1. **Breakpoint** – позволяет поставить на паузу запрос или ответ и редактировать их в режиме реального времени.
2. **Rewrite** – универсальный инструмент подмены, который по заданному правилу ищет совпадения и заменяет на необходимые нам.
3. **Map Local** – подменяет целиком ответ на локальный json.
4. **Map Remote** – автоматическая переадресация пользователя с одного URL-адреса на другой.

Breakpoint — дает возможность приостановить выполнение запроса или ответа на определенном этапе и проанализировать содержимое для того, чтобы можно было внести необходимые изменения и проверить как приложение или сервер отреагируют на внесённые нами изменения. Звучит, вероятно, слегка сложно, но на примере всё будет сильно понятнее.

Функция Rewrite по смыслу похожа на Breakpoint за исключением того, что Rewrite позволяет изменять и перезаписывать запросы и ответы **перед** их отправкой и получением соответственно, а не ВО ВРЕМЯ, как Breakpoint.

Map Remote позволяет перенаправить трафик с одного хоста на другой по настраиваемым правилам.

Вот житейский пример:

Приложение ходит за данными по адресу [https://backend.service.com](https://backend.service.com/).   
Команда бекенд-разработчиков делали задачу на фикс какого-нибудь бага, сделали ветку с изменениями этого бэкенда и хотят протестировать его до слива этой ветки в мастер, на всех. Адрес отдельного домена [https://new-backend.service.com](https://new-backend.service.com/)

Нам нужно протестировать как приложение будет работать с бэкендом с новыми изменениями от разработчиков.

В такой ситуации можно прийти к разработчикам приложения и попросить чтобы они сделали специальную версию приложения, которое будет ходить в новый бэкенд, но для этого нужно отвлекать разработчиков, ждать пока они это сделают.

С помощью Map Remote можно перенаправить трафик приложения без внесения изменений в код.

**Throttling соединения**

Одна из функций Charles - троттлинг соединения, который позволяет симулировать различное поведение Сети.  
Инструмент позволит узнать, как приложение будет работать, например, в ситуации нестабильного интернет-соединения или его внезапного отсутствия.  
  
Включить функцию троттлинга можно в меню Proxy, выбрав там пункт «Throttle Settings».

Давайте посмотрим на все настройки функции.  
  
**Enable Throttling** — это чекбокс включения/отключения троттлинга по приложению. Можно поставить чекбокс на **Only for selected hosts** — это позволит проводить троттлинг только для заданных *url*.  
Их можно добавить или удалить ниже (**Add** и **Remove**).

Далее в меню вы видите набор предустановленных настроек соединений (**Throttle preset**).  
  
**Bandwidth** — это скорость соединения,  
**Utilisation** — процент пропускной способности, которую можно предоставить пользователю в любой момент времени.  
**Round-trip latency** — здесь устанавливается задержка между клиентом и сервером.  
**MTU** — максимальный размер пакета.  
**Reliability**, измеряемая в процентах, устанавливает вероятность, что соединение не удастся. Именно эта кнопка нужна для имитации ненадежных сетевых условий. Кнопка **Stability** задает вероятность нестабильного соединения и снижения качества. Это полезно для моделирования сетей, в которых периодических падает качество связи - например, мобильных.  
  
Throttling может работать даже при установленном в приложении **SSL пиннинге** - достаточно настроить его и убрать **Enable SSL Proxying** с запросов необходимого домена.