

Современный учебник JavaScript

© Илья Кантор

Сборка от 23 июля 2013 для чтения с устройств

Внимание, эта сборка может быть устаревшей и не соответствовать текущему тексту.

Актуальный онлайн-учебник, с интерактивными примерами, доступен по адресу <http://learn.javascript.ru>.

Вопросы по JavaScript можно задавать в комментариях на сайте или на форуме javascript.ru/forum.

Вопросы по сборке, предложения по её улучшению – можно писать мне, по адресу iliakan@javascript.ru.

Глава: Сундучок с инструментами

В файле находится только одна глава учебника. Это сделано в целях уменьшения размера файла, для удобного чтения с устройств.

Содержание

Полезные расширения Firefox и Chrome

Web Developer (FF, Ch)

DNS Flusher (FF) / DNS Flusher for Chrome

Более удобный Firebug (FF)

JsonView (FF,Ch)

Xml Tree (Ch)

YSlow, PageSpeed Insights (FF, Ch)

AdBlock (FF, Ch)

DownloadHelper (FF)

P.S.

Отладка для Internet Explorer (в Visual Studio)

Установка

Microsoft Script Editor

Microsoft Visual Studio

Галочки для отладки

Способы запуска отладчика

Отладка

Окна

Горячие клавиши

Приложение. Видео.

Скриптуемый отладочный прокси Fiddler

Режимы подключения

Fiddler не под Windows

Операции над запросами

Возможности

Правила

Брейкпойнт на запросе

Отладка HTTPS

Скачать

IE HTTP Analyzer

Возможности

Просмотр и создание HTTP-запросов.

Конструирование фильтров

Дополнительные возможности

Полезные расширения Firefox и Chrome

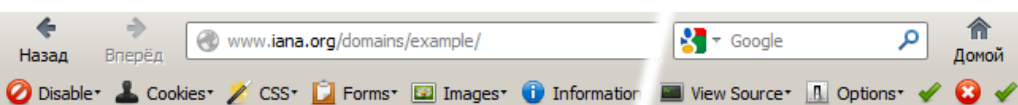
Здесь мы посмотрим ряд полезных расширений, которые, надеюсь, смогут вам пригодиться.

Если какое-то из расширений вас заинтересует — его можно поставить. В Firefox расширения устанавливаются из меню «Инструменты -> Дополнения», в Chrome — через меню «Настройки -> Расширения -> (внизу) Ещё расширения» или напрямую из «магазина» <https://chrome.google.com/webstore/category/home>.

Для расширений других браузеров такого единого способа нет.

Web Developer (FF, Ch)

Это расширение добавляет наверх браузера панель с разнообразными инструментами:



Большинство возможностей, скорее, полезны для верстальщиков, но и для разработчиков кое-что есть.

Например:

- **Disable Cache** — полностью отключает кэш браузера
- **Disable Cookies** — браузер перестанет посылать и принимать Cookie. Впрочем, Firebug это тоже умеет (вкладка Cookie)
- **Disable JavaScript** — нужно редко, но иногда полезно
- **Resize** на нужное разрешение — у разработчиков мониторы большие, а у пользователей разные. Можно добавить своё разрешение.

Иконок там много, поэтому наверняка вы чем-то не будете пользоваться. Лишние иконки можно убрать. В Firefox(Win) это делается так: Правый клик на панель -> Настроить -> Перетащить лишние иконки с панели.

DNS Flusher (FF) / DNS Flusher for Chrome

Это расширение позволяет сбросить кэш DNS одним кликом.

Оно нужно в тех случаях, когда вы меняете адреса в файле `hosts` и хотите, чтобы изменения вступили в действие тут же.

При установке в статусной строке, снизу, появляется кнопочка с

IP-адресом. По клику на ней кэш DNS сбрасывается.

Flushed: 192.0.32.8

Более удобный Firebug (FF)

Здесь собраны расширения, улучшающие работу отладчика Firebug для Firefox.

Fire Rainbow

Подсветка кода для Firebug

FireQuery

Для разработки под jQuery — выводит в разных местах вспомогательную информацию.

Firebug Autocompleter

Включает автодополнение для консоли, когда она в многострочном режиме.

CSS-X-Fire

Позволяет редактировать CSS в Firebug и сохранять изменения, интегрировано с редакторами от JetBrains (IntelliJ IDEA и другие).

JsonView (FF,Ch)

Даёт возможность открыть JSON-документ прямо в браузере.

Обычно браузер не понимает Content-Type: application/json и пытается сохранить JSON-файлы.

Но если в нём стоит это расширение, то он покажет файл в удобном виде, с возможностью навигации:

```

{
  hey: "guy",
  anumber: 243,
  - anobject: {
    whoa: "nuts",
    - anarray: [
      1,
      2,
      "thr<h1>ee"
    ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "明日がある。",
  link: http://jsonview.com,
  notLink: "http://jsonview.com is great"
}

```

Xml Tree (Ch)

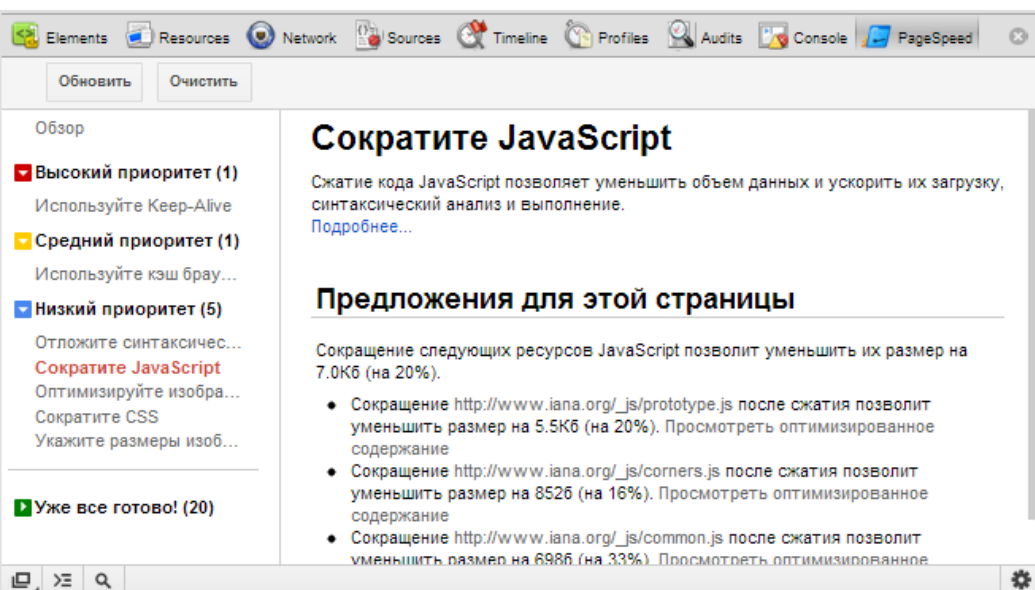
Расширение для просмотра XML для Chrome.

Остальные браузеры умеют это делать «из коробки».

YSlow, PageSpeed Insights (FF, Ch)

Эти два расширения позволяют по-быстрому оценить скорость загрузки страницы, проанализировать массу возможных причин тормозов и получить советы по улучшению производительности.

Они очень похожи, но не идентичны, так что можно поставить оба.



AdBlock (FF, Ch)

Расширение для отключения назойливой рекламы и баннеров.

Режет не всё, но многое. Да, оно не для разработки, но настолько

полезное, что я не смог удержаться и опубликовал его здесь.

Кстати, насчёт разработки... Бывает так, что AdBlock прячет рекламу на сайтах, которые мы разрабатываем. К хорошему быстро привыкаешь, и если юзер будет слать вам ошибки на страницах (а у вас их нет) — проверьте! Может быть, стоит отключить AdBlock для конкретной страницы и посмотреть без него? Исключения можно поставить в настройках расширения.

DownloadHelper (FF)

Расширение для скачивания видео из Youtube, Blip TV и других хостингов.

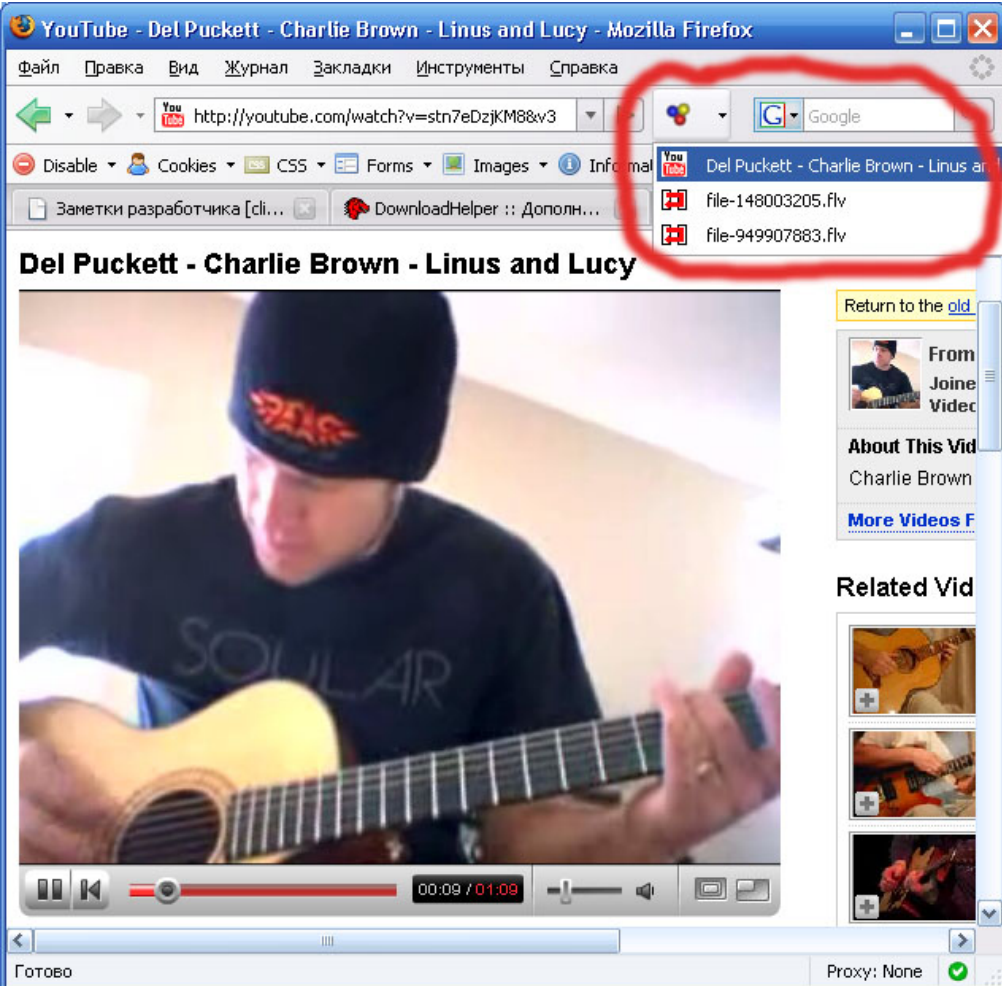
С одной стороны, это расширение тоже напрямую не связано с разработкой...

..Но с другой — бывает ли, что работая на компьютере вы получаете ссылку на интересное видео с конференции? Или заходите на сайт и видите там набор прекрасных выступлений, которые хочется скачать и посмотреть в более удобное время с планшета. Скажем, во время поездки, когда нет интернет.

Если бывает, то это расширение — для вас. Скачиваем отличные видео и смотрим, когда захотим.

Способ использования:

1. Зайти на страницу.
2. Начать смотреть видео.
3. Увидеть, как иконка расширения «оживла», кликнуть на нее и выбрать видео в нужном разрешении (если есть несколько).
4. Файл будет скачан в папку `dwhe1per` (по умолчанию), место можно поменять в настройках.



P.S.

В этот список расширений я включил самое любимое и полезное, что связано именно с расширениями браузера.

Есть дополнительные инструменты, которые становятся в систему, они идут в отдельном разделе.

У вас есть, что добавить? Расширения, которые вам очень помогли?

Укажите их в комментариях.

Отладка для Internet Explorer (в Visual Studio)

В старых версиях IE встроенный отладчик бывает неудобен, а в IE<8 его вообще нет. Зато есть очень мощное универсальное средство: Visual Studio. Которое работает и с JavaScript.

Есть два основных отладчика для IE, кроме встроенного:

Microsoft Script Editor поставляется вместе с Microsoft Office. Есть бесплатная версия — это не то. Нужна именно та, которая в поставке Microsoft Office, причем она не ставится по умолчанию. Для установки нужно найти и включить соответствующий пункт в меню.

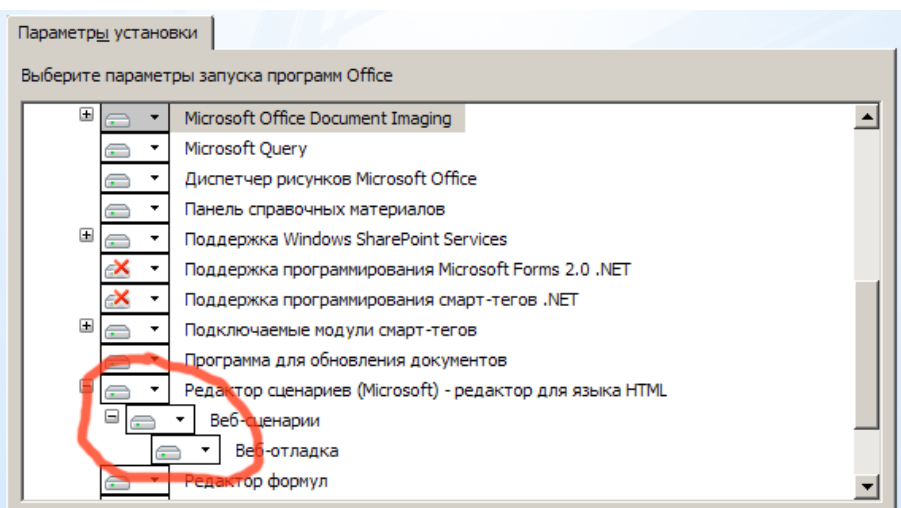
Microsoft Visual Studio — среда для разработки продуктов на самых разных языках. Она также позволяет отлаживать javascript-сценарии, как, впрочем, и многое другое.

Оба этих редактора/отладчика настраиваются похожим образом, но в общем и целом Visual Studio удобнее. С другой стороны, стоит она сильно дороже, чем Microsoft Office.

Установка

Microsoft Script Editor

При установке Microsoft Office, в подменю «средства Office», нужно отметить ветку редактор сценариев - веб-сценарии - веб-отладка, как показано на рисунке.

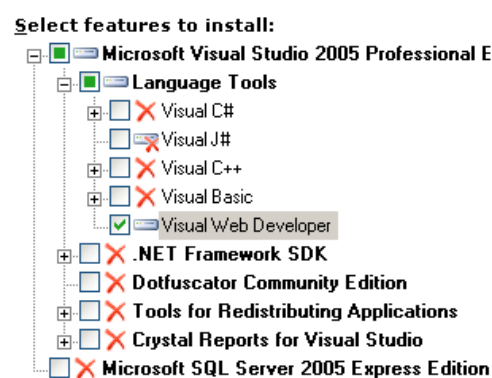


После этого, при стандартных путях, редактор/отладчик Microsoft Script Editor(MSE) будет установлен в C:\Program Files\Common Files\Microsoft Shared\OFFICE12\MSE7.EXE.

Microsoft Visual Studio

Visual Studio — большая интегрированная среда разработки, в ней большое количество компонент, большинство которых для отладки javascript вообще не нужно.

Если у Вас, скажем, профессиональная версия Visual Studio, то при установке, чтобы не засорять систему неиспользуемыми компонентами, оставим только одну галочку - напротив web-development. В дальнейшем всегда можно будет поставить что-то еще, если, например, понадобится написать программу на C#.

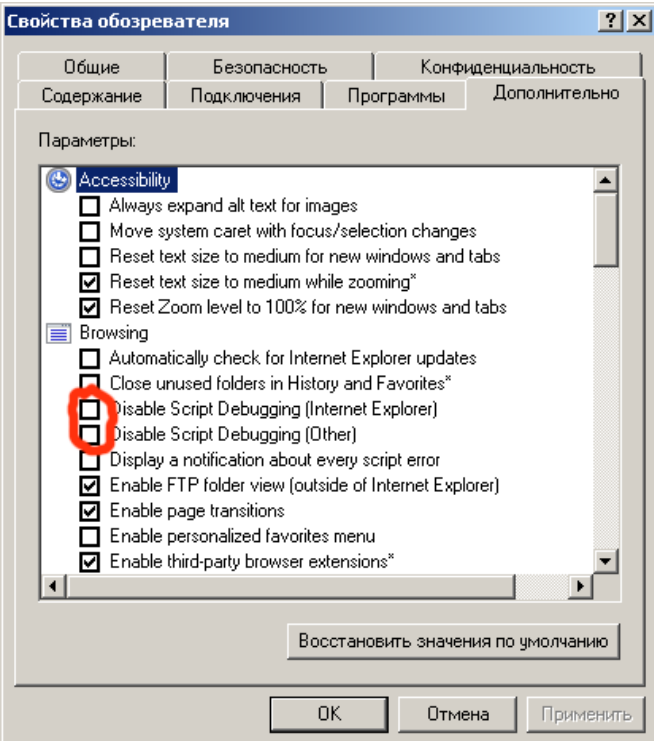


Бесплатная редакция Visual Studio называется [Visual Web Developer](#) . Она также позволяет отлаживать JavaScript, но не обладает многими возможностями профессиональной версии. Лучше уж Microsoft Office, чем она.

Галочки для отладки

Чтобы вызывать отладчик из Internet Explorer, нужно убрать две галочки из свойств, запрещающие отладку.

Галочки находятся в меню «Свойства Обозревателя» — вкладка «Дополнительно».



После установки галочек и установки Visual Studio/MSE нужно перезапустить Internet Explorer, закрыв все окна.

Способы запуска отладчика

Для Visual Studio/MSE общие варианты такие:

1. Перед загрузкой страницы отметить в IE «Вид - Внешний отладчик сценариев - прервать на следующей инструкции». Затем загрузка страницы автоматом вызовет отладчик на первой строке JavaScript-кода.

Там же можно напрямую выбрать «Отладчик сценариев - Открыть».

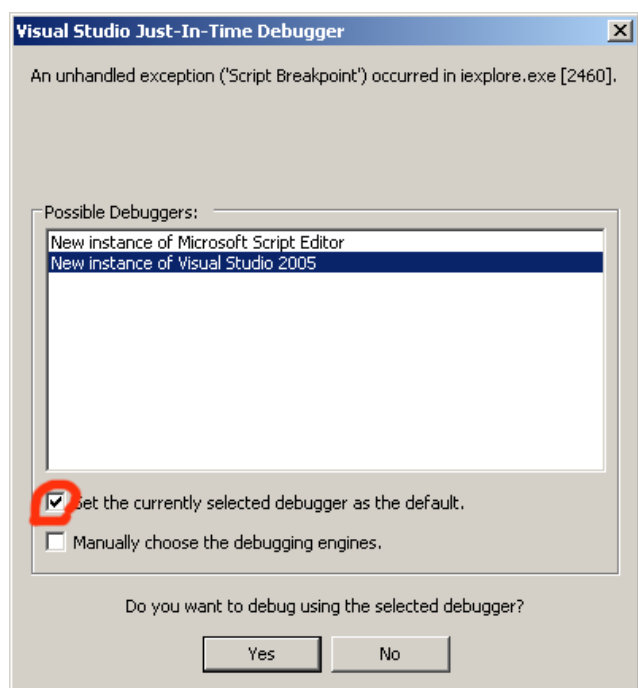
Бывает, что пункт «Отладчик сценариев» отсутствует в меню, или он там есть, но не работает. При правильной установке нужного ПО, это лечится перезапуском IE.

2. В javascript-коде поставить в нужное место оператор `debugger`.
Например:

```
1  ...  
2  var i = 5  
3  debugger;  
4  i++  
5  ...
```

Когда выполнение дойдет до строки `debugger` — вызовется отладчик точно также, как в пункте 1. Этот способ, вообще говоря, входит в стандарт языка и работает со всеми отладчиками.

Вызов отладчика этим способом осуществляется через окошко с выбором:



В этом окошке можно выбрать между MSE/Visual Studio. Если отладчик уже запущен, то также можно выбрать, запускать ли новый экземпляр программы(New instance) или использовать работающий.

Чекбокс устанавливает текущий выбор — выбором по умолчанию, так что в дальнейшем при появлении окошка достаточно будет просто нажать enter.

3. В Visual Studio Professional есть возможность подключаться к процессу: «Tools->Attach to process», и отлаживать код самого приложения или, например, запущенные скрипты. В MSE это делается через меню «Debug->Processes». При этом открывается список процессов, из них нужно выбрать соответствующий `iexplore.exe` и нажать Attach.

Отладка

После подключения браузера к отладчику, в Visual Studio можно открыть окно Script Explorer (в MSE: Running Documents), выбрать нужный JavaScript-файл, и поставить там breakpoint(ы), и они будут работать.

Если нужный breakpoint уже пройден на момент подключения — в браузере можно нажать refresh. Выполнение остановится на нужном месте.

Окна

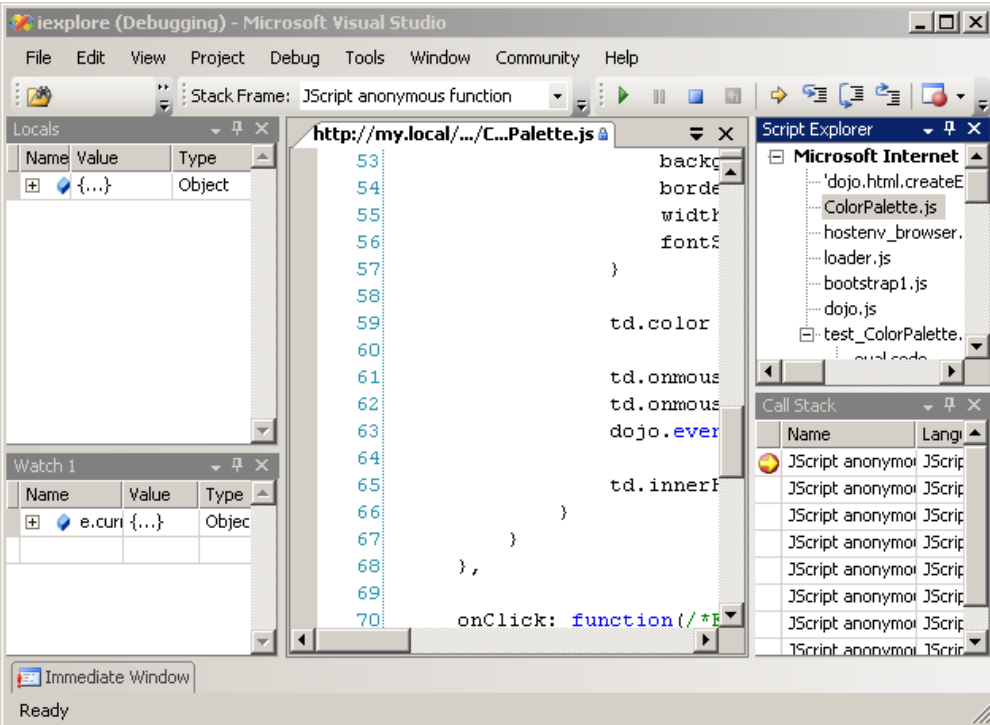
Среди вспомогательных окон Visual Studio, для отладки javascript полезны:

- Locals - список переменных в текущей области видимости
- Watch - отслеживание переменных и, вообще говоря, любых выражений
- Script Explorer - список текущих документов и скриптов
- Call Stack - стек с возможностью перемещаться по нему
- Immediate Window - javascript-консоль

Окошки можно включать-выключать из меню «Debug - Windows», или через выпадающий список на панели «Debug».

Если Script Explorer вдруг нигде нет, то его можно добавить. Для этого на свободном месте панели нужно кликнуть правой кнопкой, и выбрать Customize. В появившемся окошке слева в Categories выбрать Debug, а справа — найти кнопку Script Explorer, и перетащить ее, но не внутри окошка, а вне его — на панель Visual Studio.

Расположение и тип окон можно менять самыми разными способами, вот пример типичной конфигурации (Visual Studio открыта не на весь экран):



В зависимости от Вашего монитора, окошки Locals/Watch бывает удобно сделать плавающими(floating), или перенести вниз экрана.

Окно Immediate в этой конфигурации спряталось внизу(опция auto-hide, ставится через правый клик по заголовку окна), и вызывается по клику на заголовке.

Горячие клавиши

Работают классические hotkeys:

- F10 - выполнить строку и перейти к следующей (Step Over)
- F11 - выполнить строку, а если это функция - перейти внутрь функции (Step Into)
- Shift+F11 - выполнить остаток текущей функции и остановится сразу после return (Step Out)
- F5 - продолжить выполнение до следующего breakpoint или ошибки (Continue)
- Shift-F5 - прекратить отладку

Выделенные выражения/переменные можно добавлять в Watch правой кнопкой мыши.

Вообще, возможностей очень много, и, самое главное, их достаточно просто увидеть самому.

Успешной отладки!

Приложение. Видео.

Это видео записывалось в древнем 2007м году, но в чём-то ещё актуально.

В видео [Первый запуск и настройка](#) показан запуск первым способом и настройка Visual Studio сразу после ее установки.

В видео [Отладка загруженной страницы](#) открыт адрес в IE6, и показано как через attach подключить к нему Visual Studio и запустить отладку с breakpoint+refresh.

Скриптуемый отладочный прокси Fiddler

Fiddler — прокси, который работает с трафиком между Вашим компьютером и удаленным сервером, и позволяет инспектировать и менять его.

Fiddler можно расширять с помощью скриптов на языке JScript.NET (писать их очень просто), менять меню программы, и вообще — замечательный инструмент.

Использовать его можно с любым браузером.

Эта статья описывает Fiddler 2.4. В вашей версии Fiddler какие-то возможности могут измениться, какие-то образоваться.

Режимы подключения

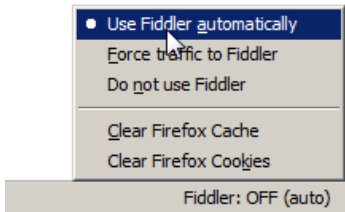
У Fiddler есть 2 способа подключения.

1. Первый — это просто запустить его. При этом он автоматически будет работать для программ, использующих WinINET.

Это, например, Internet Explorer, Chrome, приложения MS Office.

Firefox тоже автоматически подхватит Fiddler, за счёт того что при

установке Fiddler в него ставится соответствующее расширение:



- Второй — это явно поставить Fiddler как прокси для браузера, по адресу 127.0.0.1:8888. Например, так можно сделать для Opera, если уж не хочется перезапускать.

Fiddler не под Windows

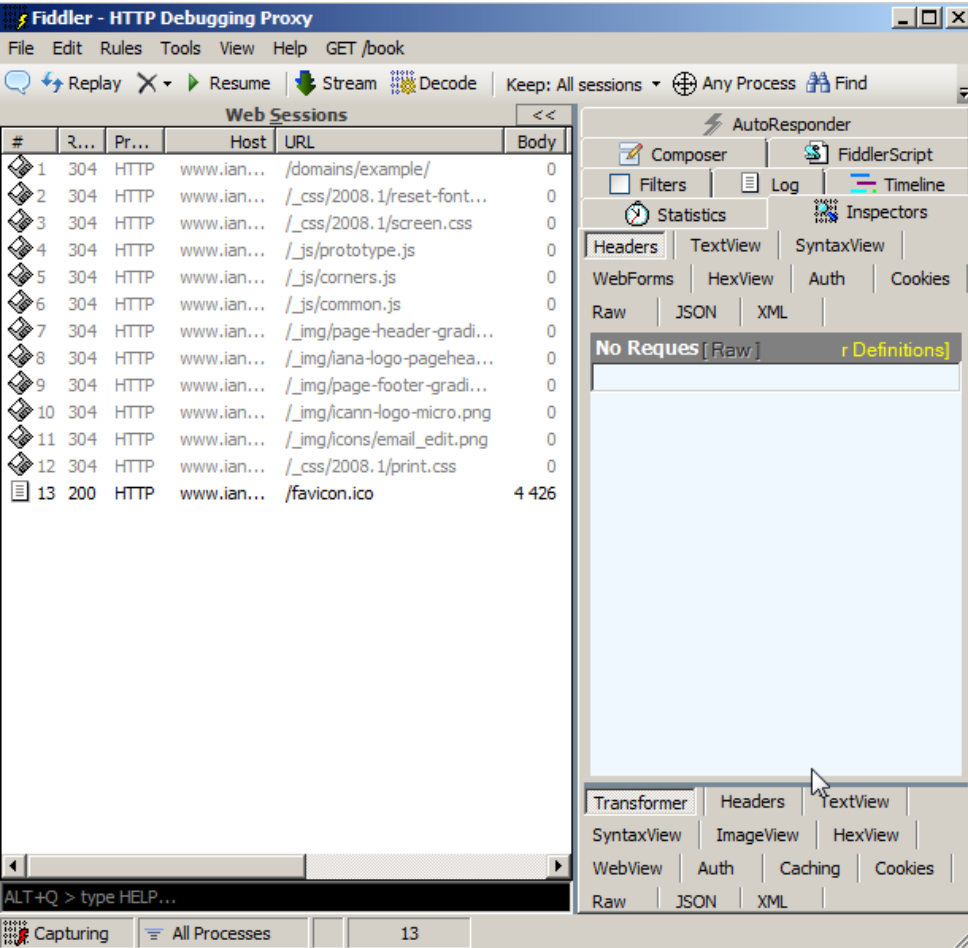
Если вы работаете не под Windows, то Fiddler можно поставить в виртуальную машину.

Чтобы сделать возможной подключение внешних браузеров, нужно включить настройках Fiddler: Tools -> Fiddler Options -> Connections(вкладка) галочку «Allow remote clients to connect». После этого Fiddler станет доступен как прокси на интерфейсе 0.0.0.0, так что можно будет браузеру из внешней ОС указать в качестве прокси виртуальную машину. И пользоваться Fiddler.

Если вы так захотите поступить, то вдобавок возьмите удобный переключатель прокси, например Elite Proxy Switcher под Firefox или [Proxy Pick](#) для IE, чтобы переключение на прокси осуществлялось в один клик.

Операции над запросами

При заходе в Fiddler, открывается окно запросов слева и рабочие вкладки справа.



Как видно, опций много, и на изображении они еле-еле помещаются. И, поверьте, возможностей — ещё больше.

Возможности

- В окне запросов слева можно просматривать и выбирать запросы, смотреть их заголовки, сохранять их на диск все вместе или по отдельности.
- **AutoResponder** — позволяет подставить свой файл вместо 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 — и исправляете, что нужно, перезагружаете страницу, проверяете — всё с комфортом.
- **Composer** — позволяет составить запрос на сервер вручную. Например, вы хотите сделать такой же AJAX-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 Execute.

- **Filters** — позволяет назначить действия в зависимости от вида запроса. Опции станут понятны после перехода на вкладку.
- **FiddlerScript** — основа мощи Fiddler: скрипт, задающий функционал. Его редактированием можно добавить или удалить пункты меню, колонки в списке запросов, и, вообще, поменять почти всё.

Язык программирования JScript.NET, который здесь используется, может взаимодействовать с Windows в полном объеме, включая коммуникацию с базой данных, Word, Excel.

Правила

Слева-сверху в меню находится пункт Rules (правила). В нём изначально находятся некоторые возможности FiddlerScript, которые поставляются «из коробки».

Хотите посмотреть, как ваш сайт будет грузиться «на GPRS»? Выберите Rules -> Performance -> Simulate Modem speeds.

Для добавления новых правил можно их задать через пункт «Customize Rules» (на JScript.NET, разумеется). В открывающемся скрипте есть пункты меню и их реализация.

При наступлении любого события из обширного списка, Fiddler вызывает соответствующий обработчик из правил. Например, onBeforeRequest, onShutdown. Стандартные правила отлично прокомментированы, и писать новые весьма просто.

FiddlerScript позволяет манипулировать заголовками, запросом, менять ширину канала, управлять выводом запроса в Fiddler и так далее и т.п.

Брейкпойнт на запросе

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 SessionInspector им можно управлять: менять сам запрос и ответ сервера (после Break on Response, когда сервер уже ответил).

Задавать прерывание на запросах определенного вида также можно через Filters.

Отладка HTTPS

Fiddler является прокси, а HTTPS шифруется от браузера до сервера-получателя, поэтому по умолчанию Fiddler не имеет доступа к содержимому HTTPS-запросов.

Чтобы его получить, Fiddler должен сыграть роль хакера-перехватчика: расшифровывать запросы, и потом отправлять дальше. Это возможно, если установить специальный сертификат: Tools -> Fiddler Options -> HTTPS(вкладка) -> выбрать все галочки.

После окончания отладки этот сертификат можно убрать.

Скачать

Fiddler можно бесплатно скачать с [сайта разработчика](#). Там же доступна [документация и видео](#).

К фиддлеру прилагается галерея расширений
<http://www.fiddlertool.com/fiddler2/extensions.asp>.

Примеры скриптов для Fiddler, которые дают общее представление о том, на что он может быть способен:

<http://www.fiddlertool.com/fiddler/dev/scriptsamples.asp>.

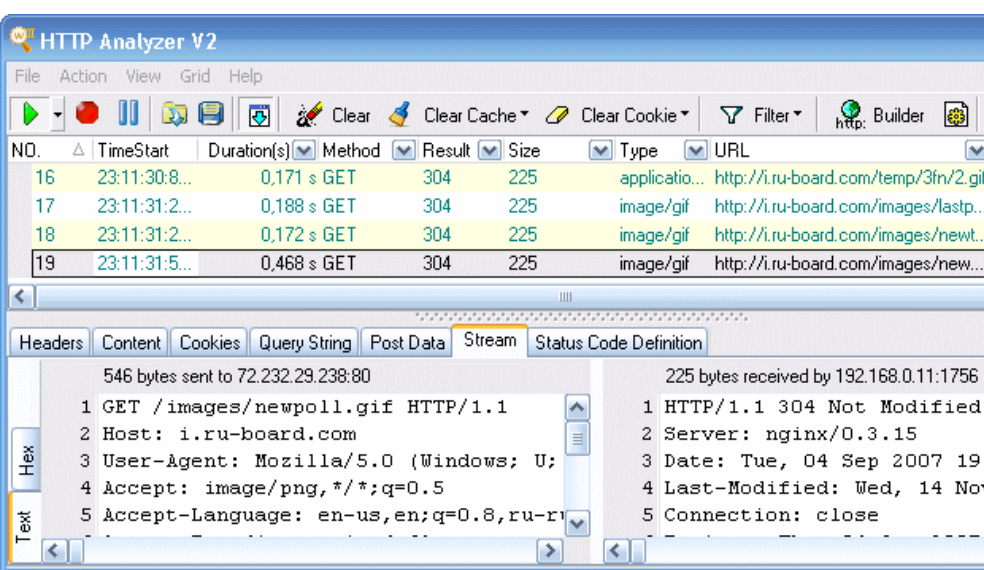
Под Windows есть как минимум два весьма основных инструмента для отладки на уровне HTTP.

Один из них — Fiddler. Другой — IE HTTP Analyzer, который представляет собой «другой угол зрения» на задачу.

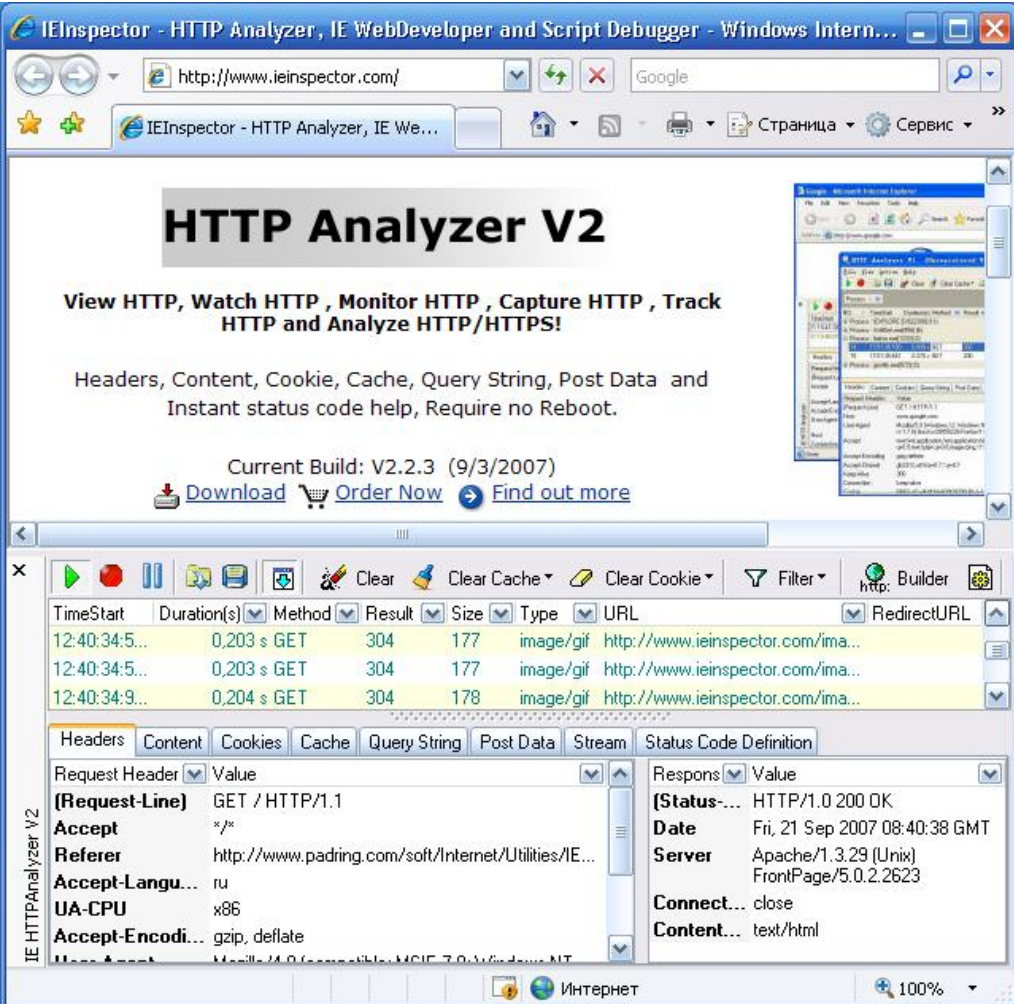
Возможности

IE HTTP Analyzer — платный, и имеет две ипостаси.

Первая - это обычная программа, запущенная в отдельном окне. При нажатии на кнопку «Play» - начинает перехват HTTP для всех браузеров, включая Firefox.



Вторая — наподобие Firebug, раскрывающаяся полоса внизу браузера Internet Explorer. Это — плагин, вызывается из панели или меню браузера.



Существуют другие программы похожего вида, но, перебрав несколько, я остановился на этой, как самой удобной.


Основные возможности сводятся к удобной отладке HTTP-запросов, да и кнопки на скриншотах выше достаточно очевидны.

Просмотр и создание HTTP-запросов.

Этим сейчас никого не удивишь, если бы не одно БОЛЬШОЕ НО.

Программа дает возможность удобно просматривать потоковые запросы.

Т.е, например, отлаживаете Вы чат. А он держит непрерывный iframe, через который сервер передает данные. Программы типа Fiddler будут честно ждать конца запроса, чтобы показать Вам его, а IE HTTP Analyzer не менее честно будет сбрасывать поток пред Ваши светлы очи.

Запросы могут поступать все новые и новые, но в панели инспектора есть кнопка , которая управляет автопереходом к следующему запросу. При анализе потока она должна быть выключена.

Эта «потоковость» IE Http Analyzer- один из важных его плюсов.

Конструирование фильтров

В отличие от правил на языке JScript.NET, которые используются для фильтрации в Fiddler, здесь фильтры на интересующие Вас запросы можно конструировать из стандартных компонент, объединяя их условиями ИЛИ/И.

Дополнительные возможности

Очистка куков и кеша браузера. Проверено, работает.

Программа платная.

Живет на <http://www.ieinspector.com/httpanalyzer/>.