**Программное обеспечение, применяемое при тестировании API**

**Postman**



[**Postman**](https://www.getpostman.com/) - REST-клиент, который изначально представлял собой плагин для браузера Chrome, но позже появился так же в виде десктопных версий для Mac и Windows.

Postman позволяет составлять и отправлять запросы на сервер и получать ответы с последующей их проверкой, а так же указывать cookies и заголовки запросов.

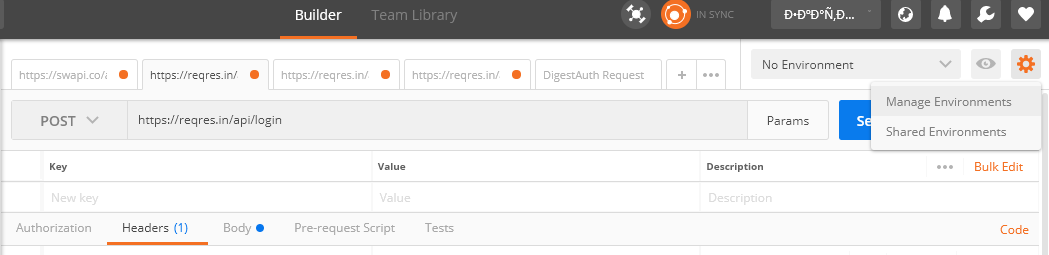
При установке Postman содержит заранее настроенные запросы (коллекция Postman Echo) для более легкого старта тестирования API.

В программе есть встроенный редактор запросов с возможностями кодирования запросов, загрузки из файла и отправки бинарных данных.

**Настройка Postman для работы**

Postman позволяет составлять запросы, используя окружения. Представьте, что у вас есть три тестовых стенда: dev, test и production стенды. У каждого из них отличаются базовые адреса, но сами запросы к ресурсам одинаковы. Для удобства можно использовать так называемые окружения, в которых будут созданы переменные, содержащие базовые адреса каждого из стендов.

Для начала необходимо нажать на иконку  в правом углу, а затем выбрать Manage Environments.



Затем необходимо нажать на кнопку Add, дать название окружению и создать необходимые переменные.

Созданное окружение будет доступно в выпадающем списке справа. Выбрав одно из них, нам будут доступны все переменные этого окружения.

После этого можно использовать переменные, просто указав их имя в двойных фигурных скобках. Таким образом, Вы можете создать переменную base\_url, к примеру, в каждой среде и, переключаясь между средами в процессе тестирования, Вам не нужно будет менять начальные части ссылок.

**Составление запросов**

Составление и редактирование запросов достаточно простое.

Для того, чтобы написать запрос, необходимо выбрать HTTP-метод из выпадающего списка и написать URL. В Postman реализована функциональность ввода параметров URL в виде "Ключ-Значение", галочками можно управлять использованием того или иного параметра в данный момент.

Хэдеры запроса указываются на соответствующей вкладке и их так же можно вводить в виде пар "Ключ-Значение".

Тело запроса формируется на вкладке Body. Здесь можно указать формат ввода параметров тела запроса, например, так же указывать параметры в виде "Ключ-Значение", JSON, XML, отправка файлов или даже обычный текстовый формат.

После составления и отправки запроса необходимо нажать кнопку Send. Каждый выполненный запрос сохраняется на вкладке History, и таким образом можно будет вернуться к выполненному запросу при необходимости.

Ответы сервера отображаются чуть ниже запроса. Здесь так же можно отдельно посмотреть хедеры, тело запроса, куки, время выполнения запроса и код HTTP-ответа.

**SoapUI**

Тестирование SOAP практически всегда подразумевает использование [SoapUI](https://www.soapui.org/). Прочитать про использование этого инструмента можно в разных источниках ([источник 1](http://www.proghouse.ru/programming/20-soapui), [источник 2](http://www.guru99.com/webservice-testing-beginner-guide.html)), но эффективнее всего - ознакомиться с [официальной документацией](https://www.soapui.org/getting-started.html). К тому же, SoapUI позволяет работать не только с SOAP, но и с REST API.

Для начала тестирования web-сервиса в SoapUI необходимо загрузить путь к WSDL сервиса при создании нового проекта. Для примера возьмем Yandex Speller API, который работает на SOAP.

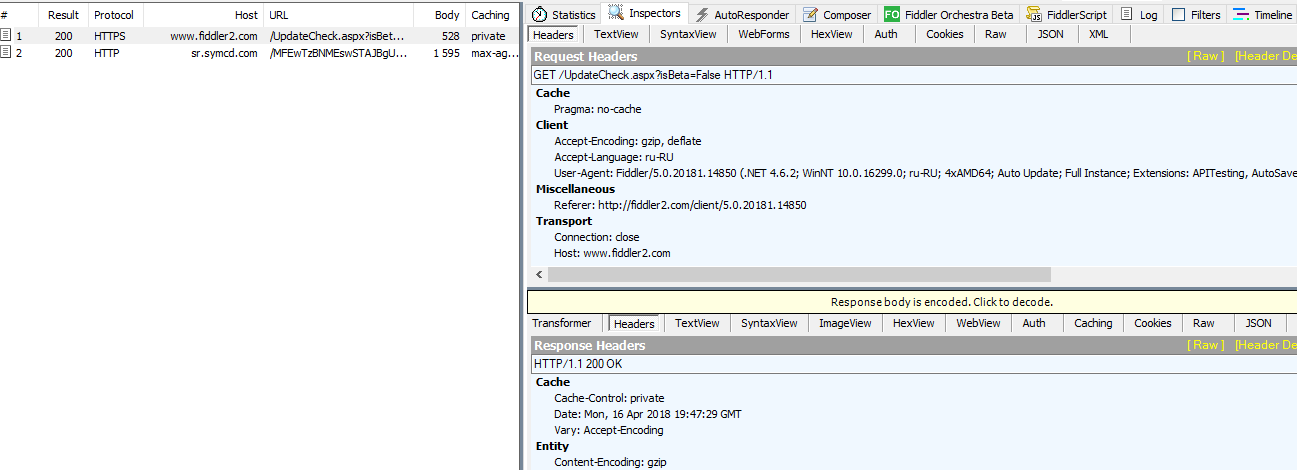
После этого проект будет готов к тестированию в виде загруженных запросов слева. Для отправки запроса необходимо раскрыть список до уровня Request, заполнить параметры в уже подготовленном XML файле запроса и отправить его, нажав на кнопку . Ответ сервера будет отображен в правой части экрана.

SoapUI так же позволяет составлять Test Suites и Test Cases. Тест-комплекты и тест-кейсы позволяют создавать сценарии тестирования API, подготавливать данные для запросов и автоматически проверять полученный ответ на соответствие ожидаемому. На первых порах их можно использовать просто как коллекции запросов. Например, если вы завели дефект и хотите быстро проверить его после фикса, можно выделить отдельный тест-комплект конкретно под запросы-дефекты.

**Fiddler**

[**Fiddler**](http://www.telerik.com/download/fiddler) - прокси, который работает с трафиком между Вашим компьютером и удаленным сервером, и позволяет инспектировать и менять его. Использовать его можно с любым браузером.

Для начала инспекции всех запросов, который отправляет браузер, необходимо просто открыть его и дать команду "начать работу" в браузере параллельно. Вы увидите, что каждый запрос, отправляемый браузером, отображается в окне запросов слева. Их можно просматривать и выбирать запросы, смотреть их заголовки, сохранять их на диск все вместе или по отдельности.



На вкладке **AutoResponder**можно настроить Fiddler так, чтобы он подставлял свой файл вместо ответа сервера. Например, приятель попросил поправить скрипт vasya.js на сайте, но доступа к серверу не дал.  
С Fiddler задача решается просто: сохраняете скрипт на диске, в AutoResponder указываете, что vasya.js нужно брать с диска, а не с сайта, перезагружаете страницу, проверяете – всё с комфортом.

Вкладка **Composer**позволяет составить запрос на сервер вручную, подобно Postman или SoapUI.  
Например, Вы хотите сделать такой же запрос, как только что делали. Для этого можно просто выбрать его слева и нажать кнопку Replay (слева-сверху).  
А если хочется поменять? Нет ничего проще. Выбираем справа Composer и перетаскиваем запрос слева в него. После этого исправляем, что хотим и нажимаем Execute.

Вкладка **Filters**позволяет назначить действия, в зависимости от вида запроса.

В меню **Rules → Automatic Breakpoints** можно включить автоматическое прерывание Fiddler при обработке запроса.  
После этого, если сделать запрос в браузере, подключенном к Fiddler, то его выполнение зависнет, а в левом окошке Fiddler этот запрос будет отмечен специальным значком.  
Если выбрать такой подвисший запрос мышкой, то во вкладке Session Inspector им можно управлять: менять сам запрос и ответ сервера (после Break on Response, когда сервер уже ответил). Данная функция очень полезна для того, чтобы легко отловить нужный запрос, особенно, когда при каждом действии в браузере отправляется много запросов и трудно выявить, какой из них необходим.

**Advanced REST Client**

[**Advanced REST Client**](https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo) - расширение для Chrome для работы с API (конструкция запросов, их демонстрация в удобном виде и другое). Подобно Postman, оно позволяет заполнять данные форм и делать POST, PUT, GET, DELETE запросы, смотреть ответы от сервера и многое другое.  
Запросы можно сохранять на Google Drive и это очень удобно, если вдруг через некоторое время захотите повторить запрос.

В Advanced Rest Client можно задавать множество параметров, как в режиме ввода XML, JSON и др., так и в графическом режиме по каждому параметру поля.