# 一、Fiddler模拟网络限限速

## 直接simulate modem speeds，使用默认值限速

工具栏Rules->Performances->Simulate Modem Speeds，将该项勾选（默认根据CustomRules.js中m\_SimulateModem内容限制上传和下载速度）

## 修改CustomRules.js中m\_SimulateModem内容，设定指定上传/下载速度

工具栏Rules->CustomRules，点击打开CustomRules.js文件（快捷键ctrl+R）

修改OnBeforeReques方法中tm\_SimulateModem中上传和下载速度

public class ExampleUnitTest {  
 @Test  
 public void addition\_isCorrect() throws Exception {  
 assertEquals(4, 2 + 2);  
 }  
}  
 static function OnBeforeRequest(oSession: Session) {  
 ...  
 //模拟显示上传下载速度，保存后勾选rules->Performances->simulate modem speeds   
 if (m\_SimulateModem) {  
 // Delay sends by 300ms per KB uploaded.默认上传时每kb延时300ms   
 oSession["request-trickle-delay"] = "300";  
 // Delay receives by 150ms per KB downloaded.<span style="font-family: Arial, Helvetica, sans-serif;">默认上传时每kb延时300ms</span>   
 oSession["response-trickle-delay"] = "150";  
 }  
 <span style="white-space:pre"> </span>...

修改完成后，保存，勾选工具栏Rules->Performances->Simulate Modem Speeds   
修改完CustomRules.js文件后，Simulate Modem Speeds会被取消勾选，所以每次保存后要再次勾选才能起到限速作用。

# 二、AutoResponder

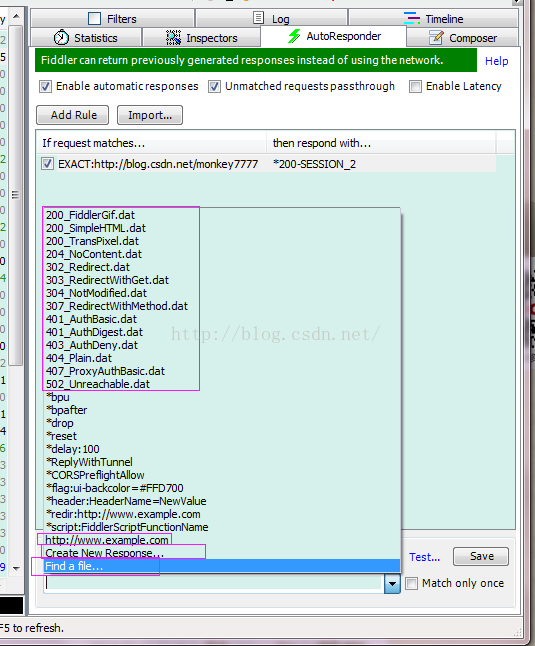
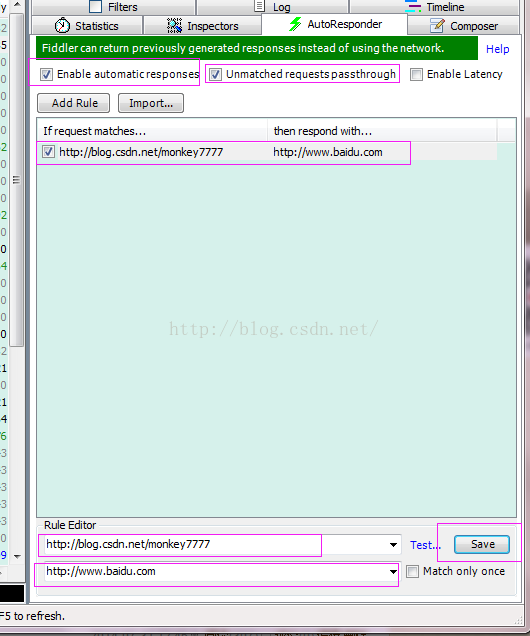
1. Fiddler地址替换
2. AutoResponder窗口设置
   1. 启动Fiddler，请求指定地址
   2. 打开Fiddler右侧AutoResponder界面
   3. 将该地址拖拽到AutoResponder里面（也可以点击选中该地址，右侧AutoResponder里面点击Add Rule）
   4. 勾选Enable automatic responses和Unmatched requests passthrough
   5. AutoResponder界面下方Rule Editor第一行为被修改地址，第二个输入框右侧箭头查看更多选项（选项如下图，为该输入框支持的输入类型），可以直接在第二个输入框输入要被替换到的地址
   6. 设置完毕，保存，重新请求该url,查看请求地址是否已被替换，已被替换地址高亮显示。

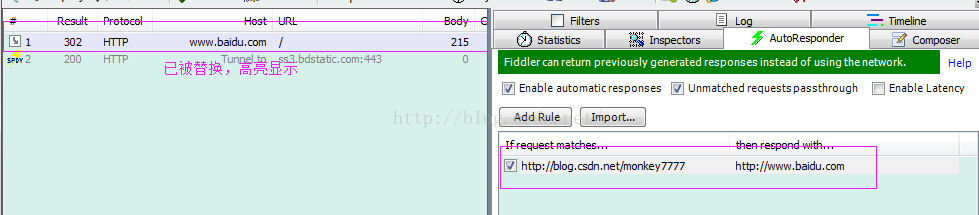
取消勾选Rules列表里该地址，即可取消该地址替换。勾选后再次请求该地址仍会执行替换。

1. 命令行替换
   1. 在Fiddler命令行出入：url replace <http://blog.csdn.ner/monkey7777> www.baidu.com
   2. 按enter键，即可将之前发送的到monkey777的请求转发成到baidu的请求了。

这种替换只是暂时的。

1. 截图如下：



1. Fiddler设置请求404

步骤与上面地址替换类似，只需在RuleEditor处第二行选择404类型“404\_Plain.dat”，保存重新请求即可。设置200-502类型错误与此类似。

1. Fiddler修改返回数据
2. Fiddler请求指定接口，查看接口返回数据
3. 修改接口返回数据，保存到本地。(在Fiddler右侧Inspector界面下方返回值的raw中查看返回数据->点击右下角View in notepad在文本中打开->文本中删去返回数据的报头->把数据改为想要的数据->另存为需要的格式（xx.json、xx.txt等)保存在本地。
4. AutoResponder中为该地址添加rule。
5. Rule Editor中编辑跳转地址，选择Choose a File…->选择本地保存的数据文件->save
6. 重新请求该地址，查看返回数据。

# 三、常用快捷键

* ctrl+R 打开CustomRules.js文件
* ctrl+X 清空url列表
* ctrl+1/2/3/4/5/6 标记url为指定颜色（需点击选中该url）
* ctrl+0 取消标记某url
* shift+r 重新请求某地址

# 四、左侧地址列表区域显示ip地址

1. 打开CustomRules.js文件
2. 找到main函数，添加FiddlerObject.UI.lvSessions.AddBoundColumn("ServerIP",120,"X-HostIP");

static function Main() {  
 var today: Date = new Date();  
 FiddlerObject.StatusText = " CustomRules.js was loaded at: " + today;  
 FiddlerObject.UI.lvSessions.AddBoundColumn("ServerIP",120,"X-HostIP");  
  
 // Uncomment to add a "Server" column containing the response "Server" header, if present  
 // FiddlerObject.UI.lvSessions.AddBoundColumn("Server", 50, "@response.server");  
  
 // Uncomment to add a global hotkey (Win+G) that invokes the ExecAction method below...  
 // FiddlerApplication.UI.RegisterCustomHotkey(HotkeyModifiers.Windows, Keys.G, "screenshot");  
}

# 五、Fiddler模拟web请求的四种方法

1. **现有请求拖拽**
2. Fiddler右侧功能面板切换到Composer界面的Parsed面板
3. 左侧区域抓包显示各种请求
4. 拖拽一条请求（可以是post/get）到右侧composer面板
5. 这时可以看到该面板自动填充相应的数据
6. 查看已填充数据，可在相应位置修改各数据

* 可修改切换发送方式（POST/GET/…）
* 可修改请求地址信息
* 可修改web请求遵循的协议
* 可修改请求头信息（Request Headers内容）
* 可修改Request Body信息（可在此处修改post时所传的参数）

1. 点击execute按钮，发送请求
2. 此时，可在左侧抓包面板中看到模拟的请求了
3. **手动创建好请求**

* Fiddler右侧功能面板切换到Composer界面的Parsed面板
* 选择发送方式，可以是post/get/…
* 填写请求地址
* 选择web请求协议
* Request Headers处填写请求头信息，非必要时可以不填
* Post方式发送时，在Request Body处填写所传参数，没有可不填
* 点击execute发送请求

1. **Row面板模拟请求**

* 拖拽某条请求进入Row面板，该面板以文本形式展示拖拽请求进入parsed面板时的请求信息
* 展示的请求信息包括：请求方式、请求地址、请求头信息、post参数（post请求时显示）
* 可修改相应的值模拟请求（同parsed面板）
* 点击execute发送该请求

1. **Scratchpad面板模拟发送多条请求（非并发）**

* 拖拽多条请求到Scratchpad面板（可每次拖一条，拖拽多次；也可选中多条请求，一次拖拽多条）
* 该面板展示各个请求的请求信息
* 可修改某些请求的某些内容
* 点击点击execute\*\*同时发送多条请求\*\*

1. **Options面板**

* Inspect Session
  + 勾选，点击execute后，直接跳转到inspector页面
  + 不勾选，点击execute后，仍保留在当前页面
* Fix Content-Length header
  + adjusts the value of the Content-Length request header (if present) to match the size of the request body.
* Follow Redirects
  + causes a HTTP/3xx redirect to trigger a new request, if possible. The Composer will follow up to fiddler.composer.followredirects.max default redirections.
* Automatically Authenticate
  + causes Fiddler to automatically respond to HTTP/401 and HTTP/407 challenges that use NTLM or Negotiate protocols using the current user’s Windows credentials.
* Tear off
  + 点击后，composer面板单独窗口显示

# 六、Fiddler查看接口返回值

1. **查看接口请求信息**

1）启动Fiddler，开启移动设备抓包

2）点击Fiddler某http请求地址

3）查看右侧inspector面板，上方显示请求信息，下方显示返回信息

* inspector - Request部分 - Headers：请求方式、http协议、Client/Transport等信息
* inspector - Request部分 - WebForms：可查看get或者post方式请求的接口的参数
* inspector - Response部分 - Headers：返回状态码、http协议、Transport等信息
* inspector - Response部分 - ImageView：查看返回图片
* inspector - Response部分 - Raw：查看返回文本信息
* inspector - Response部分 - Json：查看返回Json文件信息
* inspector - Response部分 - xml：查看返回xml信息

注：若返回信息出现乱码情况，可点击“response is encodeed and may need decodeing before ”...解决。