ОК”.

Теперь в приложении вам необходимо послать этот запрос заново.

| До подмены имелось: | Получили: |
| --- | --- |
|  |  |

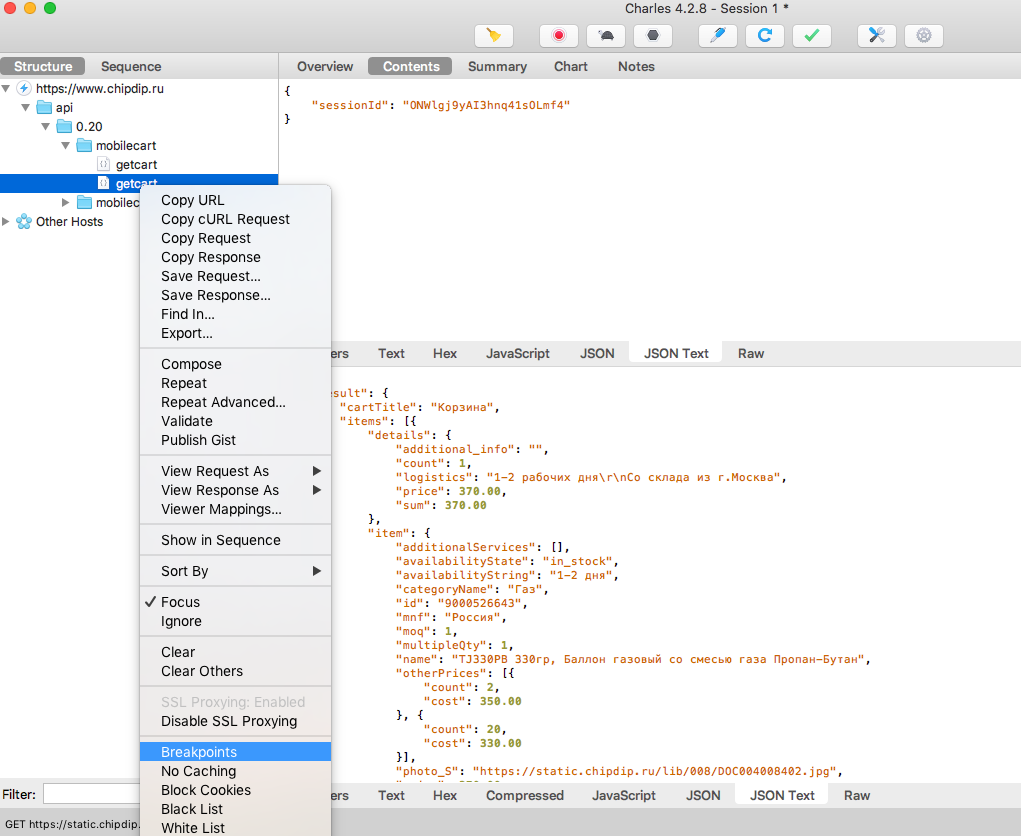
Rewrite удобно использовать для быстрого точечного изменения параметров не прибегая к переписыванию всего через Map local или ручного перебивания параметров через брейкпоинты.

## 2. Breakpoints

Инструмент Breakpoints позволяет перехватывать запросы и ответы до того, как они будут переданы через Charles. Вы можете просмотреть и отредактировать запрос или ответ в реальном времени, а затем решить, разрешить ли его выполнение или заблокировать.

Важная особенность - отредактировать запрос и/или ответ вы можете только в течение времени ожидания ответа от сервера, установленного в приложении (таймаут). Поэтому с брейкпоинтами нужно крайне оперативно работать. Из-за этого брейкпоинты удобно использовать, если подменить нужно не очень много информации.

Кликаем правой кнопкой мыши по методу, в котором хотим что-либо подменить. Появляется меню -> **Breakpoints**

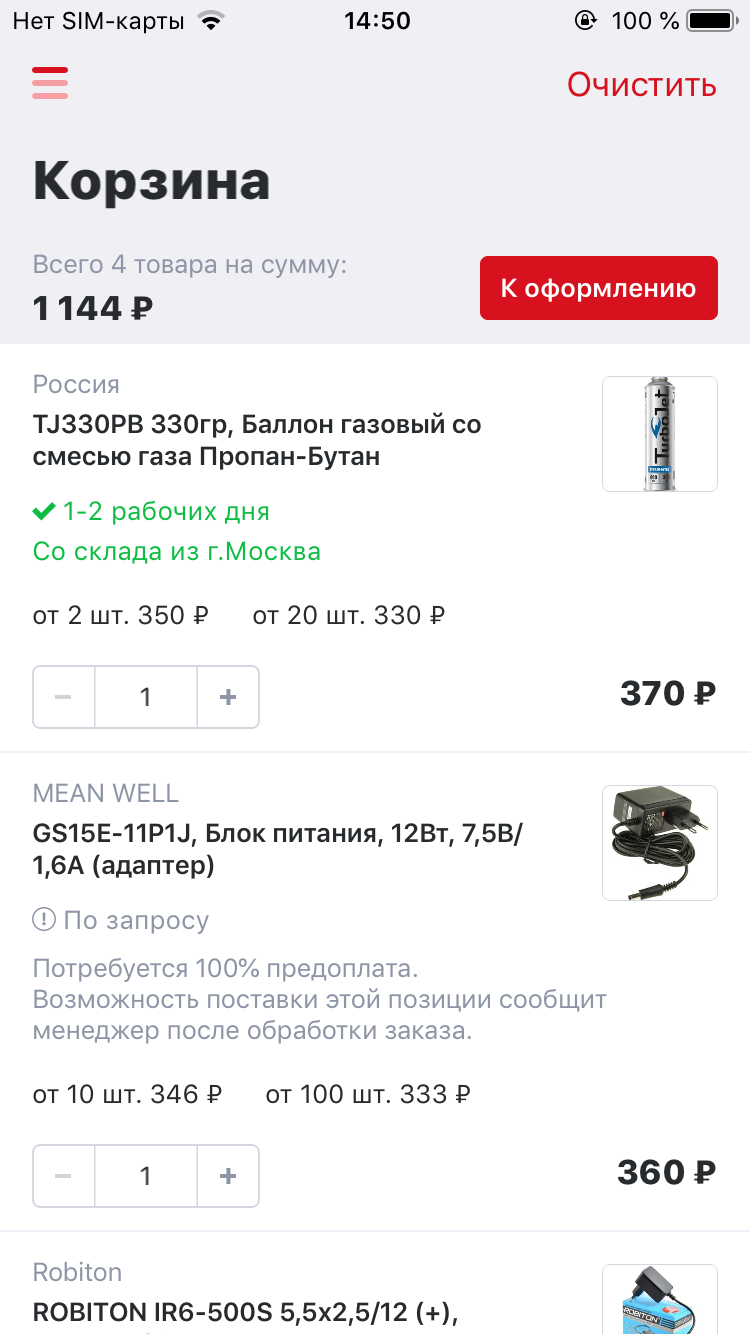


После этого действия на панели инструментов иконка с галочкой в кружочке должна включиться:

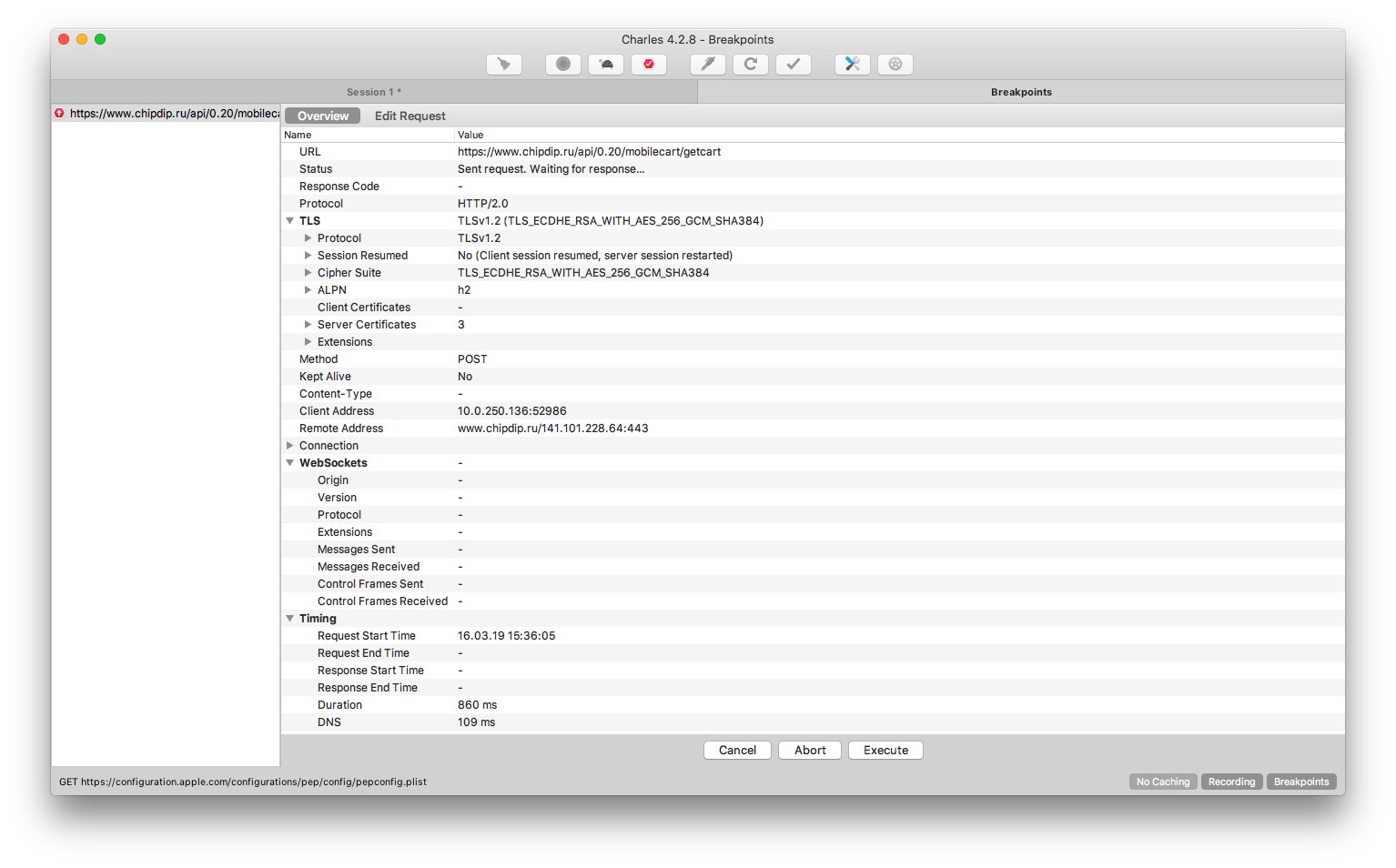


Чтобы увидеть, как работает данная техника, не обязательно перезапускать приложение. Вы можете просто в мобильном приложении повторить запрос, с которым хотите поработать.

Мой пример: в приложении хочу изменить стоимость первого по счету товара в корзине. Для этого мне необходимо будет изменить ответ от сервера.



Делаю запрос, на который ранее установила брейкпоинт -> автоматически открывается окно точки остановки:



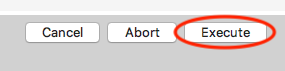
На скриншоте сверху окно для работы с запросом. Но так как мне нужно изменить ответ от сервера - нажимаем на **Execute** - кнопку, которая применяет любые внесенные изменения и позволяет продолжить запрос или ответ (в нашем случае выполнение без изменения чего-либо в запросе):{

"code": "string",

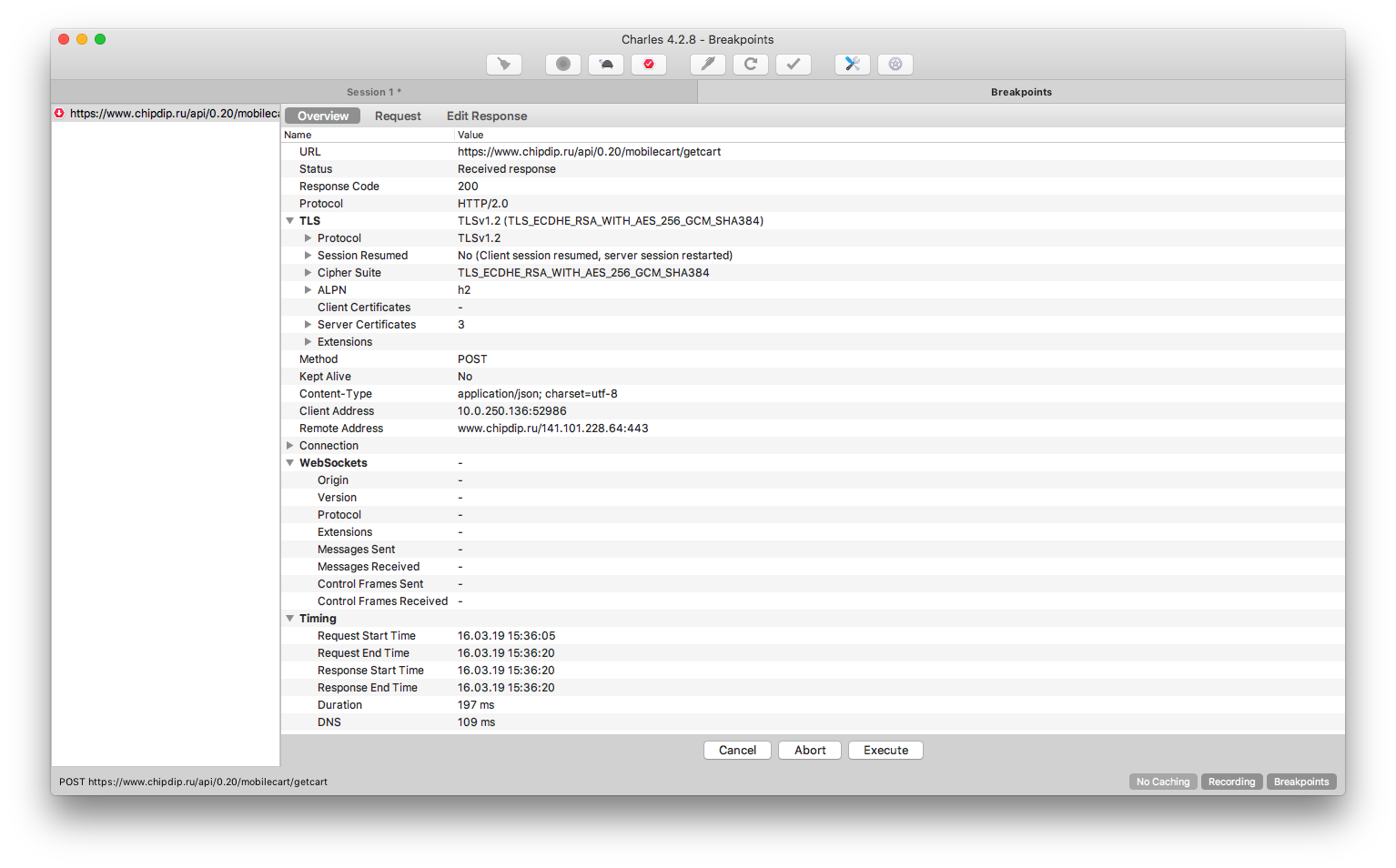
"errorMessage": "string",

"debug": "string"

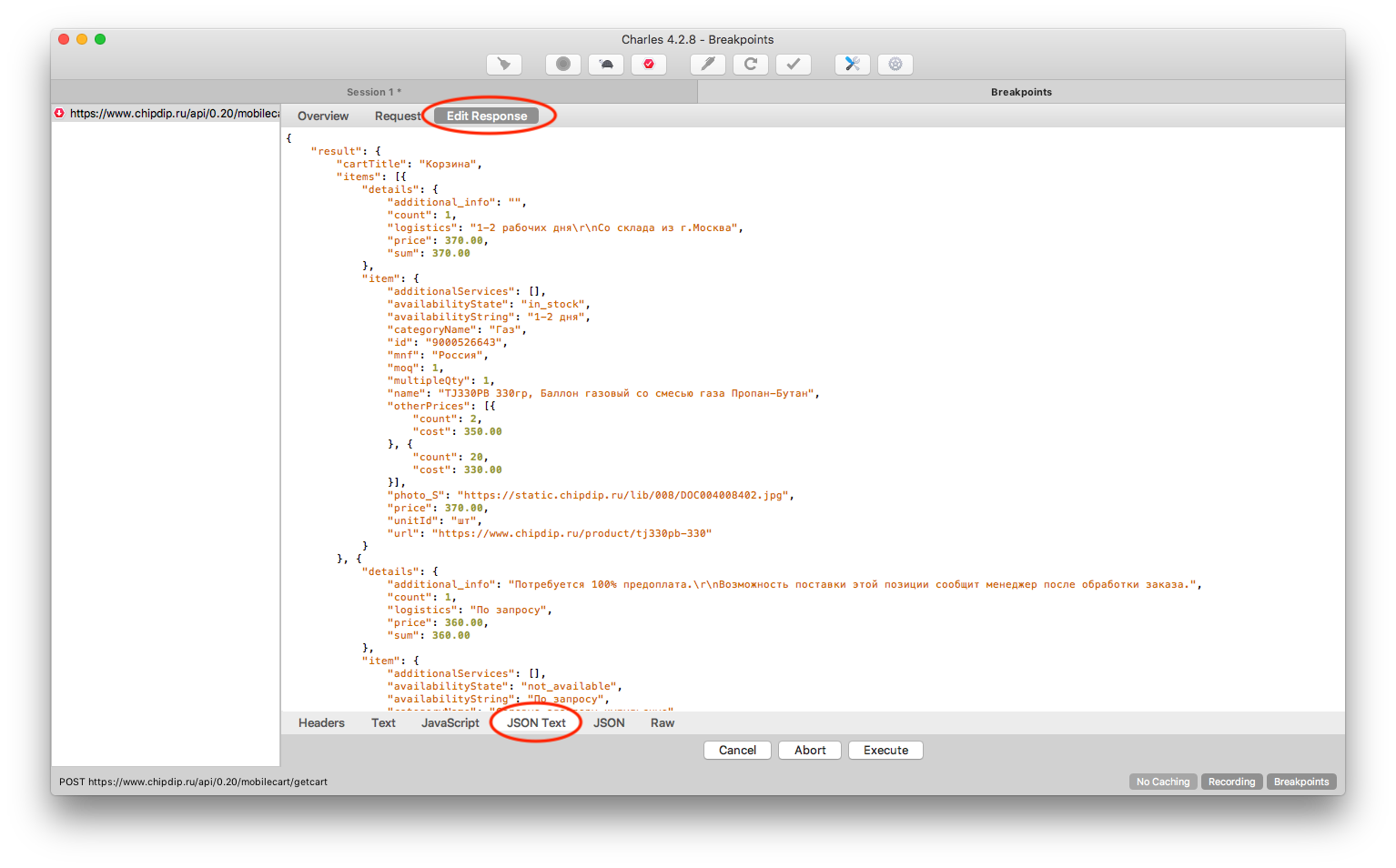
}



После этого на экране появляется окно для работы с ответом:



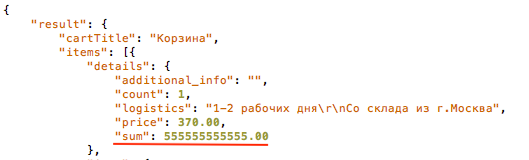
В данном окне необходимо перейти во вкладку Response и в ней выбрать формат отображения - json-text:



Итак, вы видите, что приходит от сервера, и можете что-либо подменять. На скриншоте ниже я указала, что буду подменять, чтобы в приложении получить желаемый результат:

# 

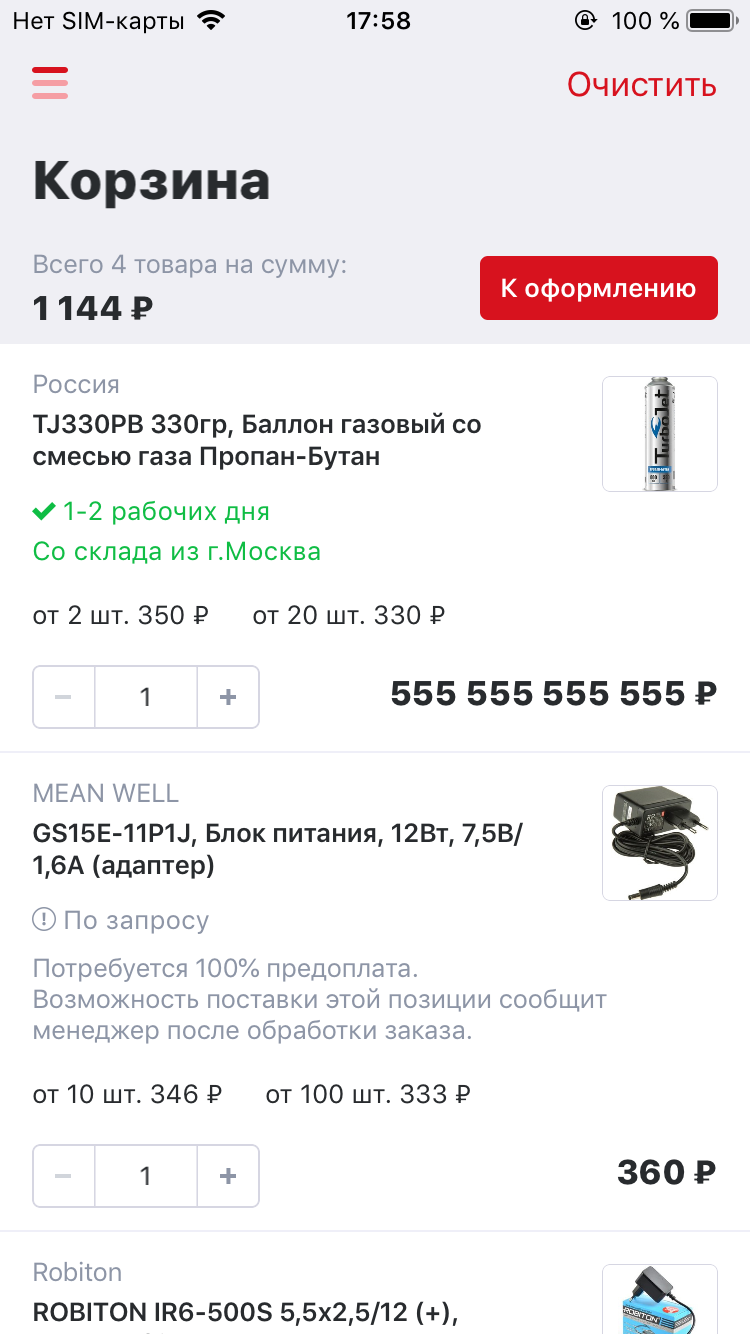
Меняем значение выделенного поля:



Затем необходимо нажать Execute, чтобы изменения вступили в силу:



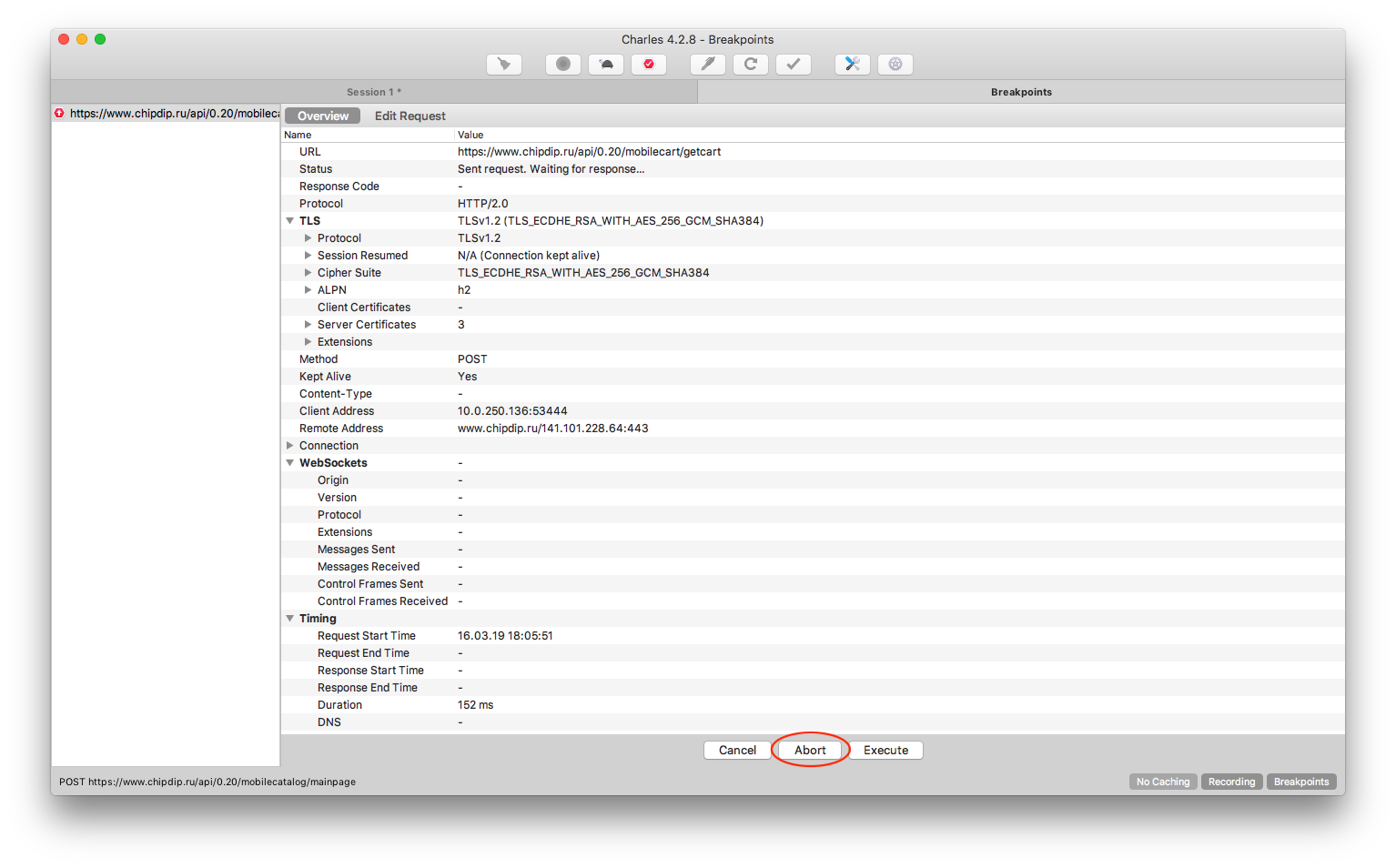
Смотрим в мобильное приложение и видим успешную подмену ответа сервера:



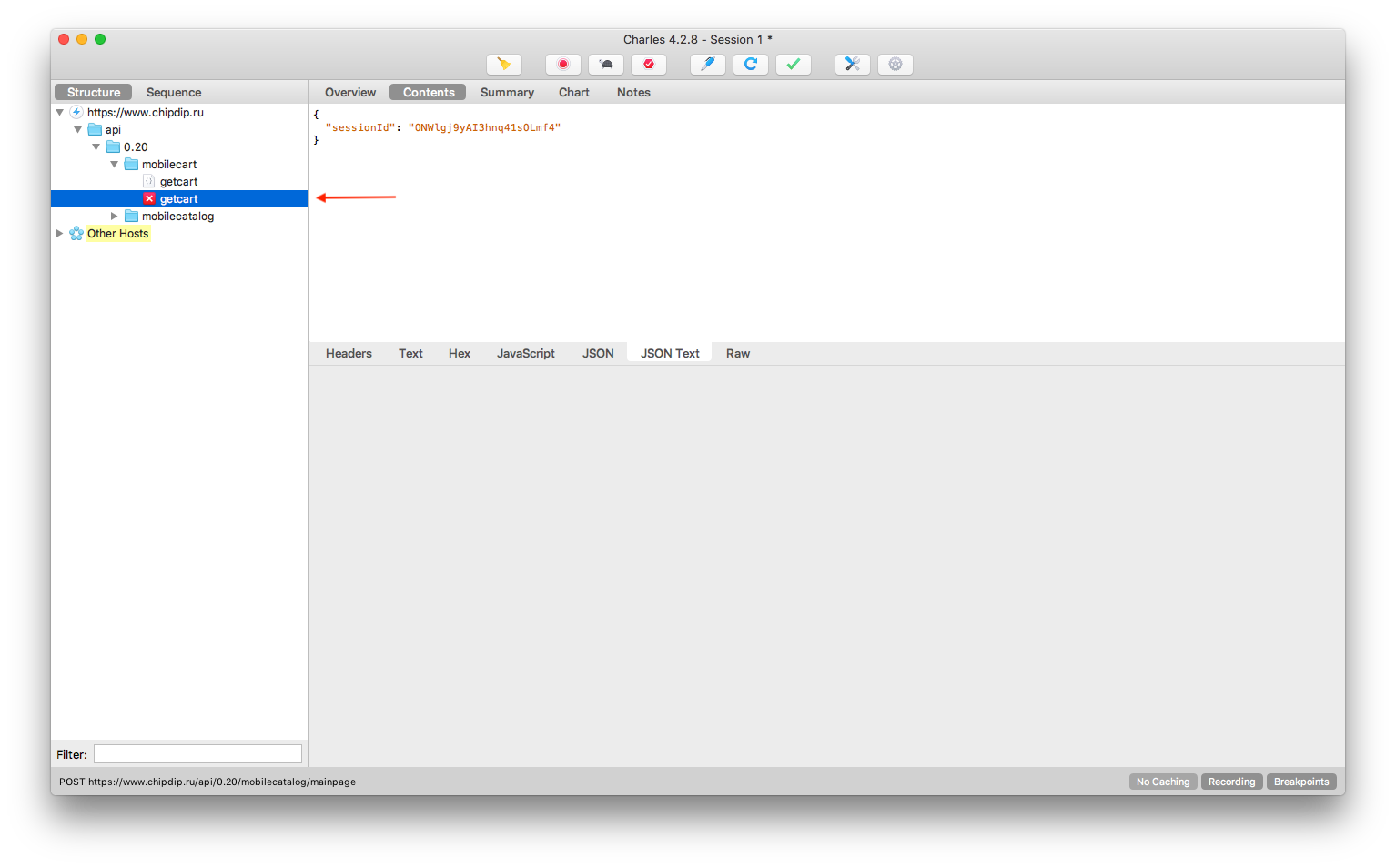
А что, если нажать не Execute?

* Кнопка **Abort** обрывает запрос или ответ. Если Превысить время ожидание установленное в приложении, то запрос оборвется по таймауту. И выведется сообщение об ошибке клиенту.

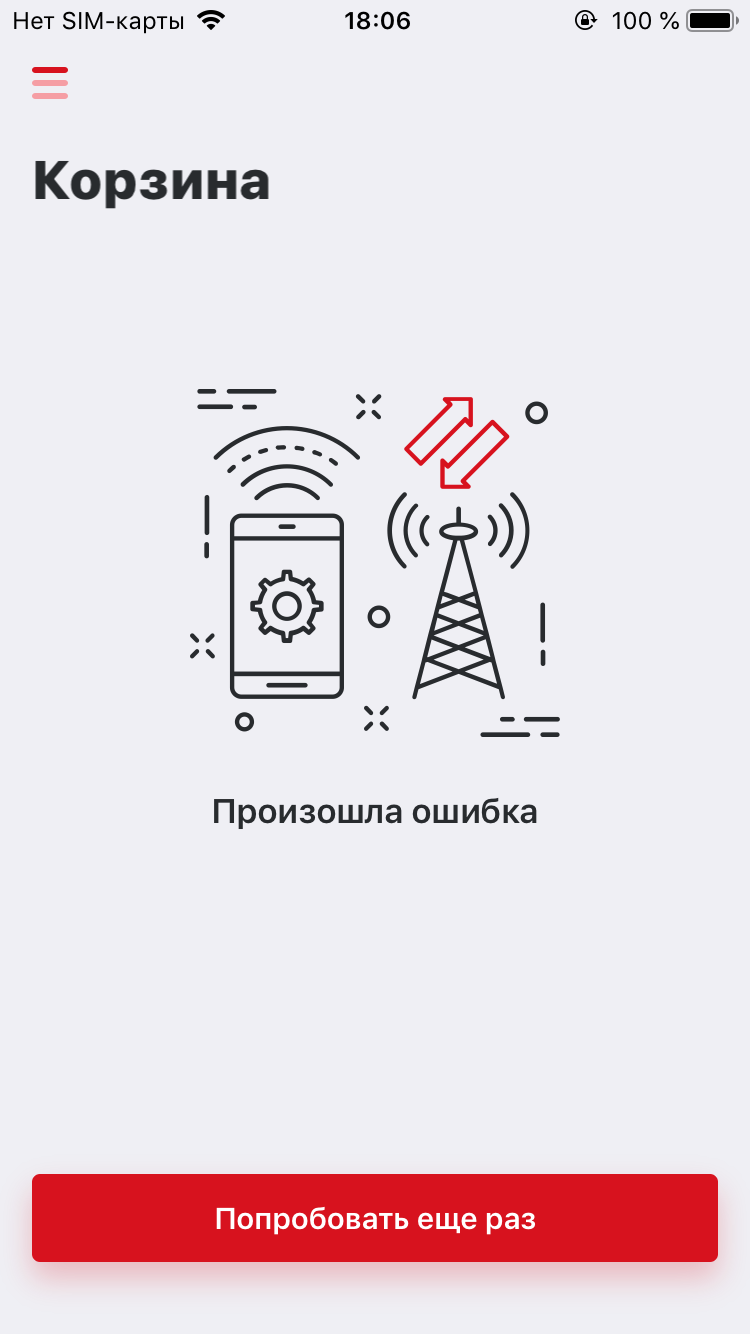
В окне, отображающем запрос, нажимаем Abort:



По красному крестику видим, что запрос пофейлился:



А теперь смотрим в мобильное приложение и видим, как у нас красиво обрабатывается ошибка:



* Кнопка **Cancel** отменяет любые внесенные вами изменения и позволяет запросу или ответу поступать так, как если бы он не был перехвачен.

## 3. Map Local

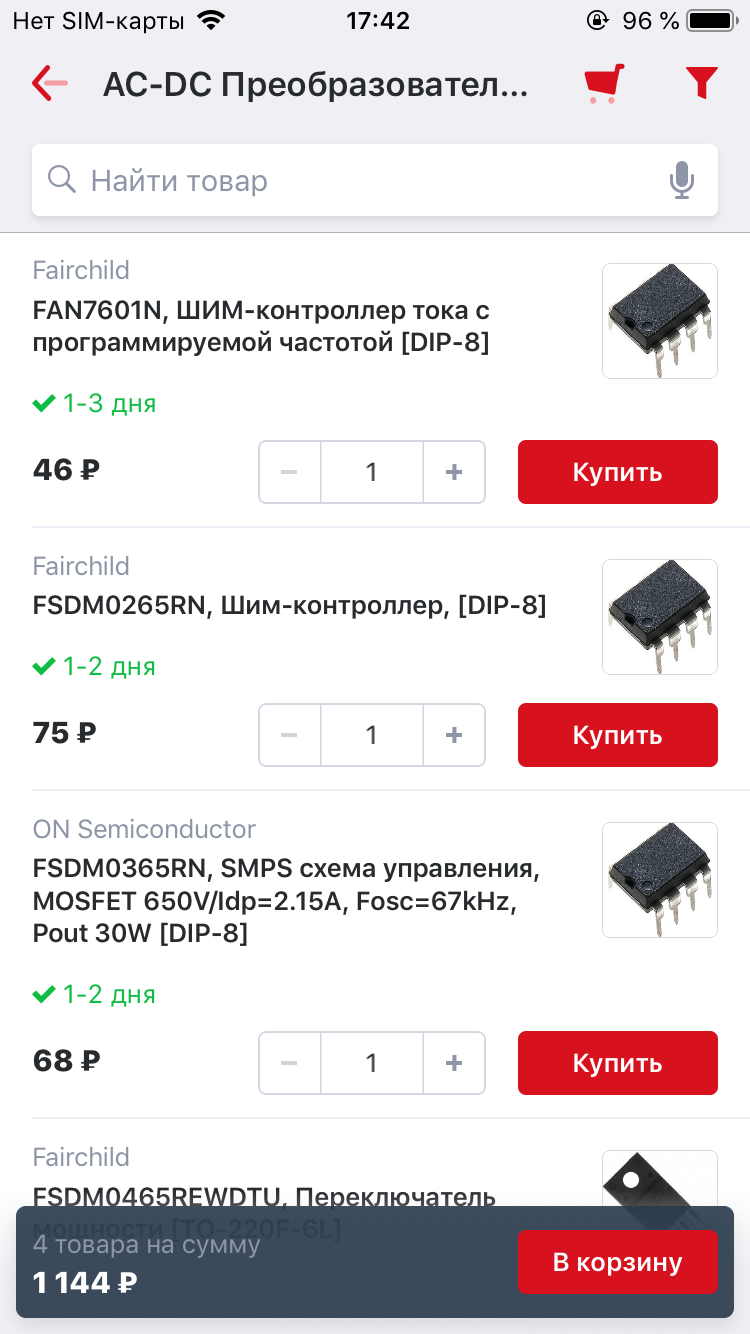
Инструмент Map Local позволяет использовать локальные файлы вместо содержимого ответа на запросы от сервера. Содержимое локального файла возвращается клиенту, как если бы это был обычный удаленный ответ.

Когда выполняется целевой запрос, он проверяет наличие указанного локального файла. Если запрошенный файл найден локально, он возвращается как ответ, как если бы он был загружен с сервера. Если запрошенный файл не найден локально, то запрос пройдет как обычно.

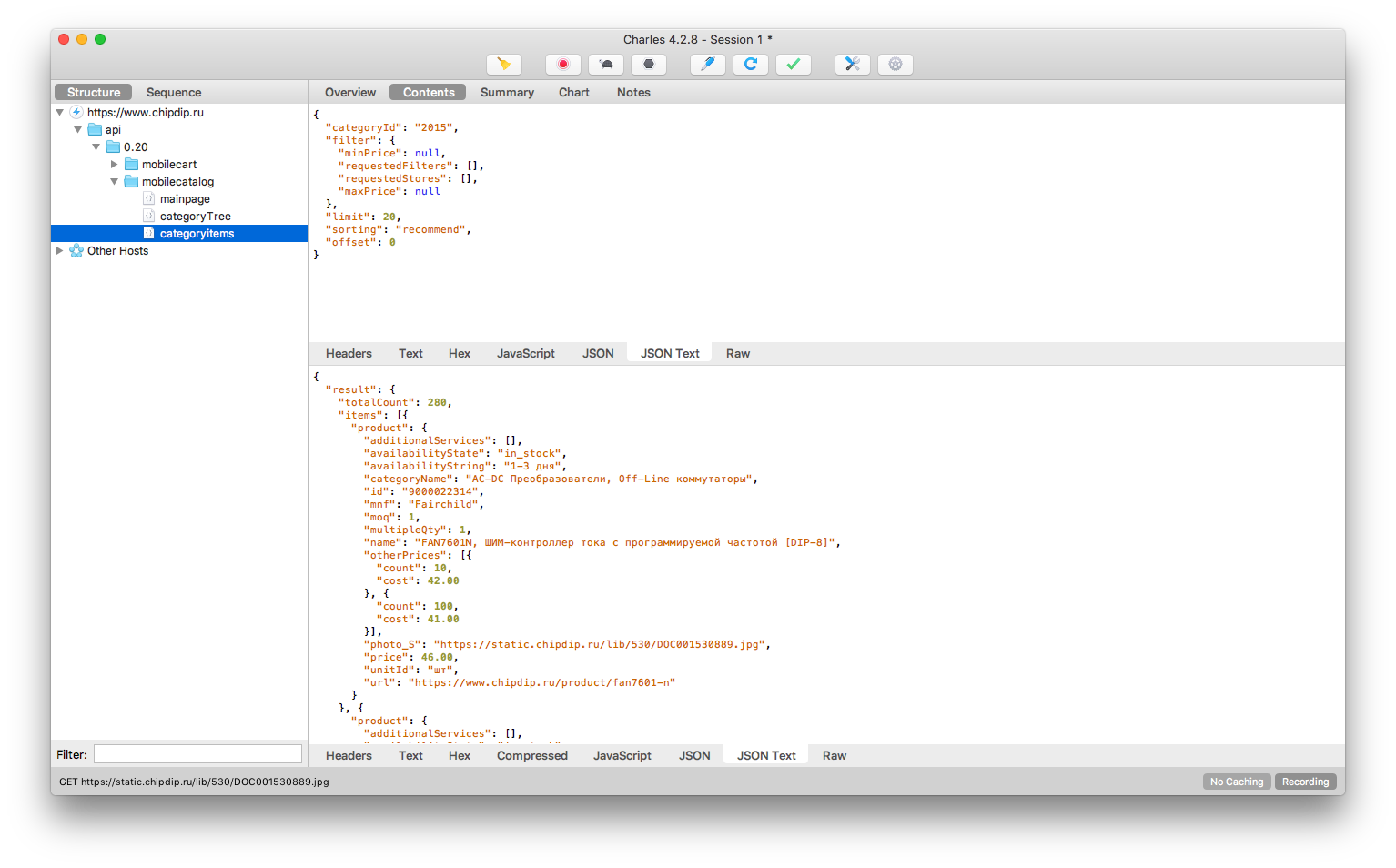
На примере покажу, как в приложении можно подменить большое количество данных, которые к нам приходят от сервера в формате json-текста.

Имеем большой список товаров в каталоге, можно много чего изменить, а удобнее сделать всё сразу, не прописывая правило подмены в Rewrite Tool или Breakpoints.

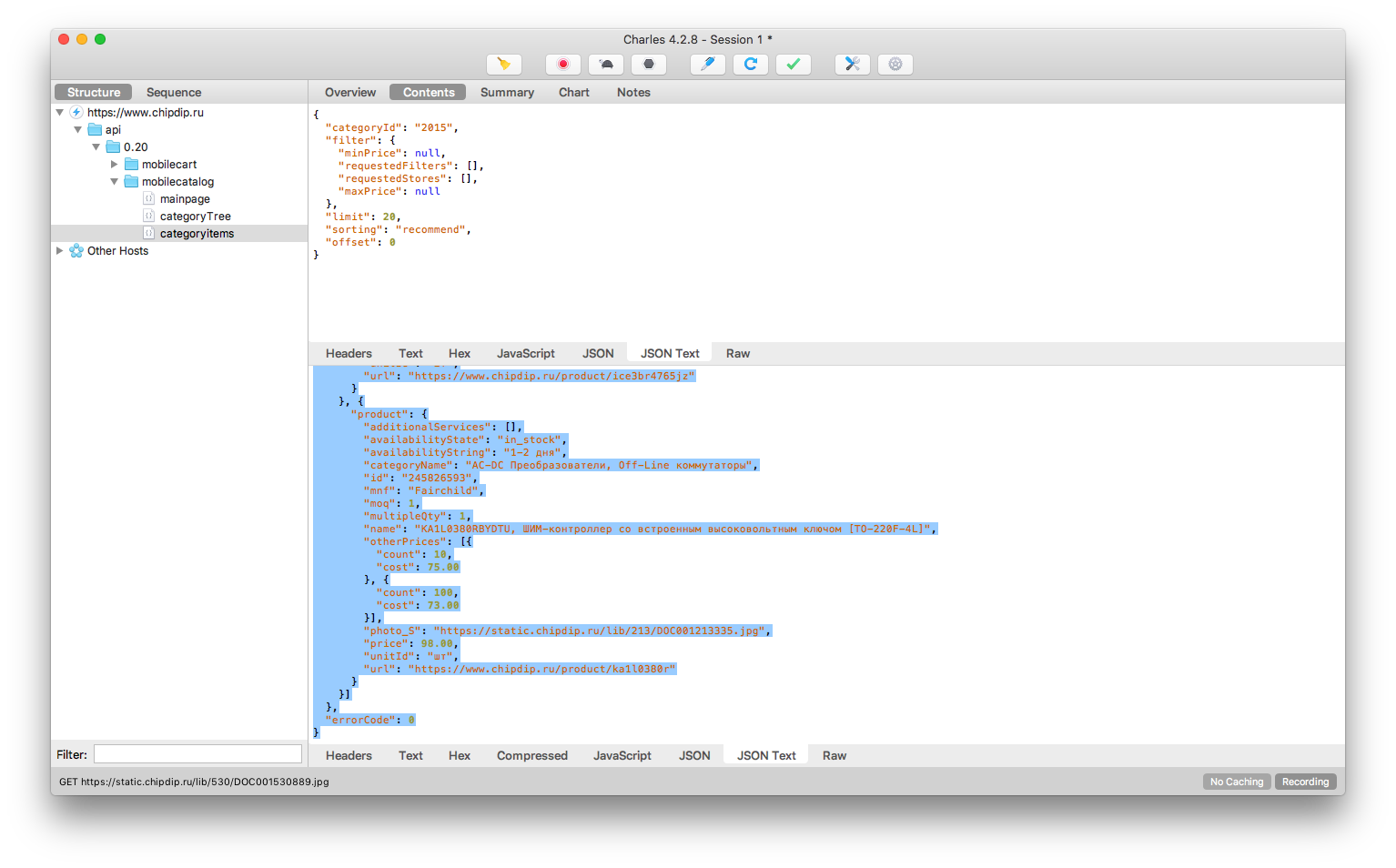
Итак, приложение:



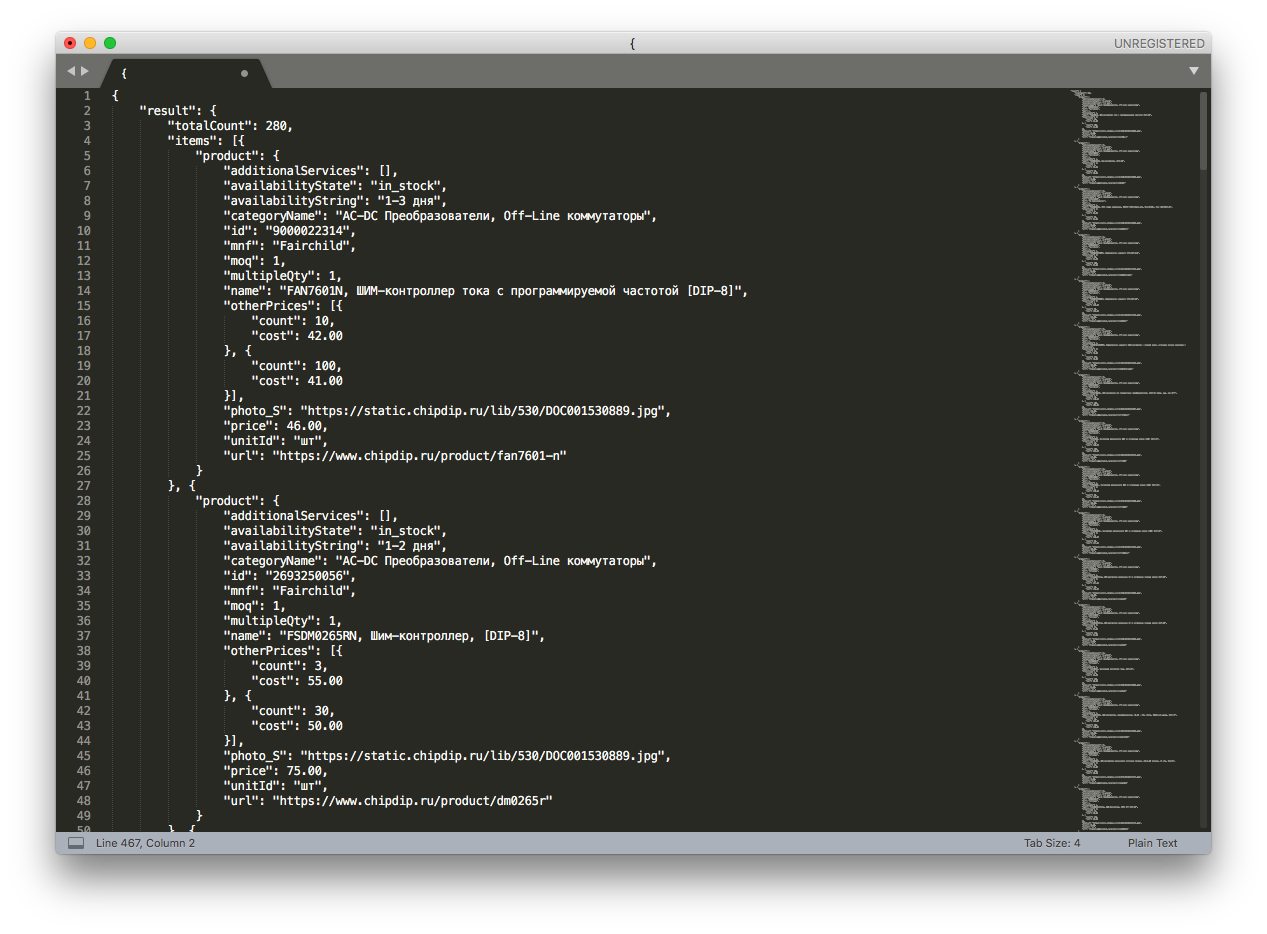
А здесь ответ в Charles’e:



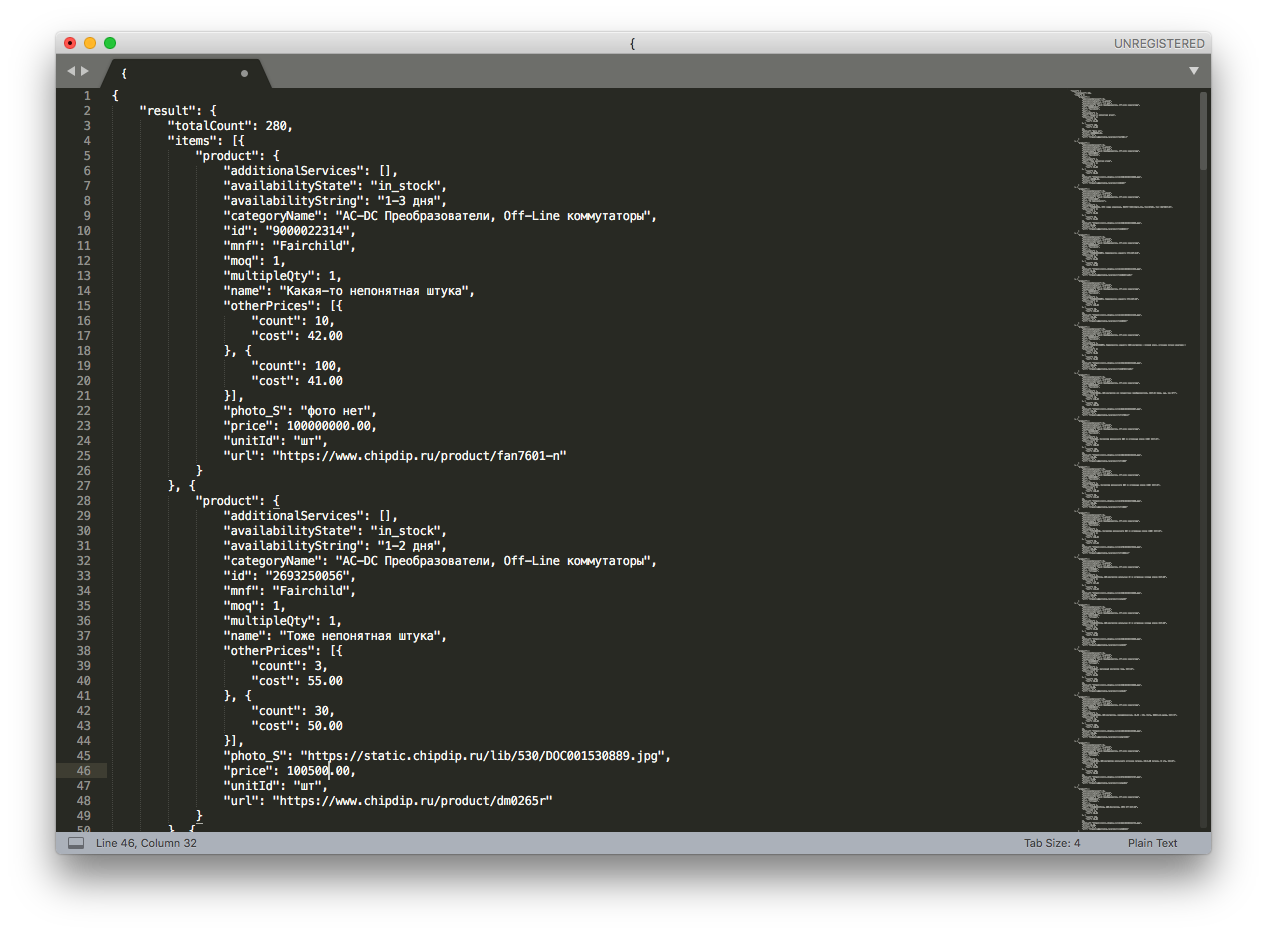
* Так как необходимо изменить ответ от сервера - копируем его. В этом инструменте можно копировать даже в приятном формате json-text:



* Скопированный ответ вставляем в любой текстовый редактор:

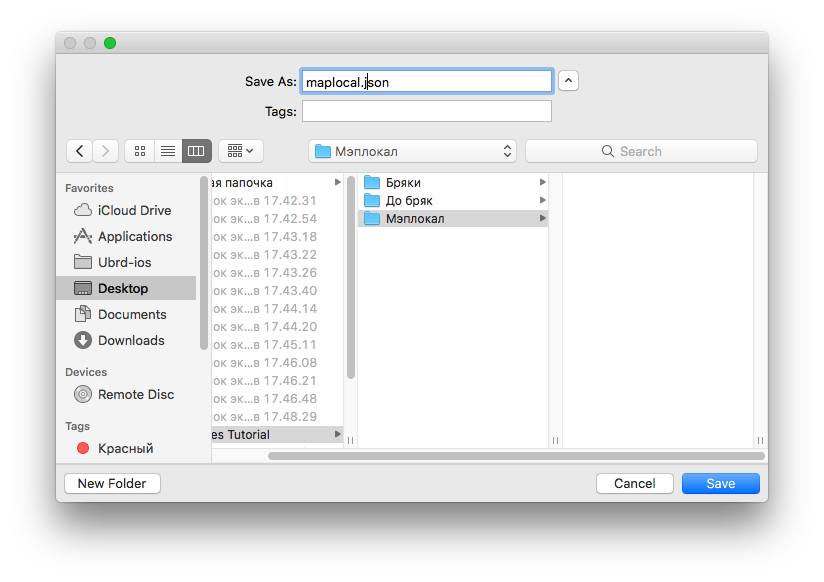


* Далее, подменяем всё, что хотим подменить в данном ответе:

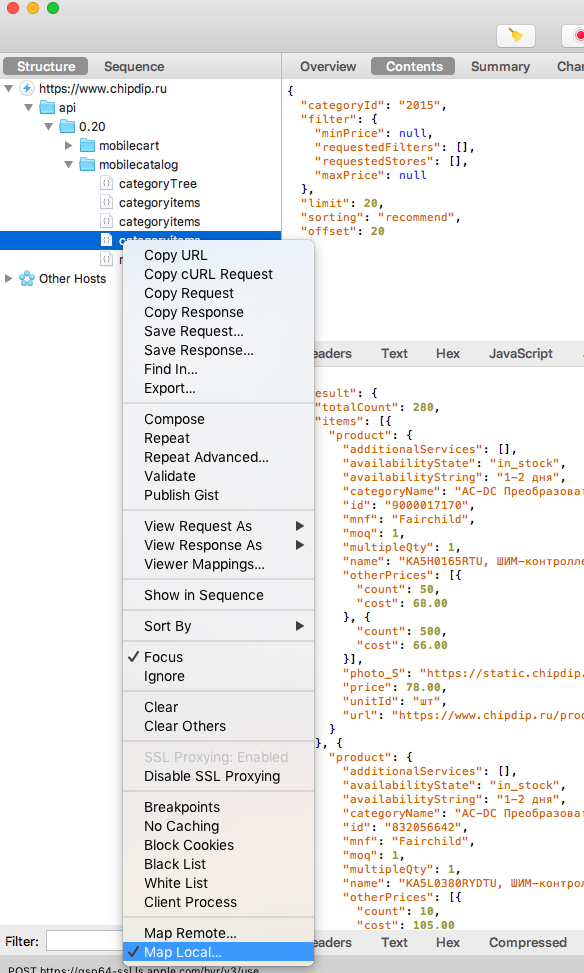


* После этого необходимо сохранить файл с расширением .json или .txt

(так как ответ от сервера у нас приходит именно в этом формате):



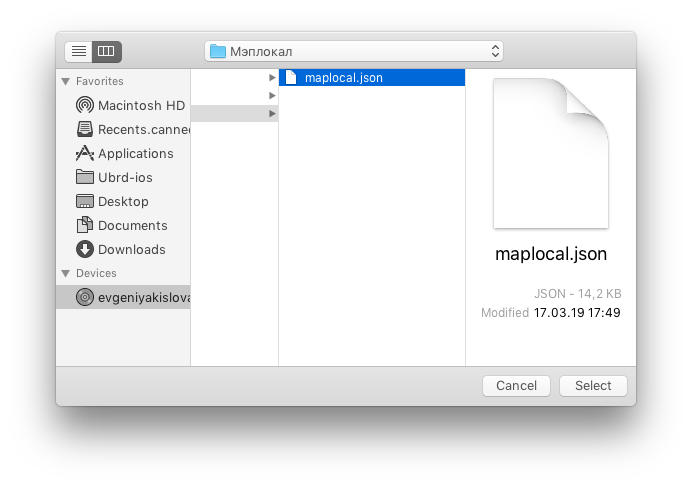
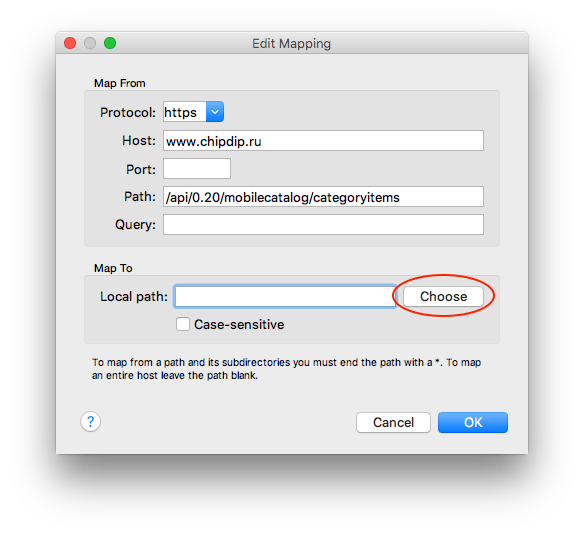
* Возвращаемся к Charles’у. Для этого выделяем правой кнопкой мыши хост, с которым работаем и в выпадающем списке выбираем Map Local:



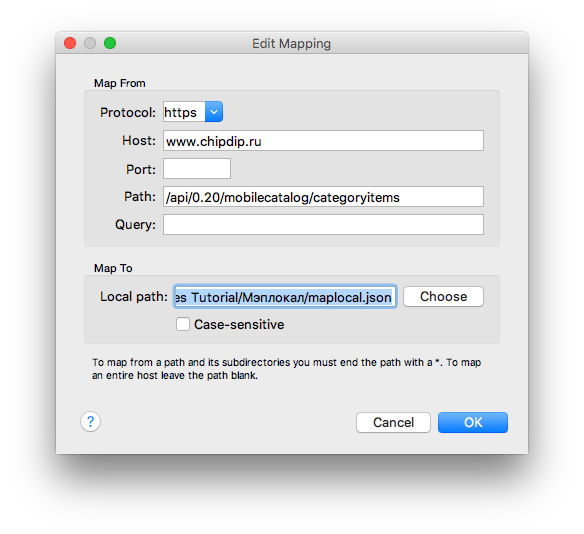
* Появляется окно Edit Mapping с уже заполненной исходной информацией:



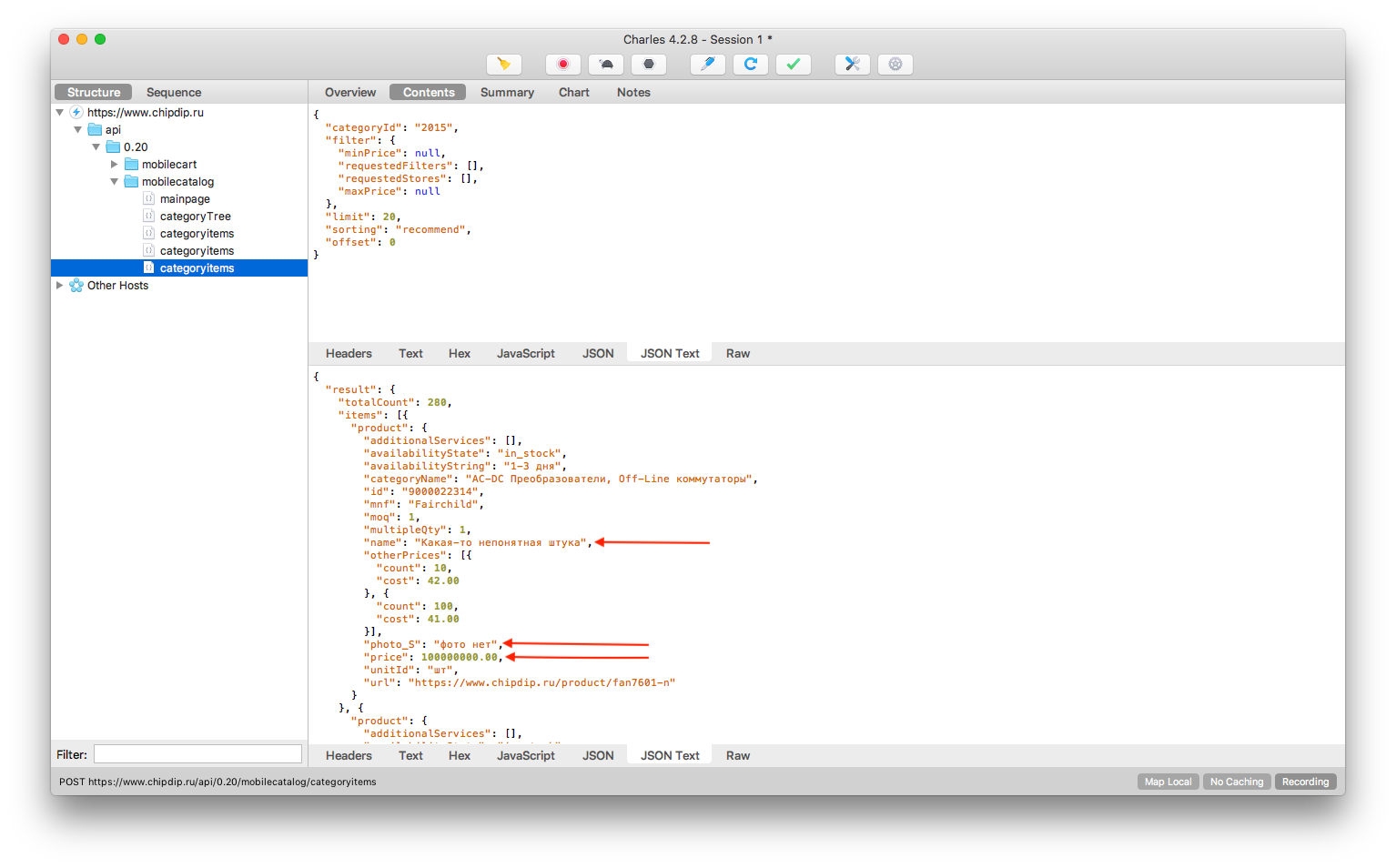
* В данном окне необходимо нажать кнопку Choose, и выбрать ранее сохраненный файл с подменой:



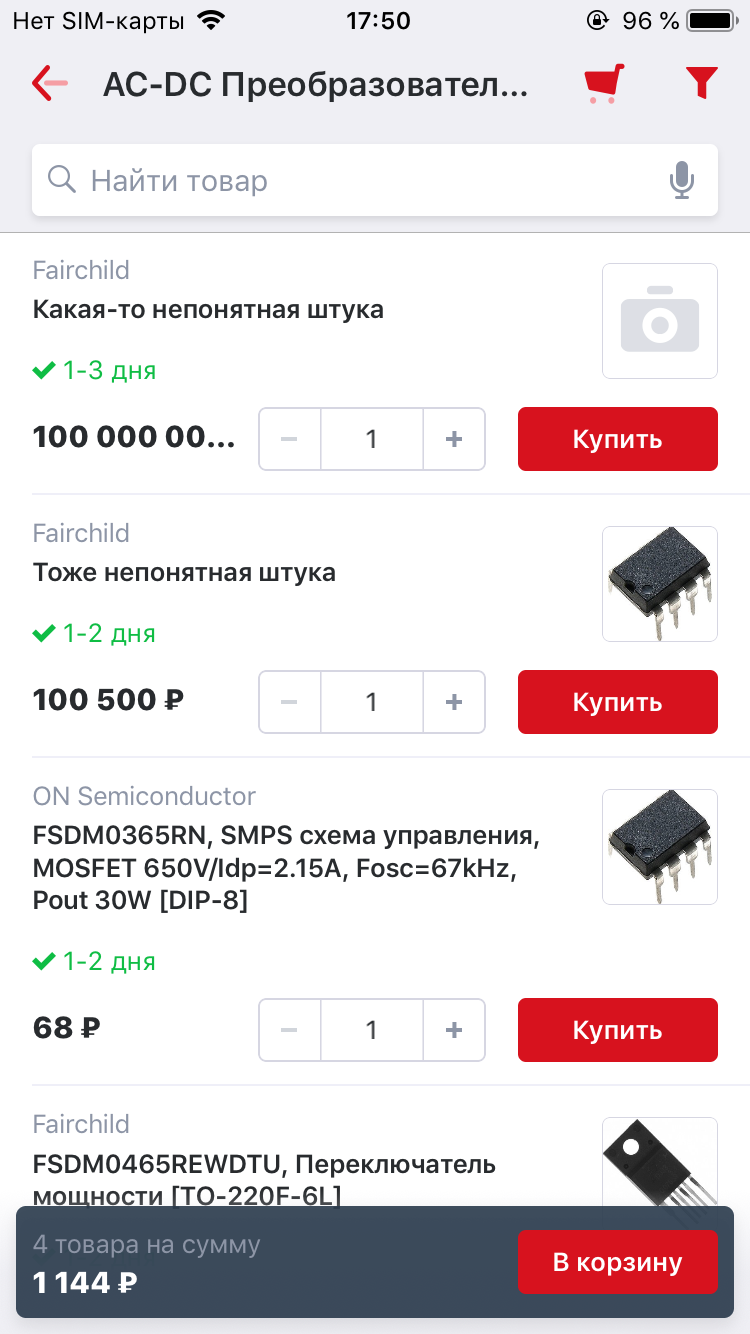
* Получится такое правило Map Local, в котором нажимаем OK:



* Теперь можно увидеть, что мы сделали. Приложение перезапускать не обязательно, повторяем метод, к которому применили Map Local Tool:

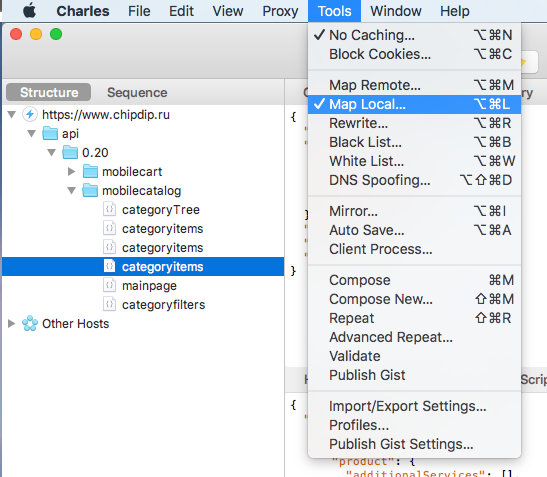


И то, что получилось в приложении:

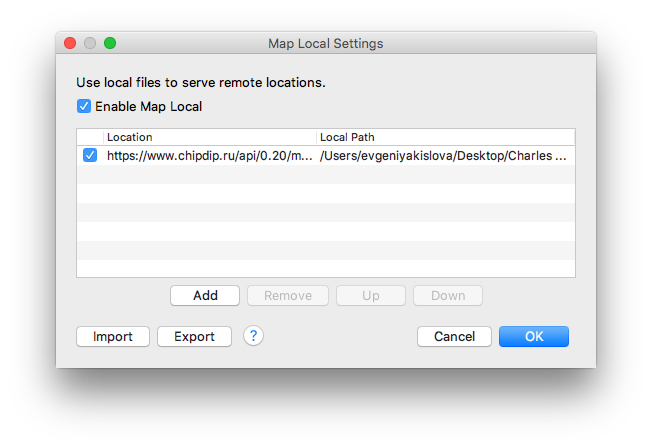


Таким образом можно заменить всё, что вам нужно в одном файле, локально. Работать этот инструмент будет, пока вы его не отключите.

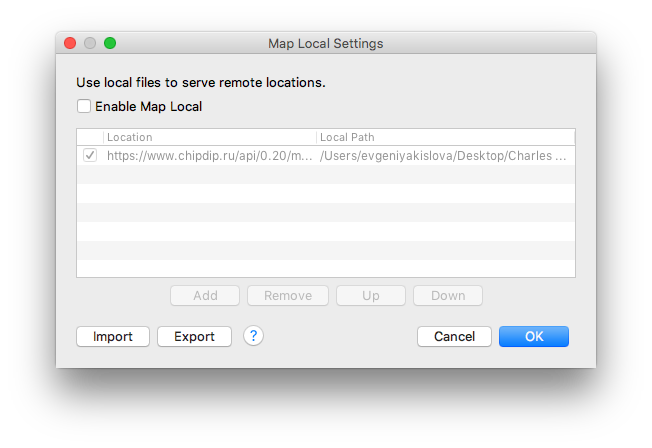
Чтобы отключить этот инструмент в меню необходимо Tools > Map Local:



Появится следующее окно:



В нем нужно убрать галочку из чекбокса Enable Map Local и нажать ОК:



Готово!

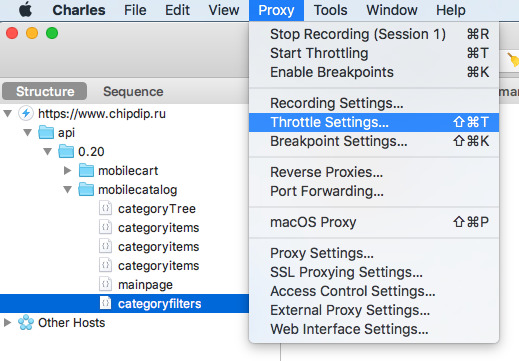
## 4. Throttle Settings

Charles можно использовать для настройки пропускной способности и задержки вашего интернет-соединения. Это позволяет вам моделировать условия модема, используя ваше высокоскоростное соединение.

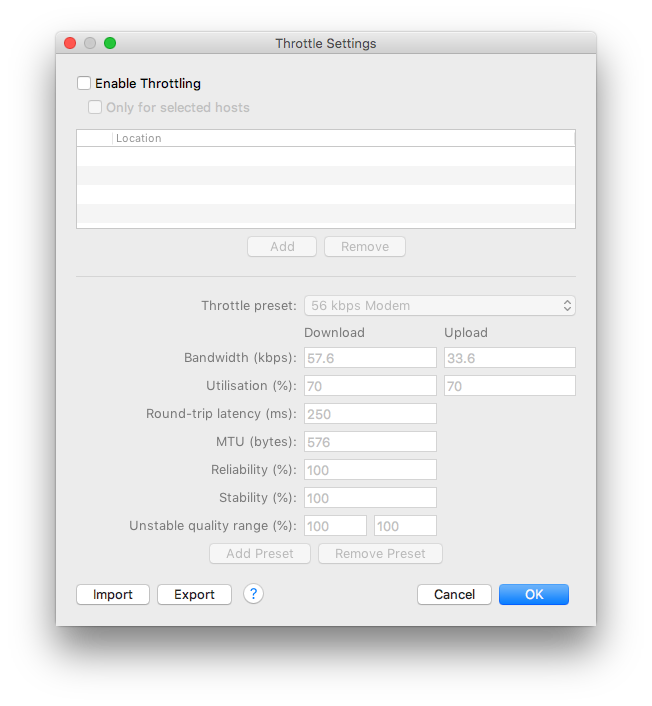
Пропускная способность может быть уменьшена до любых произвольных байтов в секунду. Это позволяет моделировать любую скорость соединения.

Также можно установить задержку ​​на любое произвольное количество миллисекунд.

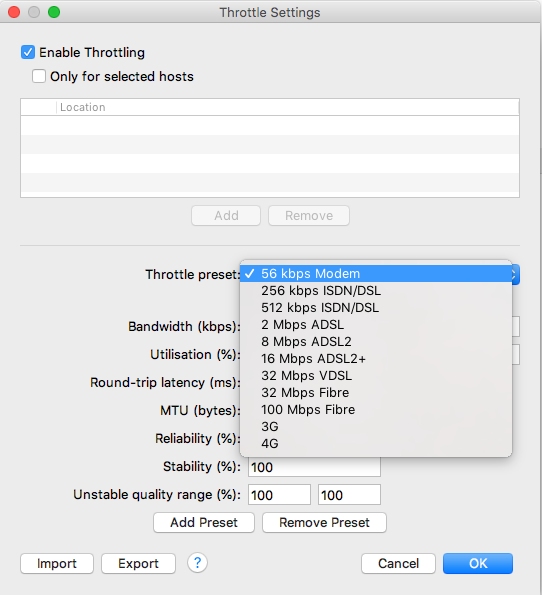
Для этого в меню нужно выбрать Proxy > Throttle Settings:



Откроется окно настроек моделирования скорости сети:

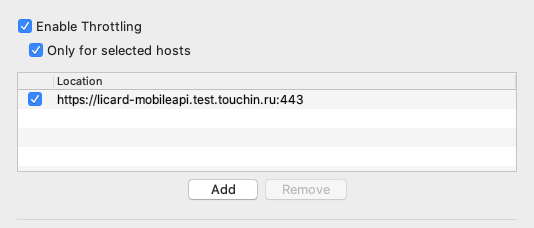


Необходимо поставить галочку в чекбокс напротив Enable Throttling:

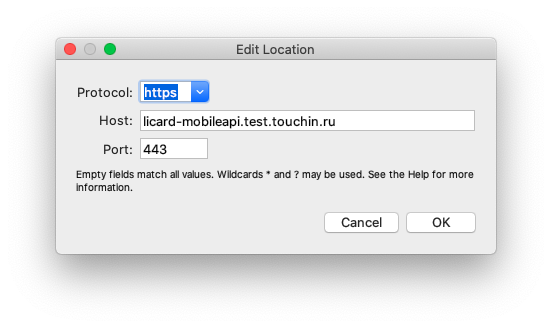


Тут можно выбрать скорость сети (выпадающий список), а также установить какие-либо задержки.

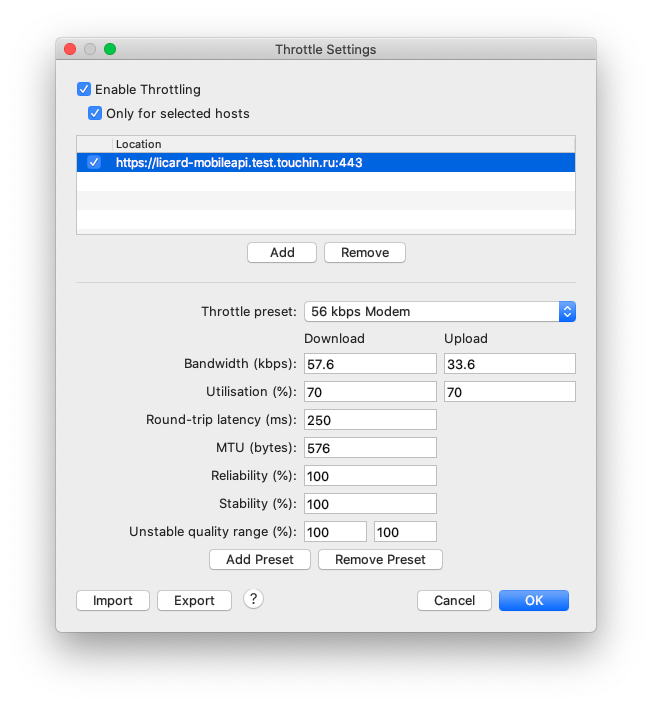
А если поставить галочку напротив “Only for selected hosts” и прописать путь хостам, то можно ставить задержки и менять скорость сети только к выбранным хостам (удобно для выбора только хостов тестируемого приложения)



Нажимаем на Add, после чего появляется окно для прописывания пути:



Нажимаем в нем ОК после его заполнения.



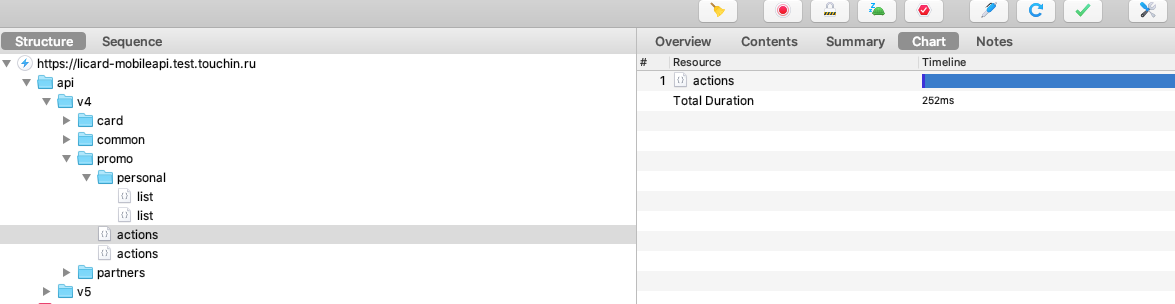
Как только мы нажали ОК в исходном окне настроек троттлинга, в панели инструментов чарльза включится отмеченная ниже кнопка:



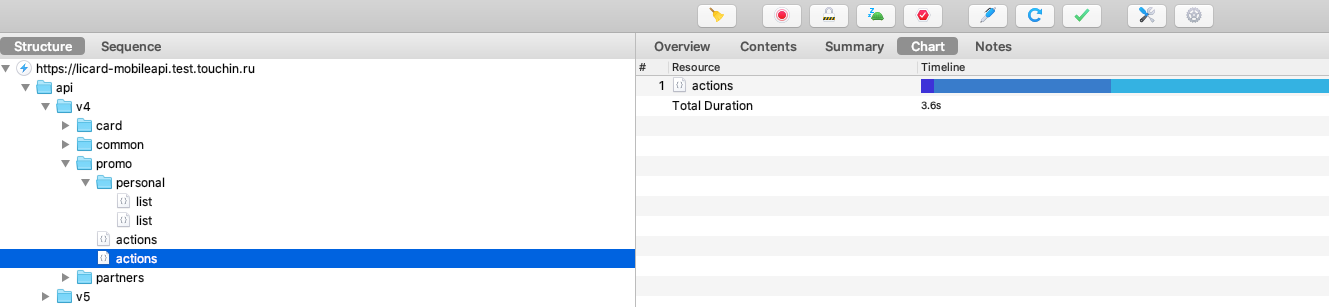
Соответственно, кликнув на неё, можно отключить Throttling Settings.

При включенном троттлинге можно увидеть как скорость выполнения запросов падает.

До включения троттлинга:

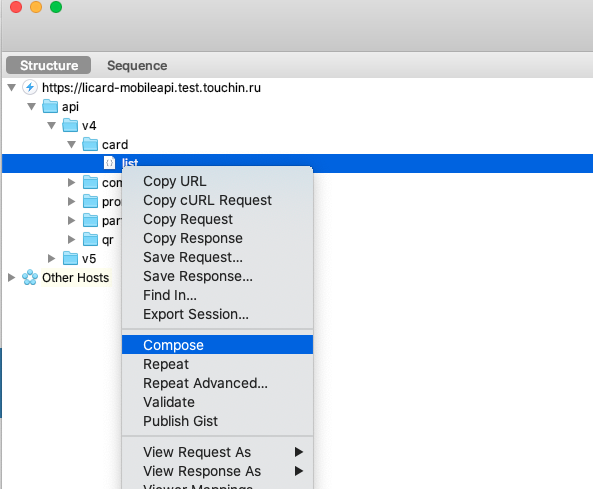


После:

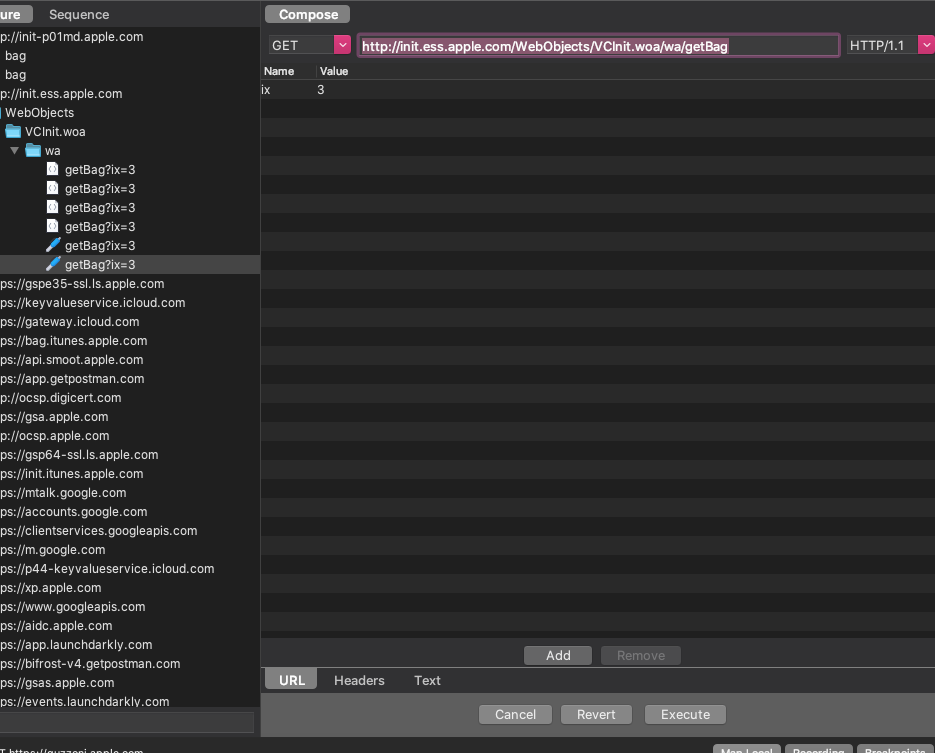


## 5. Compose

В чарльзе есть возможность составлять и отправлять запросы самостоятельно без задействования клиента. Для этого нужно нужно выбрать запрос, который мы хотим отредактировать, и нажать Compose в контекстном меню.



Откроется следующее окно:



В нем есть возможность менять тип запроса, url, query параметры, заголовки и тело запроса. Если мы хотим вернуть параметры запроса к изначальным, нужно нажать **Revert**. Если нужно отправить запрос, нажимаем **Execute**.

Удобный способ для быстрых функциональных проверок без использования сторонних инструментов или ресурсов клиента.

## 6. Repeat

Функция позволяет отослать запрос повторно. Для этого нужно выбрать конкретный запрос и в контекстном меню выбрать Repeat.

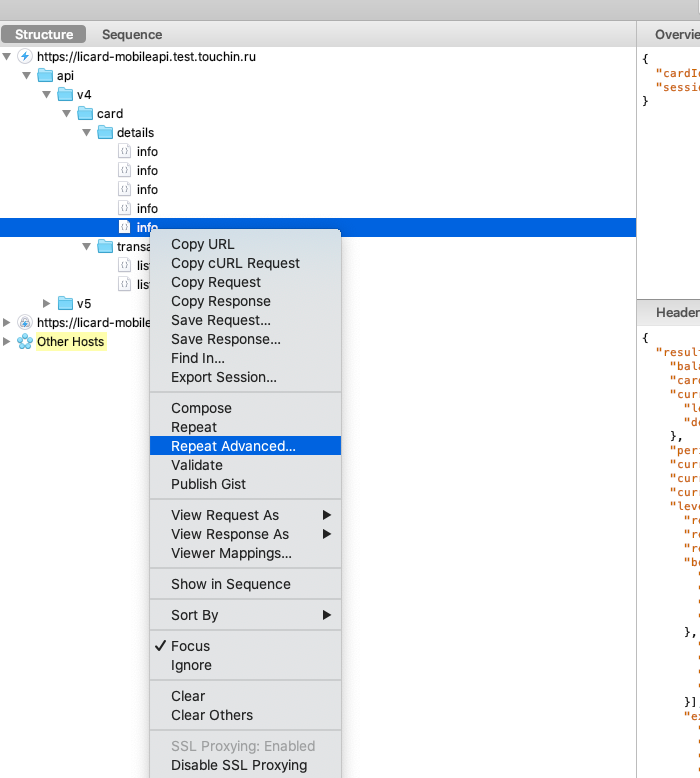


После этого отправляется запрос идентичный выбранному.

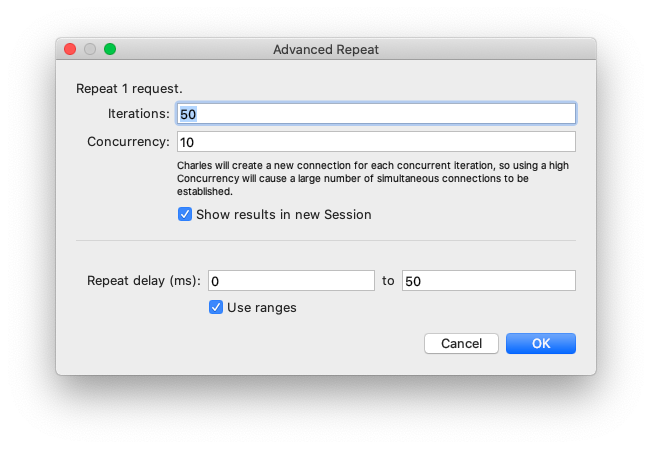
## 7. Repeat Advanced

Фича позволяет отправлять множество запросов с различными временными интервалами. Можно использовать ее как для нагрузочного тестирования, так и для специфических нужд. Например, для последовательного выполнения функционала сервера.

Для использования нужно выбрать запрос и в контекстном меню выбрать **Repeat Advanced**

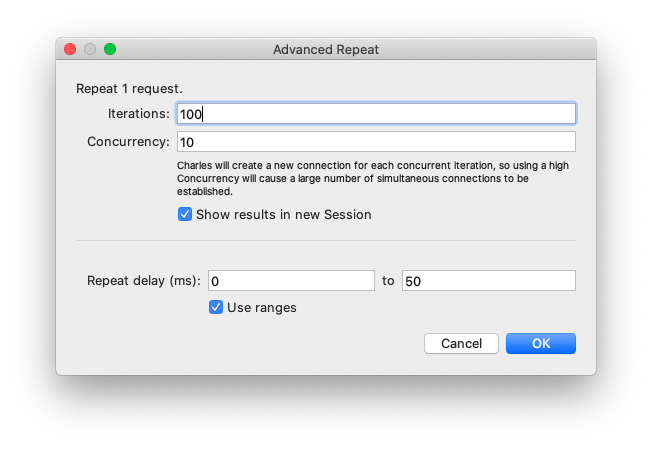


Откроется подобное окно:



* “Iterations” – количество отправляемых запросов;
* “Concurrency” – количество одновременных отправляемых запросов;
* “Repeat delay" – Задержка между отправкой запросов.

Предположим мы хотим отправить 100 запросов и чтобы одновременно отсылалось только 10. Добавим рандомную задержку от 0мс до 50мс. Для удобства поставим чекбокс Show result in new Session чтобы наши запросы отправлялись в новой сессии.



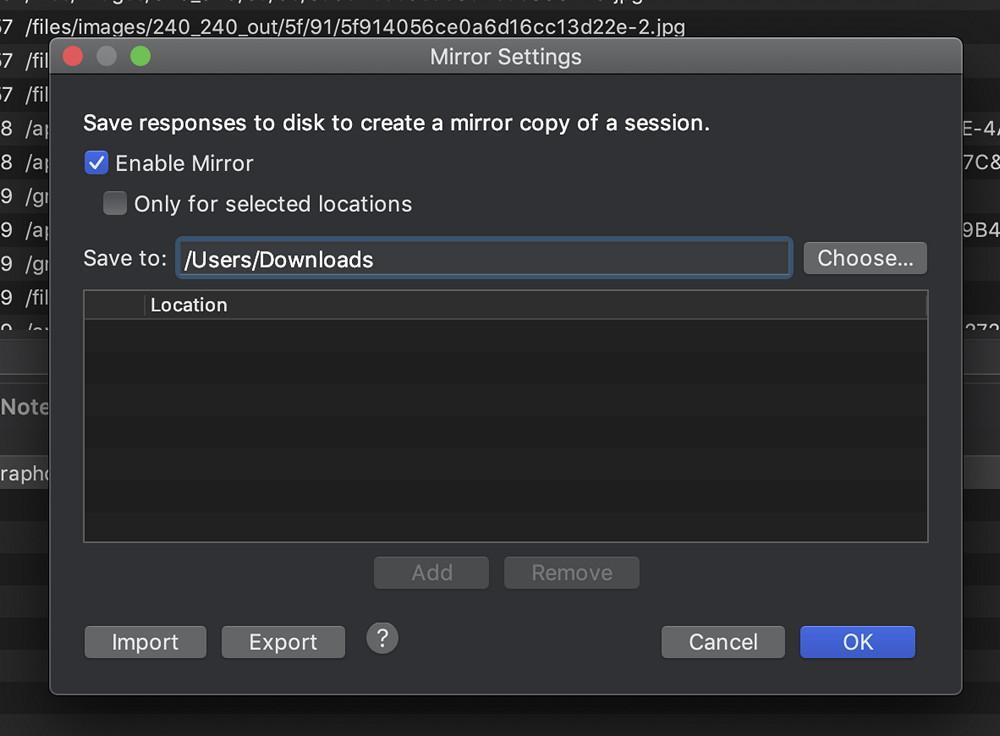
Жмем ОК и видим, как открывается новая сессия и в ней начинают уходить запросы.

## 8. Mirror

**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.

Если два ответа получены для одного и того же URL-адреса, они перезаписывают друг друга, поэтому последний ответ всегда будет сохранен в зеркале.

Активировать функцию можно так: **Tools** → **Mirror**

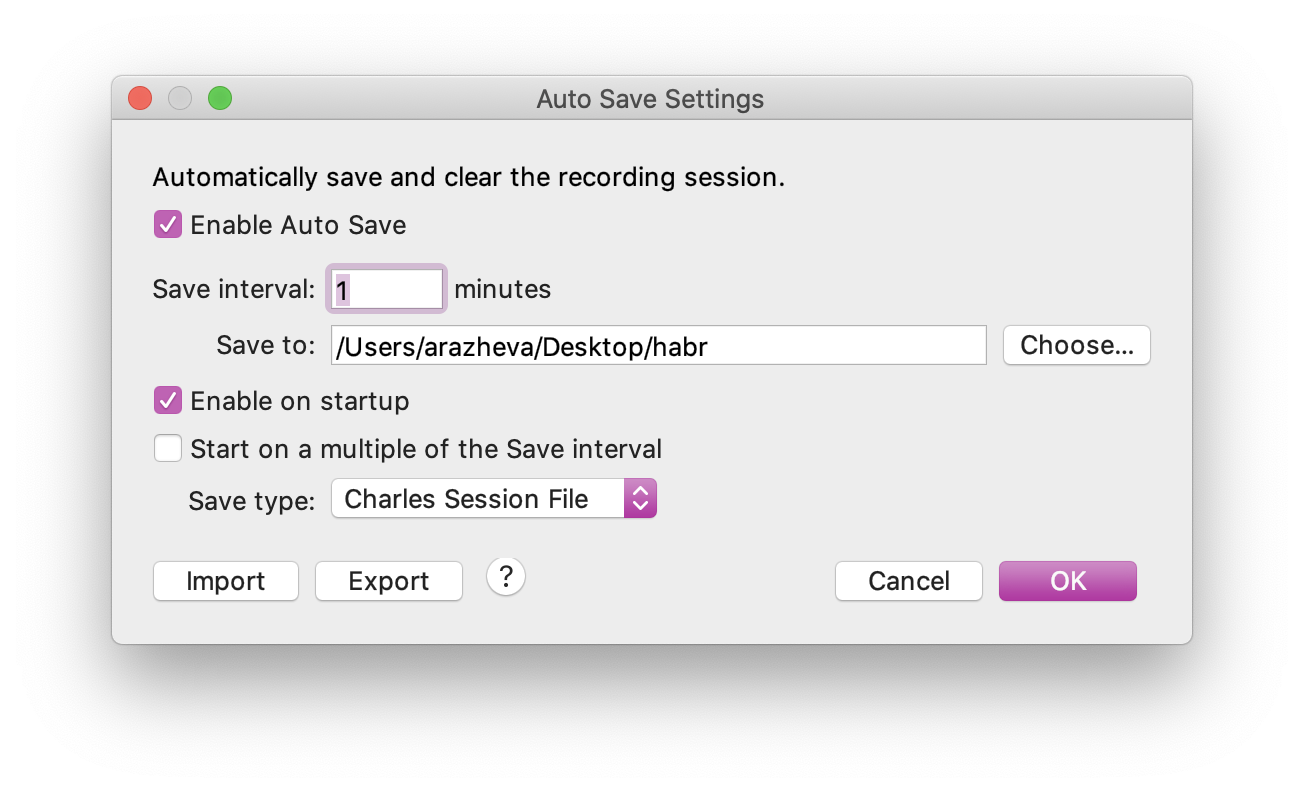


## 9. Auto Save

Инструмент автосохранения автоматически сохраняет и очищает сеанс записи через заданные промежутки времени.

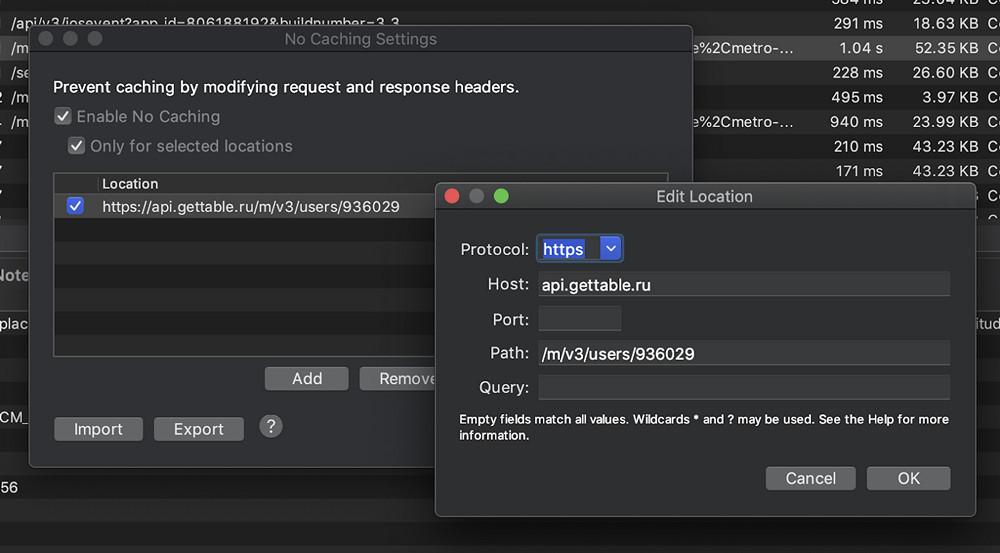
Это полезно, если вы предоставляете Чарльзу возможность отслеживать сетевую активность в течение длительного периода времени и хотите разбить запись на управляемые блоки или избежать условий нехватки памяти, которые могут возникнуть из-за больших объемов данных.

Переходим в Tools -> Auto Save, выберем интервал сохранения сессий и путь для сохранения chls-файлов. Теперь через каждую минуту сессия будет сохраняться локально, а затем очищаться в Charles без дополнительных действий с вашей стороны.



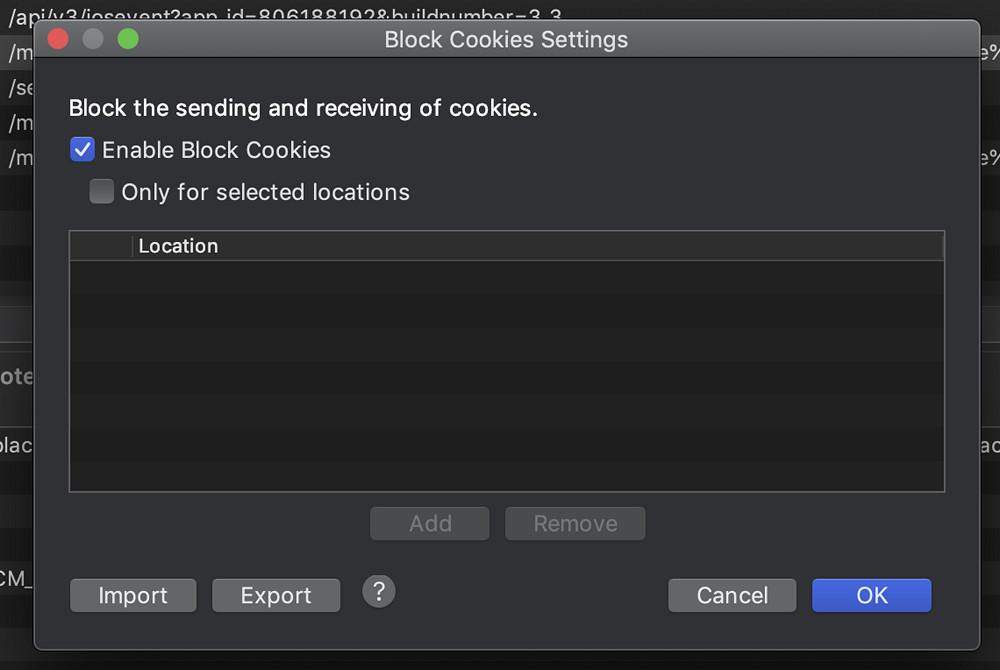
## 10. No Caching

Инструмент No Caching предотвращает кэширование, манипулируя заголовками HTTP, которые управляют кэшированием ответов. Заголовки If-Modified-Since и If-None-Match удаляются из запросов, добавляются Pragma: no-cache и Cache-control: no-cache. Заголовки Expires, Last-Modified и ETag удаляются из ответов и добавляются Expires: 0 и Cache-Control: no-cache.



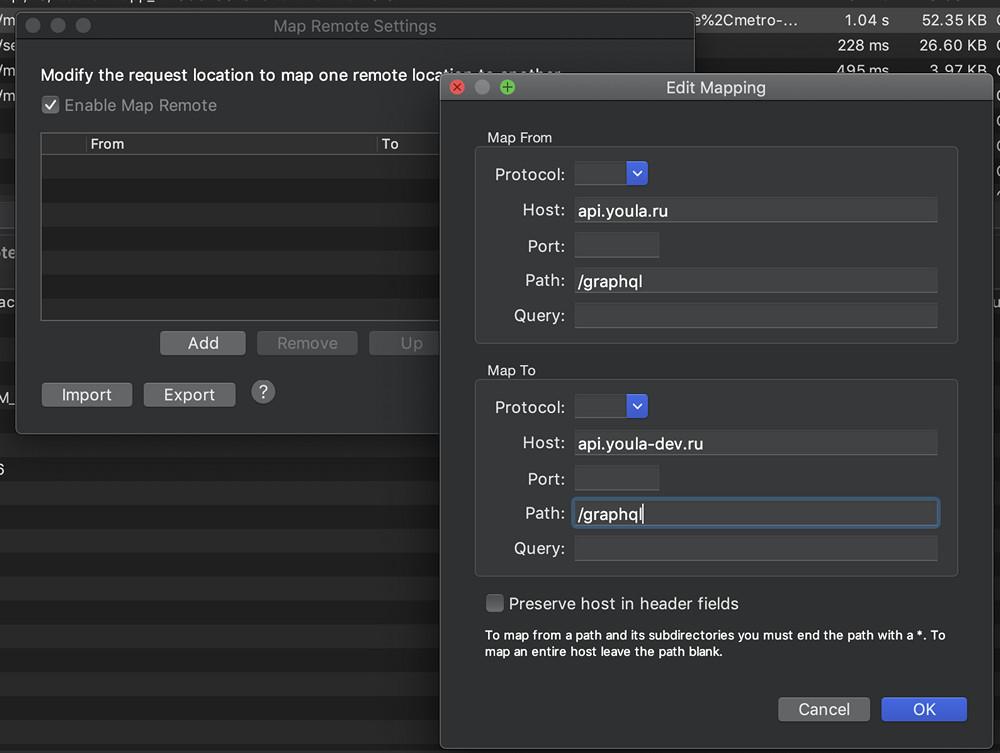
## 11. Block Cookies

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. В примере ниже включено удаление Cookie для всех запросов.



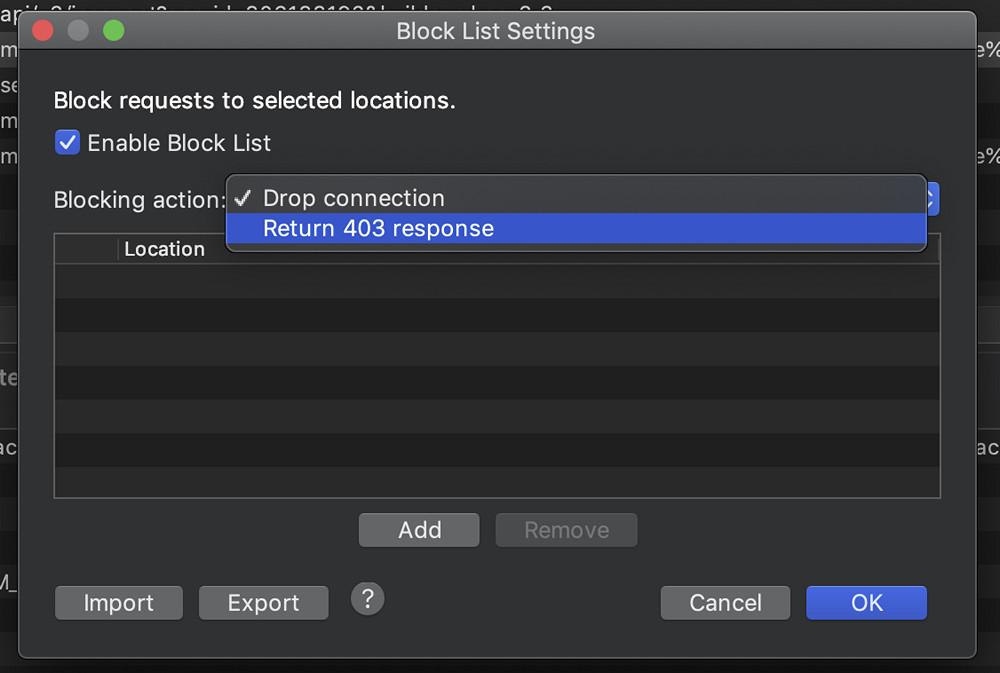
## 12. Map Remote

Map Remote — позволяет переадресовать запросы с одного URL («Map From») на другой («Map To»). Подменяет хост, путь целиком или только параметры в зависимости от вашей задачи. В примере ниже подменен запрос с prod-сервера на dev-сервер.



## 13. Block List

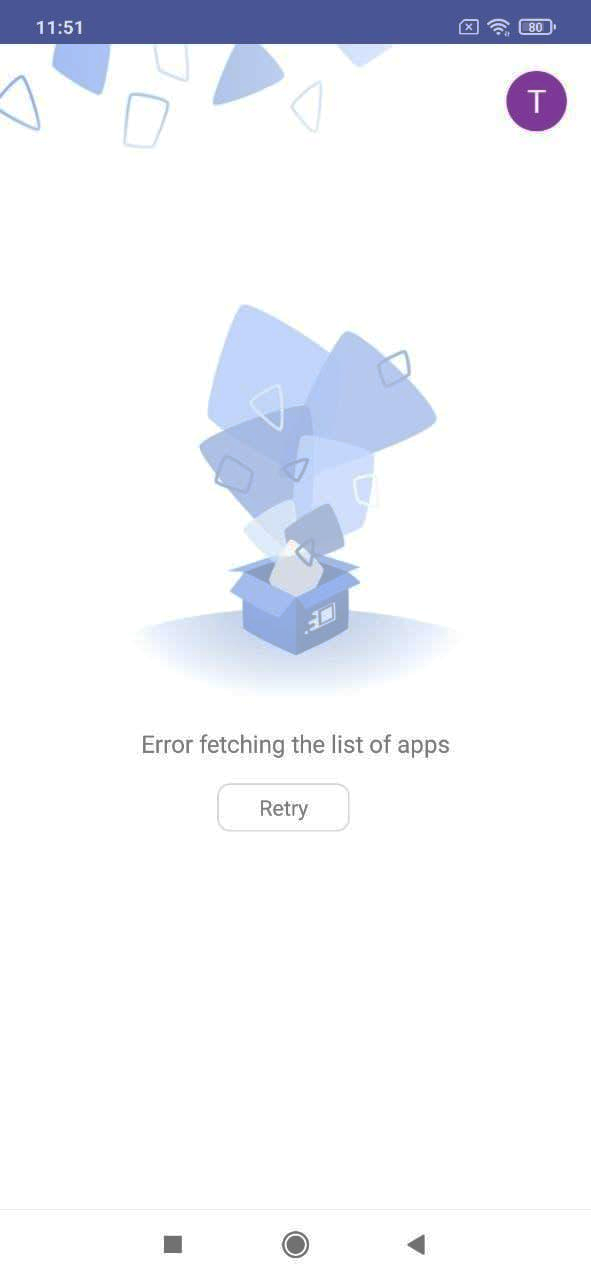
Block List — позволяет блокировать определенные доменные имена. Когда веб-браузер попытается запросить любую страницу из заблокированного доменного имени, она заблокируется. Можно выбрать либо «Drop connection», либо возврат 403 ошибки.



# Полезные советы

## 1. App Tester не загружается на Android с Charles

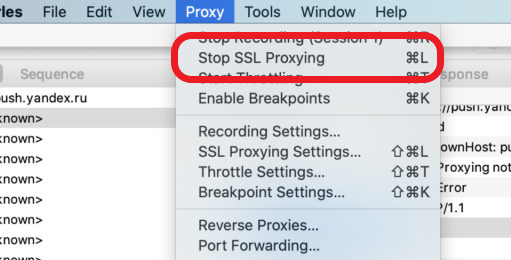
Во время установки сборок из Firebase AppDistribution с помощью приложения App Tester на Android может возникнуть ошибка подгрузки данных о сборках:



Для того, чтобы успешно загрузить данные о доступных сборках, следует остановить SSL-проксирование в Charles’e путем отжатия замочка в панели инструментов на Windows:

|  | → |  |
| --- | --- | --- |

А на macOS зайти в меню Proxy → Stop SSL Proxying:



Также можно воспользоваться сочетанием горячих клавиш:

* Windows: “Ctrl + L”;
* macOS: “⌘+L”

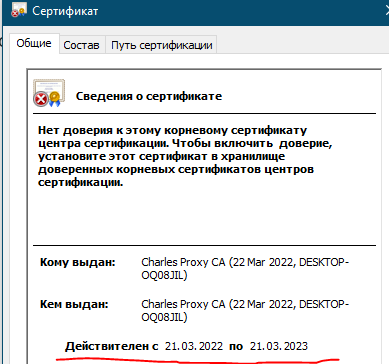
Затем в App Tester нажмите кнопку “Retry”, чтобы загрузить список приложений для тестирования:

|  | → |  |
| --- | --- | --- |

Когда необходимое для тестов МП будет загружено, нажмите замочек снова, чтобы продолжить снифать зашифрованный трафик с помощью Charles.

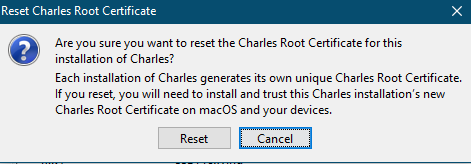
## 2. Устанавливается просроченный корневой сертификат Charles Proxy

1. Проверить не истек ли срок действия сертификата. Help - SSL Proxying - Install Charles Root Certificate



Если сертификат просрочен и при установке нового устанавливается уже просроченный сертификат необходимо сбросить сертификат.

1. Help -> SSL Proxying -> Reset Charles Root Certificate… -> Reset



## 3. Проблема когда трафик идет не через впн (Мас).

Изменение приоритета сетевых портов, используемых компьютером Mac.

Если Вы подключаетесь к интернету или другой сети несколькими способами (например, через Wi-Fi или Ethernet), можно изменить приоритет сетевых подключений, используемых компьютером.

Если у Вас несколько активных подключений, компьютер попытается использовать первое подключение в списке, а затем — следующие подключения по порядку.

Нельзя изменить порядок подключений к сети VPN, так как их приоритет уже выше, чем у подключений к другим сетям (не VPN).

1. На Mac выберите меню Apple  > «Системные настройки», затем нажмите «Сеть» .
2. Нажмите всплывающее меню действий , затем выберите «Выбрать порядок служб».
3. Порядок служб меняется перетягиванием.
4. Перенести vpn конфигурацию поверх всех
5. Нажмите «OK», затем нажмите «Применить», чтобы активировать новые настройки.

Можно прописать маршрут вручную:

Нужно чтобы коннекты к моему серверу, расположенному по адресу (пример: 91.208.121.58) шли через VPN (пример: 10.0.0.1) . Для этого нам нужно построить сетевой маршрут. Воспользуемся штатной unix-командой route:

**sudo route -n add 91.208.121.58/32 10.0.0.1**