Fiddler Training

Table of Contents

1. HTTP Requests and HTTP responses

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol that can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through the extension of its request methods, error codes, and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1" and is an update to RFC 2068.

It has a Request/Response paradigm (e.g., Browser makes a request to the server, and there is a single response for the request). Each request and each response are made up of a *header* and a *body* area.

HTTP Request

```
GET https://www.pbnet.ro/ HTTP/1.1 ← asking for the URL over HTTP 1.1
Host: www.pbnet.ro
Connection: keep-alive
sec-ch-ua: " Not;A Brand";v="99", "Google Chrome";v="91", "Chromium";v="91"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/91.0.4472.106 Safari/537.36 ← user agent for the browser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9 ← site language
```

HTTP Response

```
HTTP/1.1 200 OK ← HTTP 200 means the request is OK.
Date: Thu, 17 Jun 2021 08:54:52 GMT ← Date when the response occurred
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
```

```
X-Powered-By: PHP/7.4.20 ← Software running on the server
Link: <https://www.pbnet.ro/index.php?rest_route=/>; rel="https://api.w.org/",
<https://www.pbnet.ro/index.php?rest_route=/wp/v2/pages/1045>; rel="alternate";
type="application/json", <https://wp.me/PanZp-gR>; rel=shortlink
CF-Cache-Status: DYNAMIC
cf-request-id: 0abac73dc9000062834018c000000001
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-
cgi/beacon/expect-ct"
Report-To:
{"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v2?s=%2Bys2%2BH76noJIK5
Cr75s3%2F8I7AZj8vpYB42DO9fPdJu%2F7qKoomnX7EwjMhOdpEMu3bN5hkksQ95epmHE4NucJY5tjzqtmKQG
%2FWrKQvRvMZ%2BOSvwrMaomjc5YU"}],"group":"cf-nel","max_age":604800}
NEL: {"report_to":"cf-nel","max_age":604800}
Server: cloudflare
CF-RAY: 660b0e42db0f6283-OTP
alt-svc: h3-27=":443"; ma=86400, h3-28=":443"; ma=86400, h3-29=":443"; ma=86400,
h3=":443"; ma=86400
Content-Length: 39975

<!DOCTYPE html>
<html class="no-js" lang="en-US">
<head>
<meta charset="UTF-8">
```

SharePoint Site request:

```
GET https://pbnet.sharepoint.com/sites/test1 HTTP/1.1 ← GET request to the site.
Host: pbnet.sharepoint.com
Connection: keep-alive
sec-ch-ua: " Not;A Brand";v="99", "Microsoft Edge";v="91", "Chromium";v="91"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/91.0.4472.101 Safari/537.36 Edg/91.0.864.48 ← Browser user agent
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.9
Service-Worker-Navigation-Preload: true
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9 ← site language
Cookie:
rtFa=cvMh2eZhzHmDcdEnBt2R9+sb4oxEO+HvTTFBgOw95y4mRTk5NjY3RkMtODRENiOOMjU4LUIyNDYtMjk3
NTFDRjA3MDdGDMocprPgUL5n4wGwsxt8gPpiZNfsMpXa1ihrx1gmmx7hXyMwjcaVAzEKCukMIYfMONkGRs2ou
O/ItHJBRgidGQvPqSewwHZB3bDDHL7Ae1x7caxaIh55N7UqrwQvyDPssZt2OCYMFNvS9mvO0NOOiII2uJn4rA
8pCP7qcHJu6VBNhaJDOSpLi99mNV/9hab+X/EzOCUeZitPDbYwEcFN5q6Or0UQer0ekRxhyhnGvoyUh4b6pk4
sDxi5vqKfu27wXRp3bizXui/vpCsUxqT2PQz/J5hq9bsEGE5YgoDr1Zty8tAIZO77CX7hAKD+AhORGHemgImj
NAkkSTsfFbgJXkUAAAA=;
FedAuth=77u/PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz48U1A+VjEwLDBoLmZ8bWVtY
mVyc2hpcHwxMDAzMDAwMDhmMGJiYjk0QGxpdmUuY29tLDAjLmZ8bWVtYmVyc2hpcHxhbmRyZWlAcmFjaGl0YS
5uZXQsMTMyNjY5MzUyODMwMDAwMDAwLDEzMjQ3ODMwMDExMDAwMDAwMCwxMzI2ODgyNjMwMjI3MjM0ODUsODY
uMTIzLjE4Ni4yOSw2NyxlOTk2NjdmYy04NGQ2LTQyNTgtYjI0Ni0yOTc1MWNmMDcwN2YsLDQwYzY5ZmNiLWU4
OTItNDAzYy05NjM2LWVmYmU2YmZhM2FjZSw0NGI2ZDI5Zi0xMGIwLWMwMDAtM2YwNC0yN2Y2NTk1MDI0ZjAsN
DRiNmQyOWYtMTBiMC1jMDAwLTNmMDQtMjdmNjU5NTAyNGYwLCwwLDEzMjY4NDgwNzAyMjI1NDU0NCwxMzI2OD
Y1MzUwMjIyNTQ1NDQsLCxleUo0YlhOZlkyTWlPaUpiWENKRFVERmNJbDBpTENKNGJYTmZjM050SWpvaU1TSjk
sMjY1MDQ2NzcOMzk5OTk5OTk5OSwxMzI2ODM5NDMwMTAwMDAwMDAsZjdmMjVmMzktZWVjNi00MDM5LTgzZjYt
ZTAzNzBhY2VkYzc5LCwsLCwsTlVTZOo1bOtPeCs0ZWRhclJPYO1xVOhFTytPNjBjYlNZQW5wNHNxUDFJTFRNZ
1ZVTmpaYStMQOFGeEVUd2t5ZytMbXpSeEhYSjc2cFphN2YzejlmMTlCOFpRK2N3all5ZEhDbU9uNDJVMS9Rcn
RnTEg2ZXpPYXJaNGVyQOl4Z1NXdnBnYjk5WDBKcVdMZUNHSHF4aTFmcXR2TVE4RlVOT0dQYXRlVHNOR2J0K0V
BRDFEWllXc3JpeUtCV3ZiY3IyM2w1NWp5UmQwK3lpWjZsdmUrYO1NTmJhK2hzUDBwS3NjZzJaOG12MOZJeEVJ
S0N0aHdINlYxVlVKUElCVFNnM2JmK1p4QzJ5N3ZSSzJOYONUTzdOVXplSHJBMOticTZXdTV1blZSMGtCMDF3d
ONFROllMlExSEUvQjNoY2xmaXg0blNsS3o4NHpWVUJPSWxvVjZpUjZ3PTO8L1NQPg==
```

SharePoint site response:

```
HTTP/1.1 200 OK ← HTTP 200 means everything was OK.
Cache-Control: private
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT
COM INT NAV ONL PHY PRE PUR UNI"
Set-Cookie:
rtFa=cvMh2eZhzHmDcdEnBt2R9+sb4oxEO+HvTTFBgOw95y4mRTk5NjY3RkMtODRENiOOMjU4LUIyNDYtMjk3
NTFDRjA3MDdGDMocprPgUL5n4wGwsxt8gPpiZNfsMpXa1ihrx1gmmx7hXyMwjcaVAzEKCukMIYfMONkGRs2ou
O/ItHJBRgidGQvPqSewwHZB3bDDHL7Ae1x7caxaIh55N7UqrwQvyDPssZt2OCYMFNvS9mvOONOOiII2uJn4rA
8pCP7qcHJu6VBNhaJDOSpLi99mNV/9hab+X/EzOCUeZitPDbYwEcFN5q6OrOUQer0ekRxhyhnGvoyUh4b6pk4
sDxi5vqKfu27wXRp3bizXui/vpCsUxqT2PQz/J5hq9bsEGE5YgoDr1Zty8tAIZO77CX7hAKD+AhORGHemgImj
NAkkSTsfFbgJXkUAAAA=; domain=sharepoint.com; expires=Tue, 22-Jun-2021 09:05:02 GMT;
path=/; SameSite=None; secure; HttpOnly
Set-Cookie:
FedAuth=77u/PD94bWwgdmVyc2lvbjOiMS4wIiBlbmNvZGluZzOidXRmLTgiPz48U1A+VjEwLDBoLmZ8bWVtY
mVyc2hpcHwxMDAzMDAwMDhmMGJiYjk0QGxpdmUuY29tLDAjLmZ8bWVtYmVyc2hpcHxhbmRyZWlAcmFjaGl0YS
5uZXQsMTMyNjY5MzUyODMwMDAwMDAwLDEzMjQ3ODMwMDExMDAwMDAwMCwxMzI2ODgyNjMwMjI3MjM0ODUsODY
uMTIzLjE4Ni4yOSw2NyxlOTk2NjdmYy04NGQ2LTQyNTgtYjI0Ni0yOTc1MWNmMDcwN2YsLDQwYzY5ZmNiLWU4
OTItNDAzYy05NjM2LWVmYmU2YmZhM2FjZSw0NGI2ZDI5ZiOxMGIwLWMwMDAtM2YwNC0yN2Y2NTk1MDI0ZjAsN
DRiNmQyOWYtMTBiMC1jMDAwLTNmMDQtMjdmNjU5NTAyNGYwLCwwLDEzMjY4NDgwNzAyMjI1NDU0NCwxMzI2OD
Y1MzUwMjIyNTQ1NDQsLCxleUo0YlhOZlkyTWlPaUpiWENKRFVERmNJbDBpTENKNGJYTmZjM05OSWpvaU1TSjk
sMjY1MDQ2NzC0Mzk5OTk5OTk5OSwxMzI2ODM5NDMwMTAwMDAwMDAsZjdmMjVmMzktZWVjNi00MDM5LTgzZjYt
ZTAzNzBhY2VkYzc5LCwsLCwsTlVTZ0o1b0tPeCs0ZWRhclJPY01xV0hFTytPNjBjYlNZQW5wNHNxUDFJTFRNZ
1ZVTmpaYStMQ0FGeEVUd2t5ZytMbXpSeEhYSjc2cFphN2YzejlmMTlCOFpRK2N3all5ZEhDbU9uNDJVMS9Rcn
RnTEg2ZXpPYXJaNGVyQOl4Z1NXdnBnYjk5WDBKcVdMZUNHSHF4aTFmcXR2TVE4RlVOT0dQYXRlVHNOR2J0K0V
BRDFEWllXc3JpeUtCV3ZiY3IyM2w1NWp5UmQwK3lpWjZsdmUrY01NTmJhK2hzUDBwS3NjZzJaOG12M0ZJeEVJ
S0N0aHdINlYxVlVKUElCVFNnM2JmK1p4QzJ5N3ZSSzJOY0NUTzdOVXplSHJBM0ticTZXdTV1blZSMGtCMDF3d
0NFROllMlExSEUvQjNoY2xmaXg0blNsS3o4NHpwVUJPSWxvVjZpUjZ3PT08L1NQPg==; expires=Tue, 22-
Jun-2021 09:05:02 GMT; path=/; SameSite=None; secure; HttpOnly
Set-Cookie:
rtFa=cvMh2eZhzHmDcdEnBt2R9+sb4oxEO+HvTTFBgOw95y4mRTk5NjY3RkMtODRENiOOMjU4LUIyNDYtMjk3
NTFDRjA3MDdGDMocprPgUL5n4wGwsxt8gPpiZNfsMpXa1ihrx1gmmx7hXyMwjcaVAzEKCukMIYfMONkGRs2ou
O/ItHJBRgidGQvPqSewwHZB3bDDHL7Ae1x7caxaIh55N7UqrwQvyDPssZt2OCYMFNvS9mvOONOiII2uJn4rA
8pCP7qcHJu6VBNhaJDOSpLi99mNV/9hab+X/EzOCUeZitPDbYwEcFN5q6OrOUQer0ekRxhyhnGvoyUh4b6pk4
sDxi5vqKfu27wXRp3bizXui/vpCsUxqT2PQz/J5hq9bsEGE5YgoDr1Zty8tAIZO77CX7hAKD+AhORGHemgImj
NAkkSTsfFbgJXkUAAAA=; domain=sharepoint.com; expires=Tue, 22-Jun-2021 09:05:02 GMT;
path=/; SameSite=None; secure; HttpOnly
Set-Cookie:
FedAuth=77u/PD94bWwgdmVyc2lvbjOiMS4wIiBlbmNvZGluZzOidXRmLTgiPz48U1A+VjEwLDBoLmZ8bWVtY
mVyc2hpcHwxMDAzMDAwMDhmMGJiYjk0QGxpdmUuY29tLDAjLmZ8bWVtYmVyc2hpcHxhbmRyZWlAcmFjaGl0YS
5uZXQsMTMyNjY5MzUyODMwMDAwMDAwLDEzMjQ3ODMwMDExMDAwMDAwMCwxMzI2ODgyNjMwMjI3MjM0ODUsODY
uMTIzLjE4Ni4yOSw2NyxlOTk2NjdmYy04NGQ2LTQyNTgtYjI0Ni0yOTc1MWNmMDcwN2YsLDQwYzY5ZmNiLWU4
OTItNDAzYy05NjM2LWVmYmU2YmZhM2FjZSw0NGI2ZDI5ZiOxMGIwLWMwMDAtM2YwNC0yN2Y2NTk1MDI0ZjAsN
DRiNmQyOWYtMTBiMC1jMDAwLTNmMDQtMjdmNjU5NTAyNGYwLCwwLDEzMjY4NDgwNzAyMjI1NDU0NCwxMzI2OD
Y1MzUwMjIyNTQ1NDQsLCxleUo0YlhOZlkyTWlPaUpiWENKRFVERmNJbDBpTENKNGJYTmZjM05OSWpvaU1TSjk
sMjY1MDQ2NzC0Mzk5OTk5OTk5OSwxMzI2ODM5NDMwMTAwMDAwMDAsZjdmMjVmMzktZWVjNi00MDM5LTgzZjYt
ZTAzNzBhY2VkYzc5LCwsLCwsTlVTZ0o1b0tPeCs0ZWRhclJPY01xV0hFTytPNjBjYlNZQW5wNHNxUDFJTFRNZ
1ZVTmpaYStMQ0FGeEVUd2t5ZytMbXpSeEhYSjc2cFphN2YzejlmMTlCOFpRK2N3all5ZEhDbU9uNDJVMS9Rcn
RnTEg2ZXpPYXJaNGVyQOl4Z1NXdnBnYjk5WDBKcVdMZUNHSHF4aTFmcXR2TVE4RlVOT0dQYXRlVHNOR2J0K0V
BRDFEWllXc3JpeUtCV3ZiY3IyM2w1NWp5UmQwK3lpWjZsdmUrY01NTmJhK2hzUDBwS3NjZzJaOG12M0ZJeEVJ
S0N0aHdINlYxVlVKUElCVFNnM2JmK1p4QzJ5N3ZSSzJOY0NUTzdOVXplSHJBM0ticTZXdTV1blZSMGtCMDF3d
0NFROllMlExSEUvQjNoY2xmaXg0blNsS3o4NHpwVUJPSWxvVjZpUjZ3PT08L1NQPg==; expires=Tue, 22-
Jun-2021 09:05:02 GMT; path=/; SameSite=None; secure; HttpOnly
Set-Cookie:
SPWorkLoadAttribution=Url=https://pbnet.sharepoint.com/sites/test1&AppTitle=WebPartPa
geHomePage; expires=Thu, 17-Jun-2021 09:10:52 GMT; path=/
Set-Cookie:
e99667fc84d64258b24629751cf0707f5e56f38de6c64ef9868445846d71ea4bi%3AO%23%2Ef%7Cmember
ship%7Candrei%40rachita%2Enet=0; expires=Fri, 18-Jun-2021 09:10:22 GMT; path=/;
HttpOnly
Set-Cookie: SearchSession=f5e8ffb3%2D795e%2D46ac%2Daabc%2D9d0a66c608f3;
path=/sites/test1; secure
Set-Cookie:
https%3A%2F%2Fpbnet%2Esharepoint%2Ecom%2Fsites%2Ftest1%2FDiscovery=WorkspaceSiteName=
VGVzdCBTUE8=&WorkspaceSiteUrl=aHROcHM6Ly9wYm5ldC5zaGFyZXBvaW50LmNvbS9zaXRlcy9OZXNOMQ=
=&WorkspaceSiteTime=MjAyMS0wNiOxN1QwOToxMDoyNQ==; expires=Sat, 17-Jul-2021 09:10:25
GMT; path=/_vti_bin/Discovery.asmx; secure
X-SharePointHealthScore: 2 ← SharePoint Health Score. Will be discussed in this
training.
Referrer-Policy: no-referrer, strict-origin-when-cross-origin
X-AspNet-Version: 4.0.30319 ← ASP.NET Version
SPRequestGuid: 92b6d29f-f0c7-c000-1fc5-c95a5d6ad419 ← Correlation ID
request-id: 92b6d29f-f0c7-c000-1fc5-c95a5d6ad419
MS-CV: n9K2ksfwAMAfxclaXWrUGQ.0
Report-To: {"group":"network-
errors","max_age":86400,"endpoints":[{"url":"https://spo.nel.measure.office.net/api/r
```

```
eport?tenantId=e99667fc-84d6-4258-b246-29751cf0707f&destinationEndpoint=Edge-Prod-
BUH01r4&frontEnd=AFD"}]}
NEL: {"report_to":"network-
errors","max_age":86400,"success_fraction":0.001,"failure_fraction":1.0}
Strict-Transport-Security: max-age=31536000
X-FRAME-OPTIONS: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self' teams.microsoft.com
*.teams.microsoft.com *.skype.com *.teams.microsoft.us local.teams.office.com
*.powerapps.com *.yammer.com *.officeapps.live.com *.office.com *.stream.azure-
test.net *.microsoftstream.com;
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 16.0.0.21402 ← SharePoint Farm Build version
X-Content-Type-Options: nosniff
X-MS-InvokeApp: 1; RequireReadOnly
X-Cache: CONFIG_NOCACHE
X-MSEdge-Ref: Ref A: B99C10FFB998406B9D60F8094F179633 Ref B: BUH01EDGE0320 Ref C:
2021-06-17T09:10:21Z
Date: Thu, 17 Jun 2021 09:10:24 GMT ← date when the response was received
Content-Length: 444614


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html dir="ltr" lang="en-US">
<head><meta name="GENERATOR" content="Microsoft SharePoint" /><meta http-
equiv="Content-type" content="text/html; charset=utf-8" /><meta http-equiv="X-UA-
Compatible" content="IE=edge"/><script type='text/javascript'>var _browserisFlight =
true;</script><meta http-equiv="Expires" content="0" /><meta name="msapplication-
TileImage" content="/_layouts/15/images/SharePointMetroAppTile.png" /><meta
name="msapplication-TileColor" content="#0072C6" /><title>

       Test SPO - Home
```

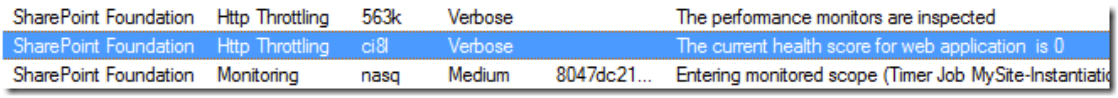
X-SharePointHealthScore explained

The SharePoint Health Score was first introduced in SharePoint 2010. The Health Score determines the health of a SharePoint server/web application on a scale from 0 to 10, where 0 is the healthiest state. SharePoint automatically starts throttling requests once the Health Score is too high. The Health Score can be calculated using many parameters, such as memory usage, concurrent requests, etc.

How the SharePoint Health Score is calculated

So, how does SharePoint calculate this Health Score?
When a SharePoint Web Application spins up, internally a new thread is created (you can see it in debuggers with the name SPPerformanceInspector).
This thread regularly reads performance counters and calculates the Health Score.
Each time it calculates the Health Score it will log this to the Trace Logs (Category=Http Throttling, Level=Verbose):

"The current health score for web application is X."



SharePoint Foundation	Http Throttling	563k	Verbose		The performance monitors are inspected
SharePoint Foundation	Http Throttling	ci8l	Verbose		The current health score for web application is 0
SharePoint Foundation	Monitoring	nasq	Medium	8047dc21...	Entering monitored scope (Timer Job MySite-Instantiati

Performance Counters

The Health Score is calculated from a set of Performance Counters. By default, SharePoint 2013 (and SharePoint 2010) uses two performance counters for this:

- Memory/Available MBytes
- ASP.NET/Requests Current

Refresh Interval

The Health Scores are by default updated every 5 seconds.

2. What is Fiddler?

In two words: Fiddler is an HTTP Tracing tool.

Fiddler is an HTTP debugging proxy server application written by Eric Lawrence, formerly a Program Manager on the Internet Explorer development team at Microsoft. Fiddler is now the property of Telerik.

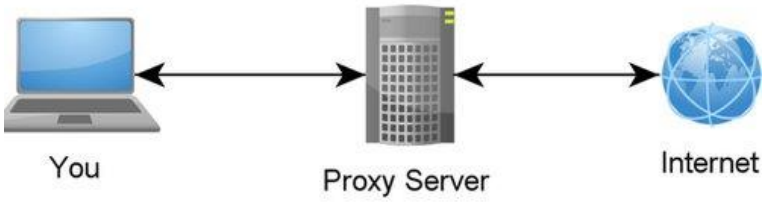
Fiddler is written in C#.

It is used to:

- Troubleshoot problems.
- Performance evaluation
- "Play (fiddle)" with requests and responses.
- Security testing (through request modification)
- Visualize page requests (using the timeline feature)
- Site reviews

3. How does it work?

- Basically, it is a proxy (a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or another resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems. Today, most proxies are web proxies, facilitating access to content on the World Wide Web, providing anonymity, and may be used to bypass IP address blocking.

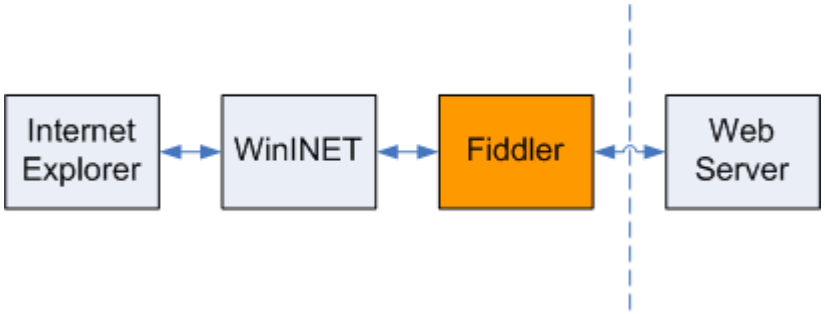


Fiddler can be used as a:

  - Local Proxy
  - Remote Proxy (for smartphones, tablets, and non-Windows platforms)
- It adjusts your browser's proxy configuration (through WinInet) to intercept traffic. In some scenarios, you can use the "embedded FQDN" that Fiddler offers and use: ipv4.fiddler or ipv6.fiddler.

After you start Fiddler, the program registers itself as the system proxy for Microsoft Windows Internet Services (WinInet), the HTTP layer used by Internet Explorer, Microsoft Office, and many other products.

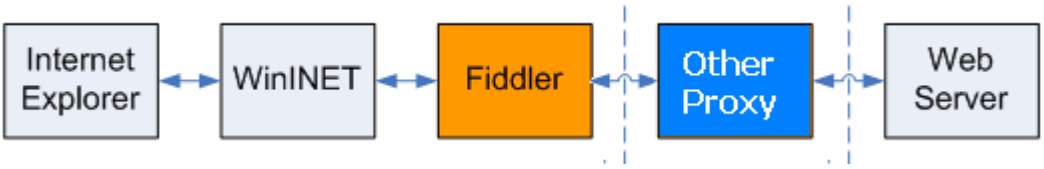
As the system proxy, all HTTP requests from WinInet flow through Fiddler before reaching the target Web servers. Similarly, all HTTP responses flow through Fiddler before being returned to the client application.



When you close Fiddler, it unregisters itself as the system proxy before shutting down.

All current versions of Fiddler support chaining to upstream proxies (either autodetected or manually specified).

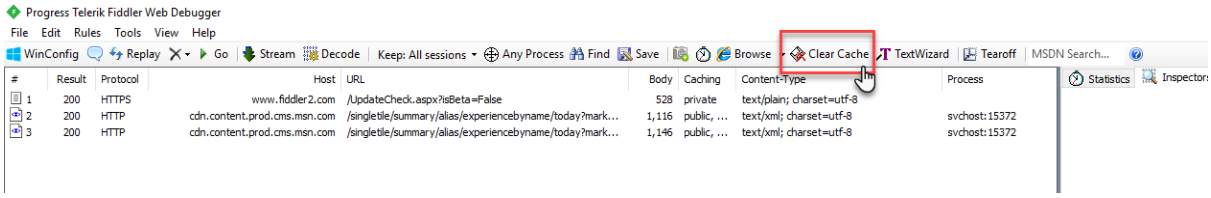
The result is an architecture like this:



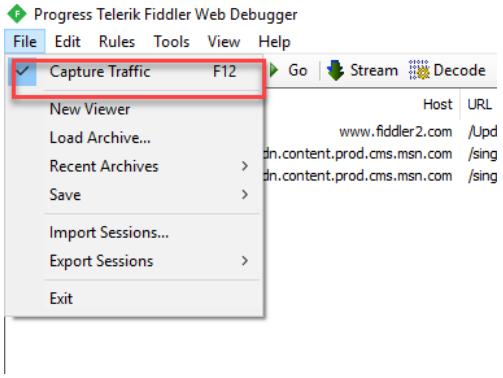
Note that Fiddler does not support upstream proxy configuration scripts that are accessed using the FILE:// protocol, only those accessed using the HTTP or HTTPS protocols (so far, no one seems to have hit this limitation in the last 6 years).

Basic Fiddler Usage

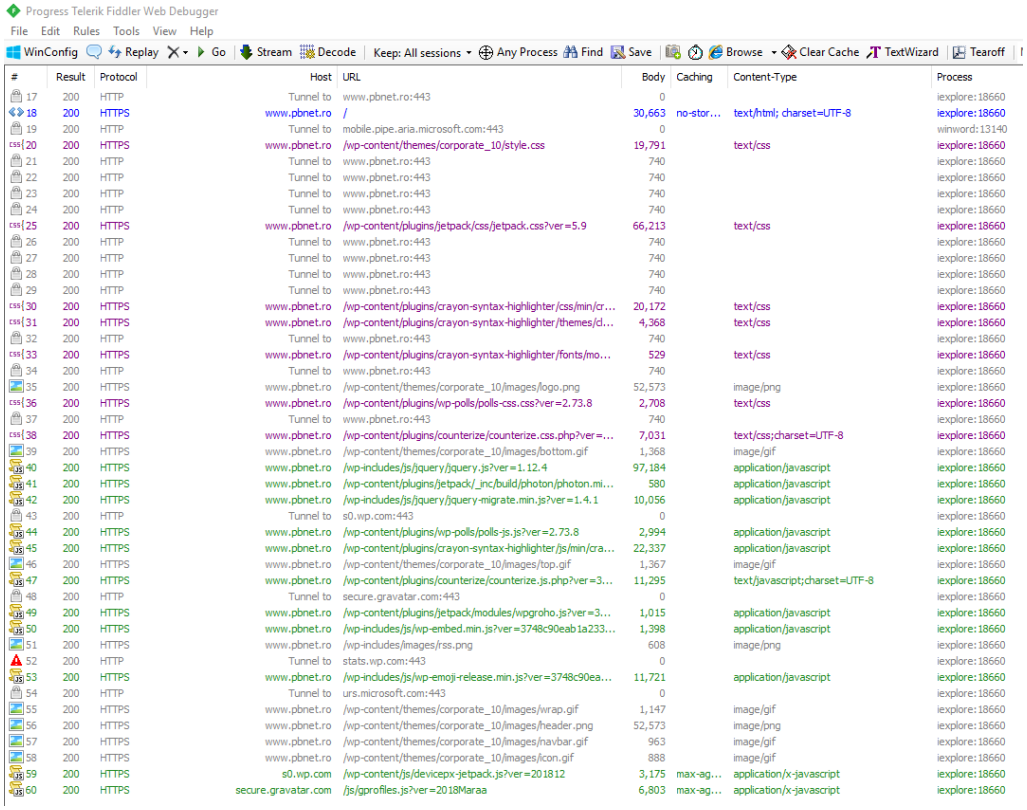
It is good practice to clear the cache before starting any trace:



Be sure that the capture traffic is checked:



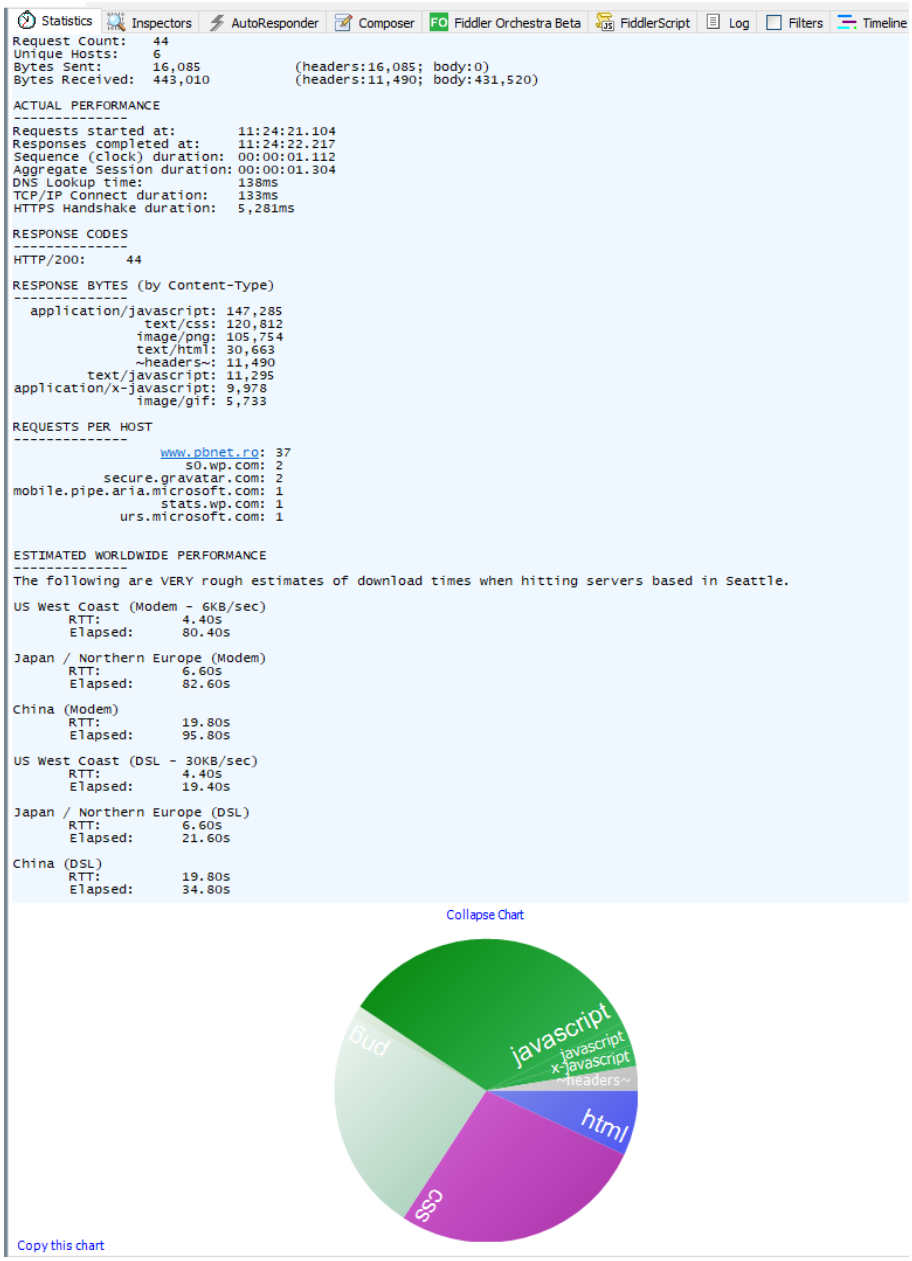
In the main Fiddler window, you can easily see the frames (from 17 to 60 in our demo):



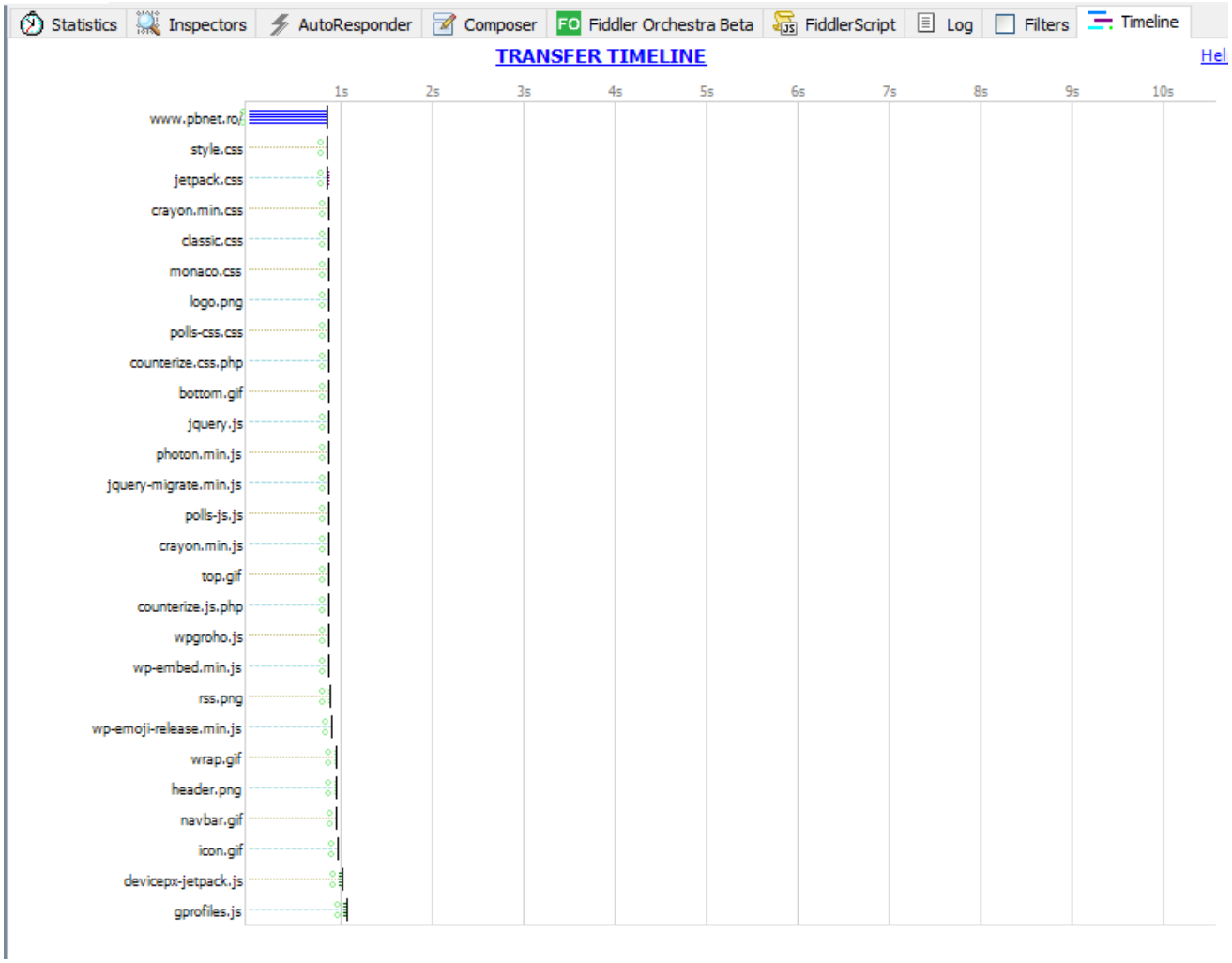
You can also notice that Fiddler uses a different color-coding for each component of a website (like purple for CSS, green for JavaScript, gray for images (PNG/GIF/JPEG).

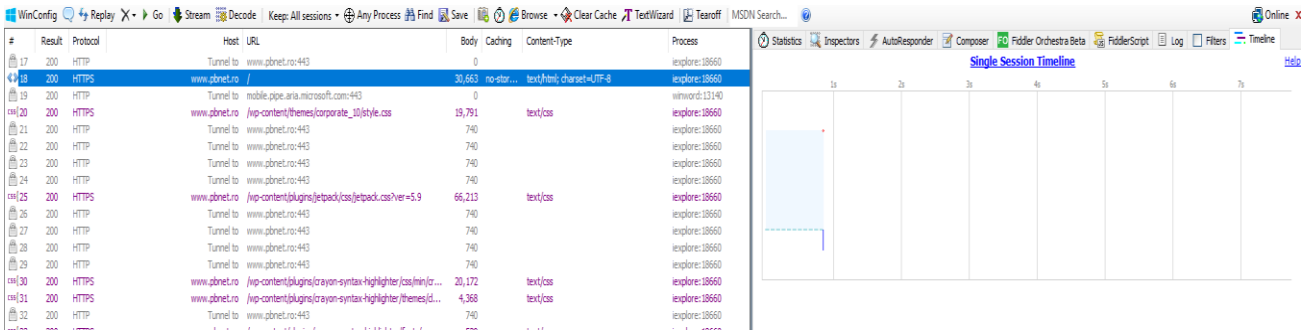
By looking at the statistics tab, we can have a clear view of how the site is loaded:

- For example, we had 44 HTTP 200 requests (also see Request Count)
- A lot of the wait time is due to the JavaScript.
- You can see how many bytes were sent/received.

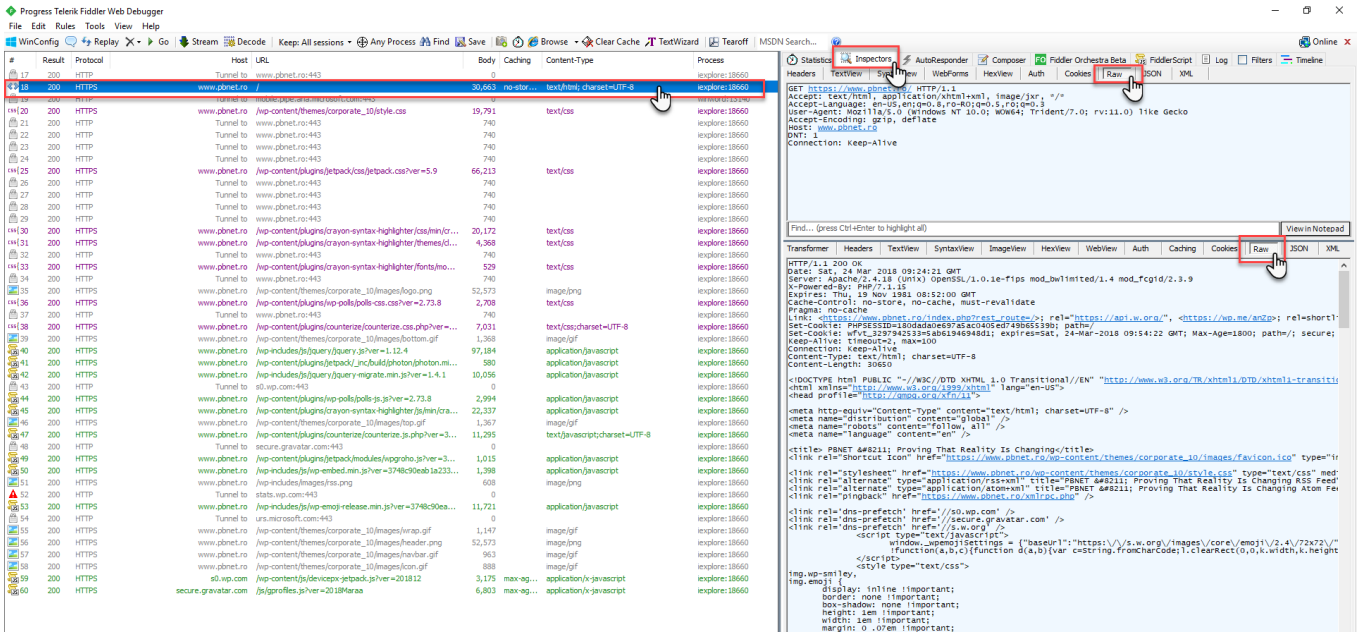


By looking at the TimeLine view, we can see the basic timeline of the page loading and troubleshoot site load performance:

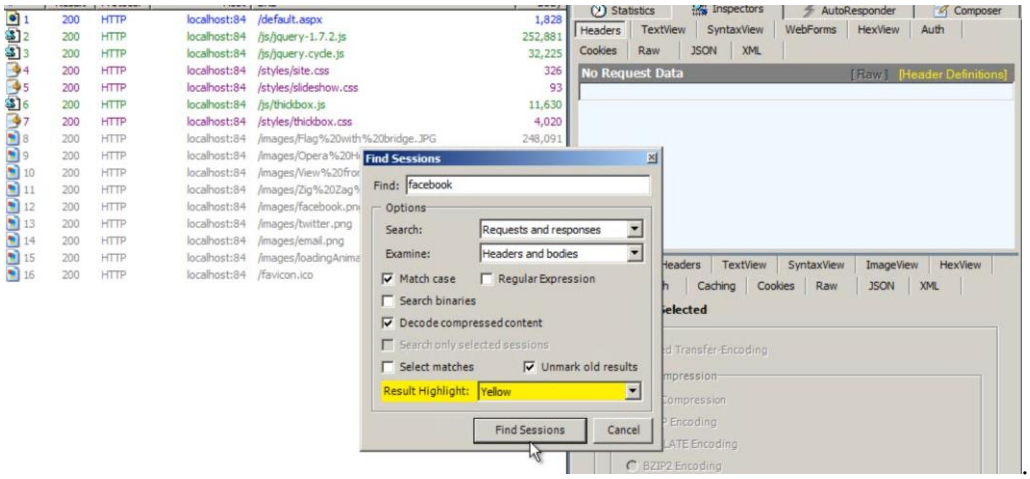

Statistics
Inspectors
AutoResponder
Composer
Fiddler Orchestra Beta
FiddlerScript
Log
Filters
Timeline
TRANSFER TIMELINE
Hel
1s
2s
3s
4s
5s
6s
7s
8s
9s
10s
www.pbnet.ro/
style.css
jetpack.css
crayon.min.css
classic.css
monaco.css
logo.png
polls-css.css
counterize.css.php
bottom.gif
jquery.js
photon.min.js
jquery-migrate.min.js
polls-js.js
crayon.min.js
top.gif
counterize.js.php
wpgroho.js
wp-embed.min.js
rss.png
wp-emoji-release.min.js
wrap.gif
header.png
navbar.gif
icon.gif
devicepx-jetpack.js
gprofiles.js

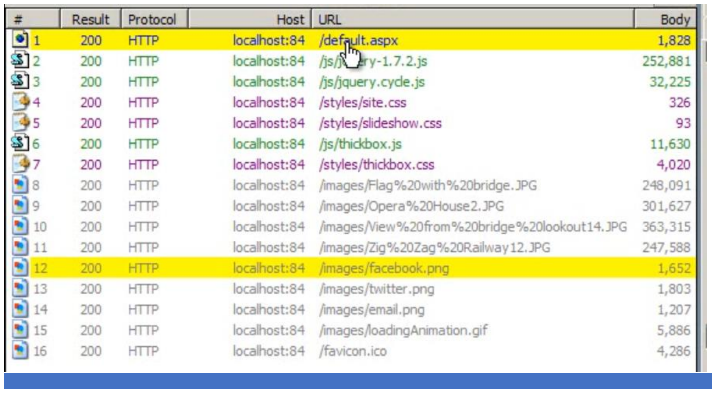

WinConfig
Replay
Go
Stream
Decode
Keep: All sessions
Any Process
Find
Save
Browse
Clear Cache
TextWizard
Tearoff
MSDN Search...
Online
Statistics
Inspectors
AutoResponder
Composer
Fiddler Orchestra Beta
FiddlerScript
Log
Filters
Timeline
Single Session Timeline
Help
Result Protocol Host URL Body Caching Content-Type Process
17 200 HTTP Tunnel to www.pbnet.ro:443 0 iexplore:18660
18 200 HTTPS www.pbnet.ro / 30,663 no-stor... text/html; charset=UTF-8 iexplore:18660
19 200 HTTP Tunnel to mobile.pipe.aria.microsoft.com:443 0 winword:13140
20 200 HTTPS www.pbnet.ro /wp-content/themes/corporate_10/style.css 19,791 text/css iexplore:18660
21 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
22 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
23 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
24 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
25 200 HTTPS www.pbnet.ro /wp-content/plugins/jetpack/css/jetpack.css?ver=5.9 66,213 text/css iexplore:18660
26 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
27 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
28 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
29 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660
30 200 HTTPS www.pbnet.ro /wp-content/plugins/crayon-syntax-highlighter/css/min/cr... 20,172 text/css iexplore:18660
31 200 HTTPS www.pbnet.ro /wp-content/plugins/crayon-syntax-highlighter/themes/cl... 4,368 text/css iexplore:18660
32 200 HTTP Tunnel to www.pbnet.ro:443 740 iexplore:18660

To troubleshoot a single request, go to the Inspectors Tab and use the RAW view:



Searching within URLs, Requests, Responses:

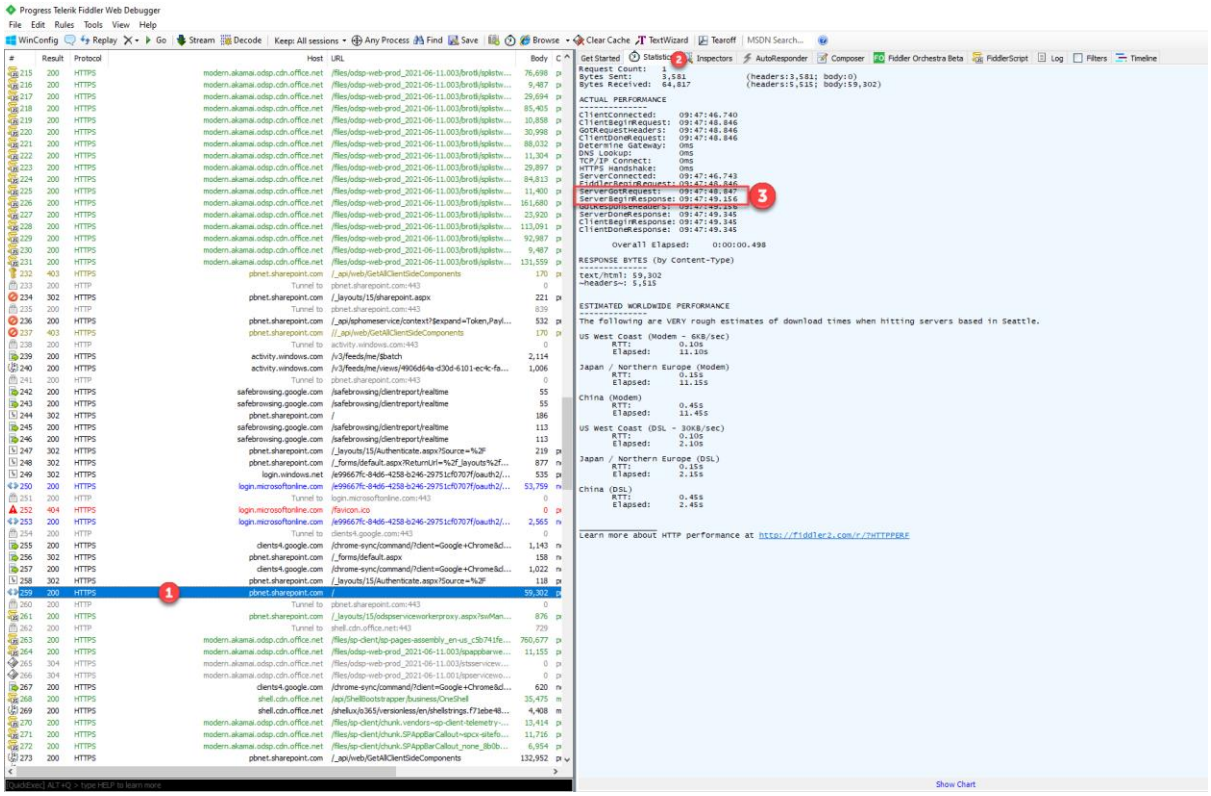




#	Result	Protocol	Host	URL	Body
1	200	HTTP	localhost:84	/default.aspx	1,828
2	200	HTTP	localhost:84	/js/jquery-1.7.2.js	252,881
3	200	HTTP	localhost:84	/js/jquery.cycle.js	32,225
4	200	HTTP	localhost:84	/styles/site.css	326
5	200	HTTP	localhost:84	/styles/slideshow.css	93
6	200	HTTP	localhost:84	/js/thickbox.js	11,630
7	200	HTTP	localhost:84	/styles/thickbox.css	4,020
8	200	HTTP	localhost:84	/images/Flag%20with%20bridge.JPG	248,091
9	200	HTTP	localhost:84	/images/Opera%20House2.JPG	301,627
10	200	HTTP	localhost:84	/images/View%20from%20bridge%20lookout14.JPG	363,315
11	200	HTTP	localhost:84	/images/Zig%20Zag%20Railway12.JPG	247,588
12	200	HTTP	localhost:84	/images/facebook.png	1,652
13	200	HTTP	localhost:84	/images/twitter.png	1,803
14	200	HTTP	localhost:84	/images/email.png	1,207
15	200	HTTP	localhost:84	/images/loadingAnimation.gif	5,886
16	200	HTTP	localhost:84	/favicon.ico	4,286

For SharePoint servers, we can also see how much time it took for the request to be processed by the server:

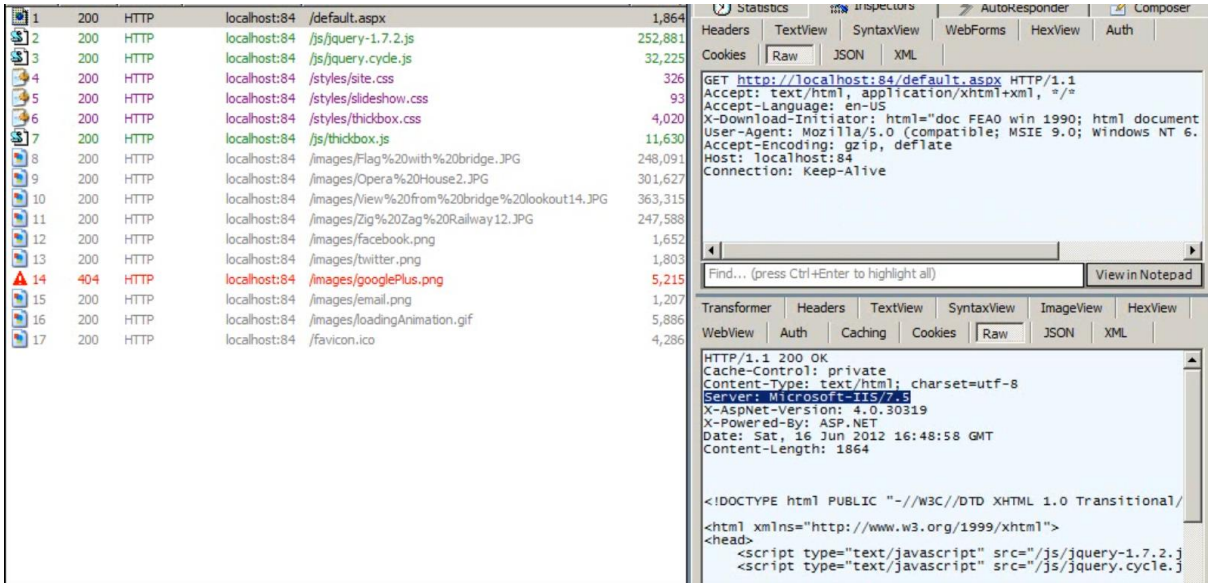
- Choose your request (e.g., accessing a SharePoint online site collection)
- Select the “Statistics” tab
- Look for the actual performance section.
- See the difference between “ServerGotRequest” and “ServerBeginResponse”



3.2 Performing a site review.

How to perform a site review

- Determine the technologies used by the site.
  - Response headers: ASP.NET/IIS/Apache/PHP
  - Source code (e.g. DIV vs TABLES)
- The quality of the site
  - HTTP Status codes (e.g., 404, 500)
- Number of unique domains used.
- 3rd party services used (like CDNs or other content providers).



Opening the response in Notepad we can see more:

Notice the IIS version, ASP Version, and the use of JQuery.

As for the site quality, notice that we have an HTTP 404:

1	200	HTTP	localhost:84	/default.aspx	1,864
2	200	HTTP	localhost:84	/js/jquery-1.7.2.js	252,881
3	200	HTTP	localhost:84	/js/jquery.cycle.js	32,225
4	200	HTTP	localhost:84	/styles/site.css	326
5	200	HTTP	localhost:84	/styles/slideshow.css	93
6	200	HTTP	localhost:84	/styles/thickbox.css	4,020
7	200	HTTP	localhost:84	/js/thickbox.js	11,630
8	200	HTTP	localhost:84	/images/Flag%20with%20bridge.JPG	248,091
9	200	HTTP	localhost:84	/images/Opera%20House2.JPG	301,627
10	200	HTTP	localhost:84	/images/View%20from%20bridge%20lookout14.JPG	363,315
11	200	HTTP	localhost:84	/images/Zig%20Zag%20Railway12.JPG	247,588
12	200	HTTP	localhost:84	/images/facebook.png	1,652
13	200	HTTP	localhost:84	/images/twitter.png	1,803
14	404	HTTP	localhost:84	/images/googlePlus.png	5,215
15	200	HTTP	localhost:84	/images/email.png	1,207
16	200	HTTP	localhost:84	/images/loadingAnimation.gif	5,886
17	200	HTTP	localhost:84	/favicon.ico	4,286

As for the CDNs/3rd party services used, we can see the use of WordPress (wp.com) and gravatar.

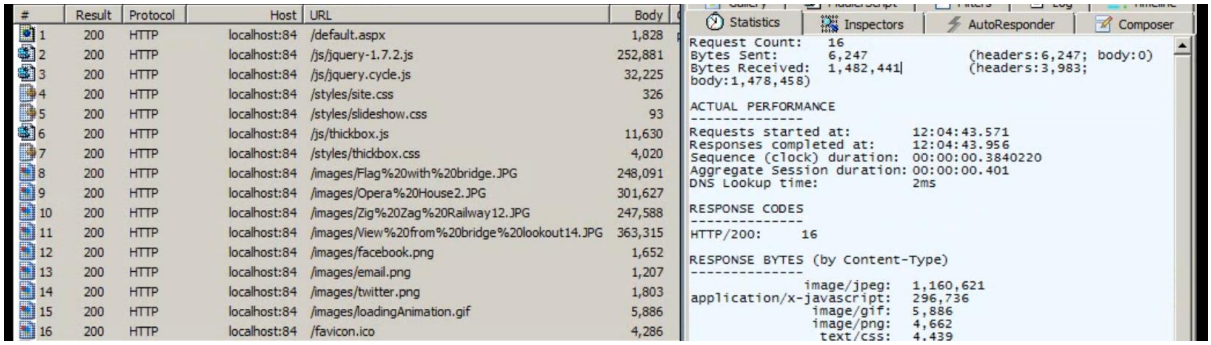
```
REQUESTS PER HOST
-----------------
                    www.pbnet.ro: 37
                       s0.wp.com: 2
              secure.gravatar.com: 2
mobile.pipe.aria.microsoft.com: 1
                    stats.wp.com: 1
               urs.microsoft.com: 1
```

3.3 Site Performance Evaluation

Performance Evaluation

- Statistics tab
  - Request Count
  - Bytes Received (aka how big the page is)

Before

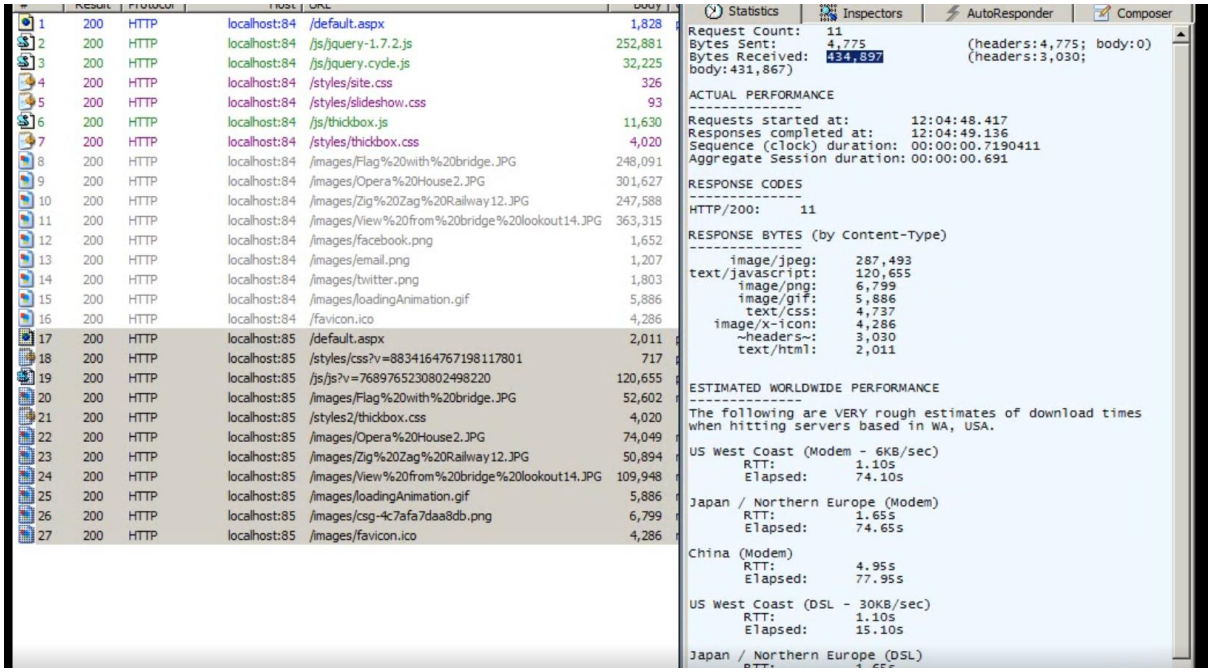


#	Result	Protocol	Host	URL	Body
1	200	HTTP	localhost:84	/default.aspx	1,828
2	200	HTTP	localhost:84	/js/jquery-1.7.2.js	252,881
3	200	HTTP	localhost:84	/js/jquery.cycle.js	32,225
4	200	HTTP	localhost:84	/styles/site.css	326
5	200	HTTP	localhost:84	/styles/slideshow.css	93
6	200	HTTP	localhost:84	/js/thickbox.js	11,630
7	200	HTTP	localhost:84	/styles/thickbox.css	4,020
8	200	HTTP	localhost:84	/images/Flag%20with%20bridge.JPG	248,091
9	200	HTTP	localhost:84	/images/Opera%20House2.JPG	301,627
10	200	HTTP	localhost:84	/images/Zig%20Zag%20Railway12.JPG	247,588
11	200	HTTP	localhost:84	/images/View%20from%20bridge%20lookout14.JPG	363,315
12	200	HTTP	localhost:84	/images/facebook.png	1,652
13	200	HTTP	localhost:84	/images/email.png	1,207
14	200	HTTP	localhost:84	/images/twitter.png	1,803
15	200	HTTP	localhost:84	/images/loadingAnimation.gif	5,886
16	200	HTTP	localhost:84	/favicon.ico	4,286

Statistics | Inspectors | AutoResponder | Composer

```
Request Count:   16
Bytes Sent:      6,247          (headers:6,247; body:0)
Bytes Received:  1,482,441      (headers:3,983;
body:1,478,458)

ACTUAL PERFORMANCE
--------------
Requests started at:        12:04:43.571
Responses completed at:     12:04:43.956
Sequence (clock) duration:  00:00:00.3840220
Aggregate Session duration: 00:00:00.401
DNS Lookup time:            2ms

RESPONSE CODES
--------------
HTTP/200:     16

RESPONSE BYTES (by Content-Type)
--------------
                image/jpeg:   1,160,621
application/x-javascript:   296,736
                image/gif:    5,886
                image/png:    4,662
                 text/css:    4,439
```

After

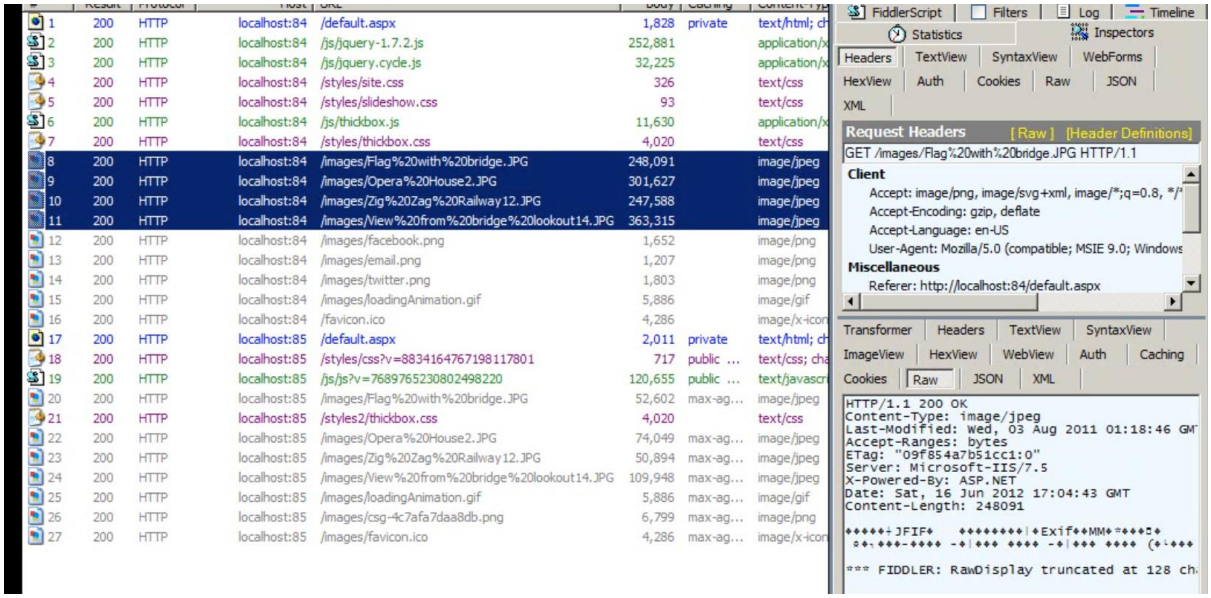


#	Result	Protocol	Host	URL	Body
1	200	HTTP	localhost:84	/default.aspx	1,828
2	200	HTTP	localhost:84	/js/jquery-1.7.2.js	252,881
3	200	HTTP	localhost:84	/js/jquery.cycle.js	32,225
4	200	HTTP	localhost:84	/styles/site.css	326
5	200	HTTP	localhost:84	/styles/slideshow.css	93
6	200	HTTP	localhost:84	/js/thickbox.js	11,630
7	200	HTTP	localhost:84	/styles/thickbox.css	4,020
8	200	HTTP	localhost:84	/images/Flag%20with%20bridge.JPG	248,091
9	200	HTTP	localhost:84	/images/Opera%20House2.JPG	301,627
10	200	HTTP	localhost:84	/images/Zig%20Zag%20Railway12.JPG	247,588
11	200	HTTP	localhost:84	/images/View%20from%20bridge%20lookout14.JPG	363,315
12	200	HTTP	localhost:84	/images/facebook.png	1,652
13	200	HTTP	localhost:84	/images/email.png	1,207
14	200	HTTP	localhost:84	/images/twitter.png	1,803
15	200	HTTP	localhost:84	/images/loadingAnimation.gif	5,886
16	200	HTTP	localhost:84	/favicon.ico	4,286
17	200	HTTP	localhost:85	/default.aspx	2,011
18	200	HTTP	localhost:85	/styles/css?v=8834164767198117801	717
19	200	HTTP	localhost:85	/js/js?v=7689765230802498220	120,655
20	200	HTTP	localhost:85	/images/Flag%20with%20bridge.JPG	52,602
21	200	HTTP	localhost:85	/styles2/thickbox.css	4,020
22	200	HTTP	localhost:85	/images/Opera%20House2.JPG	74,049
23	200	HTTP	localhost:85	/images/Zig%20Zag%20Railway12.JPG	50,894
24	200	HTTP	localhost:85	/images/View%20from%20bridge%20lookout14.JPG	109,948
25	200	HTTP	localhost:85	/images/loadingAnimation.gif	5,886
26	200	HTTP	localhost:85	/images/csg-4c7afa7daa8db.png	6,799
27	200	HTTP	localhost:85	/images/favicon.ico	4,286

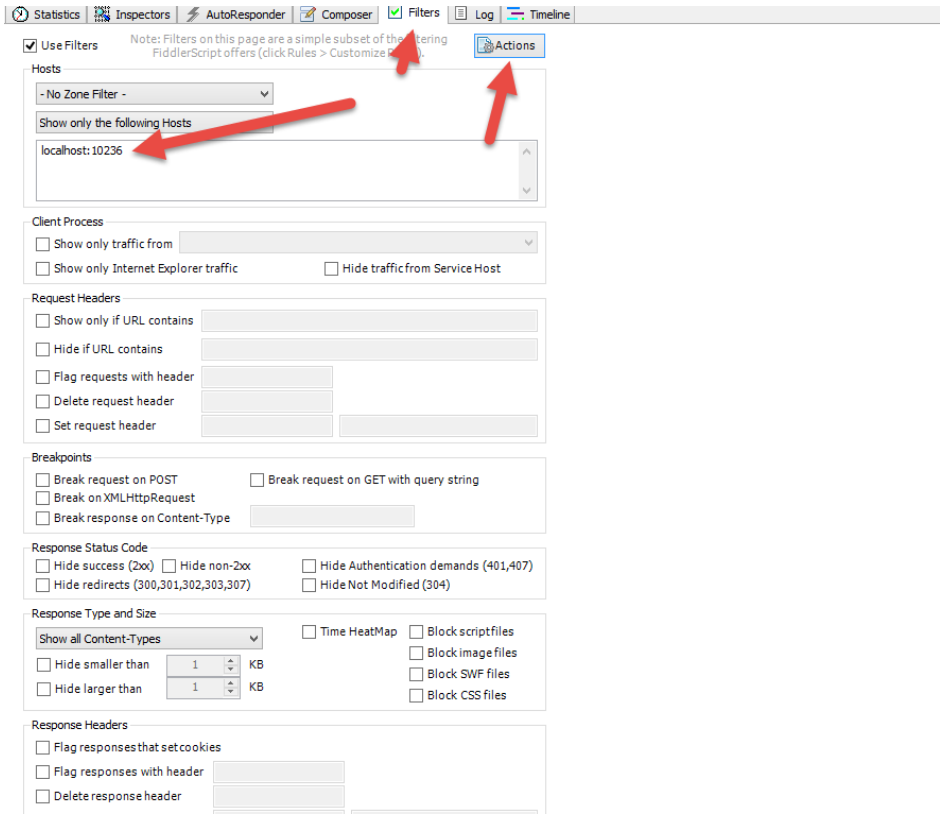
Statistics | Inspectors | AutoResponder | Composer

```
Request Count:   11
Bytes Sent:      4,775          (headers:4,775; body:0)
Bytes Received:  434,897        (headers:3,030;
body:431,867)

ACTUAL PERFORMANCE
--------------
Requests started at:        12:04:48.417
Responses completed at:     12:04:49.136
Sequence (clock) duration:  00:00:00.7190411
Aggregate Session duration: 00:00:00.691

RESPONSE CODES
--------------
HTTP/200:     11

RESPONSE BYTES (by Content-Type)
--------------
      image/jpeg:     287,493
text/javascript:      120,655
      image/png:      6,799
      image/gif:      5,886
       text/css:      4,737
   image/x-icon:      4,286
      ~headers~:      3,030
      text/html:      2,011


ESTIMATED WORLDWIDE PERFORMANCE
--------------
The following are VERY rough estimates of download times
when hitting servers based in WA, USA.

US West Coast (Modem - 6KB/sec)
       RTT:          1.10s
       Elapsed:      74.10s

Japan / Northern Europe (Modem)
       RTT:          1.65s
       Elapsed:      74.65s

China (Modem)
       RTT:          4.95s
       Elapsed:      77.95s

US West Coast (DSL - 30KB/sec)
       RTT:          1.10s
       Elapsed:      15.10s

Japan / Northern Europe (DSL)
```

- Sessions List
  - Body
  - Caching (is it cached or not – e.g., HTTP 304)
  - Consolidation
- Inspectors
  - Compression (see gzip)
  - Minification (CSS/JavaScript whitepages removed)



- Timeline
- Filters (e.g. Heat Maps)



- Always have a before and after file saved!

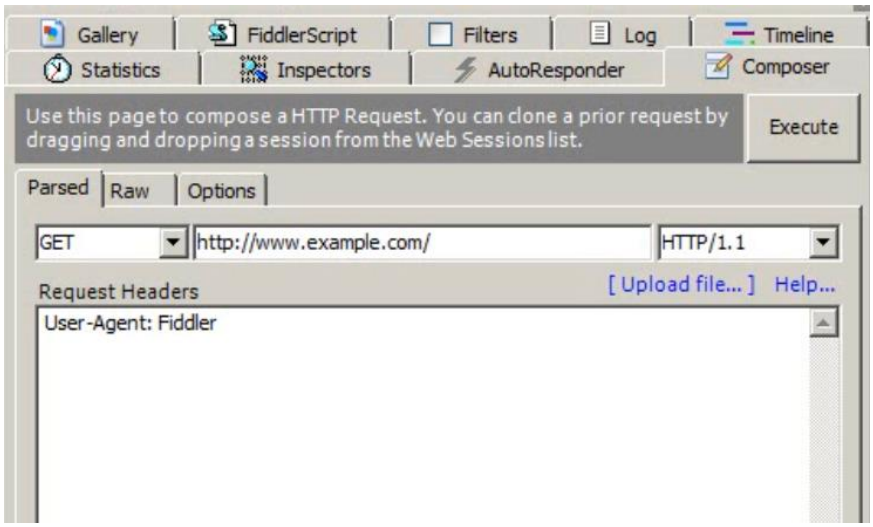
3.4 Modifying a request.

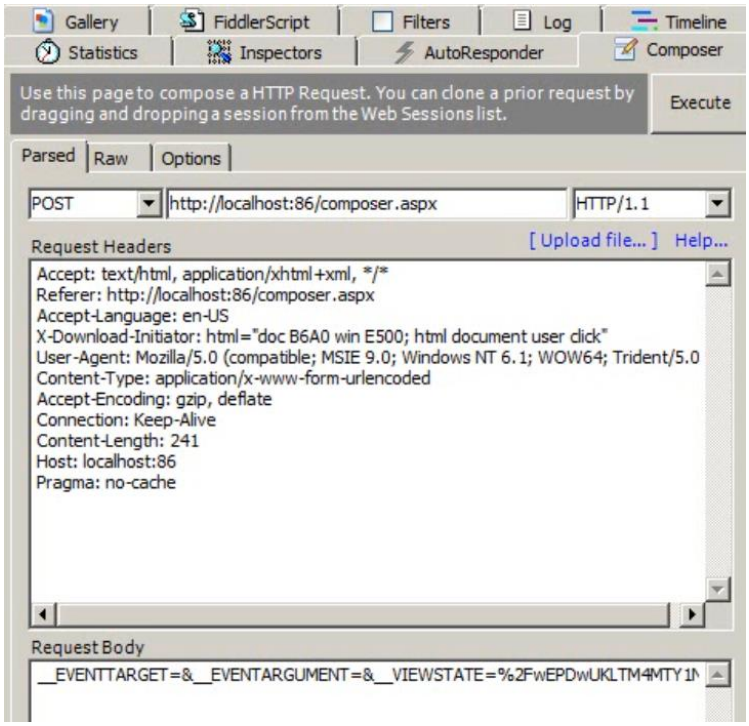
How to modify a request

You can modify a request and send it to the server.

You can change:

- User agents
- Security-related settings
- Language requested
- Compression



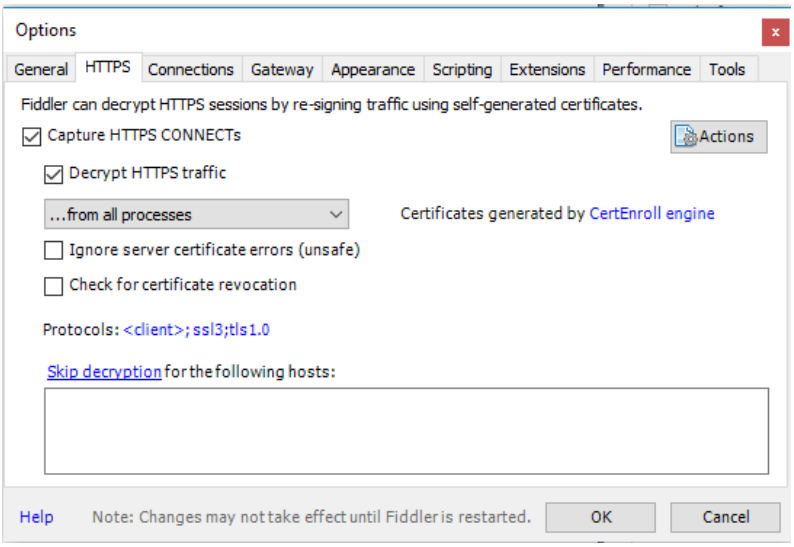


3.5 HTTPS Decryption

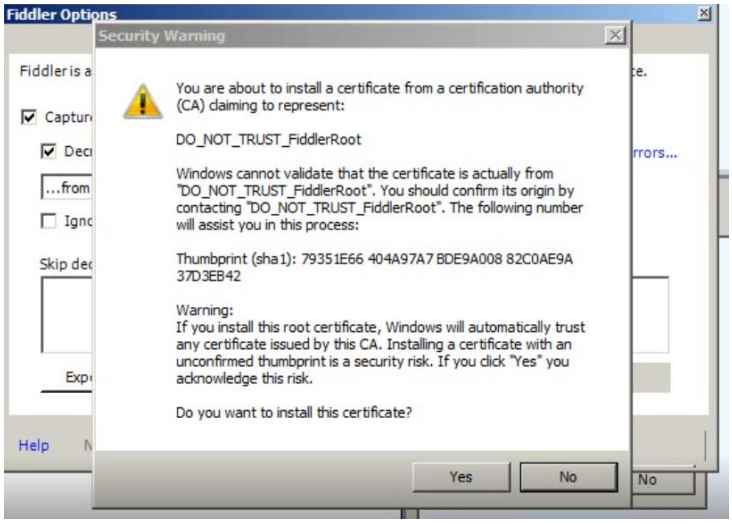
HTTPS Decryption

Fiddler acts as a "man in the middle" and presents its own SSL certificate to the browser. It decrypts the traffic, captures, opens a connection to the remote server and re-encrypts the traffic using that server's certificate.

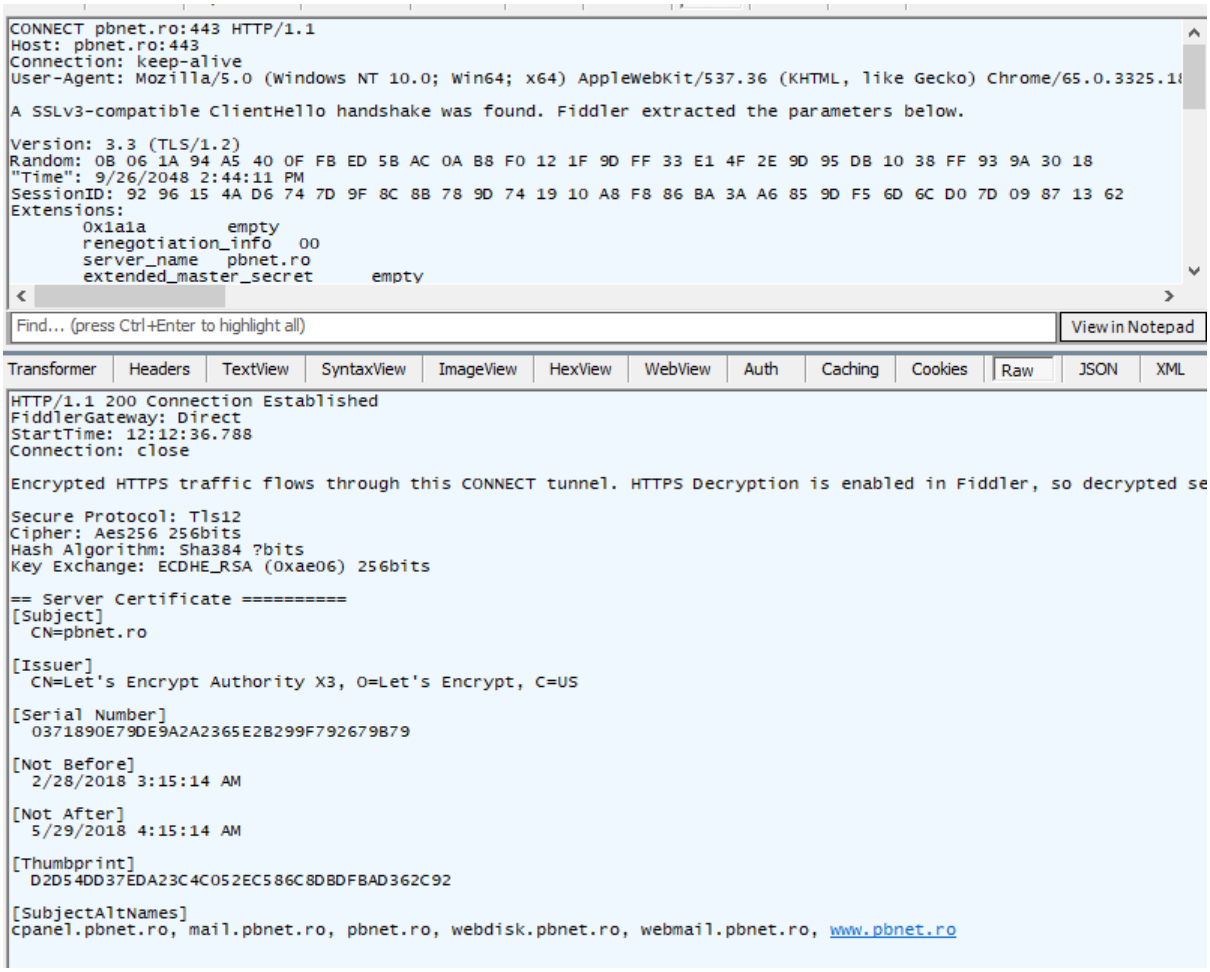
You turn it on: Fiddler Options → HTTPS → Decrypt HTTPS Traffic:



Don't forget to install this certificate, otherwise, fiddler won't be able to decrypt the traffic:

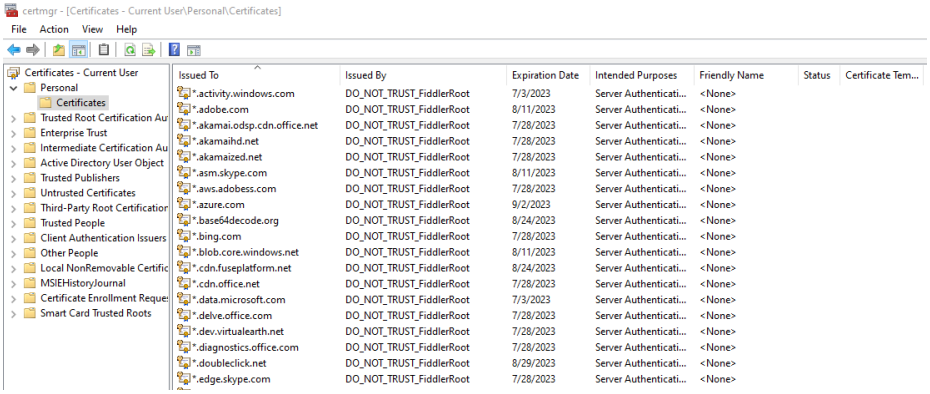


You can see the SSL connection is established:



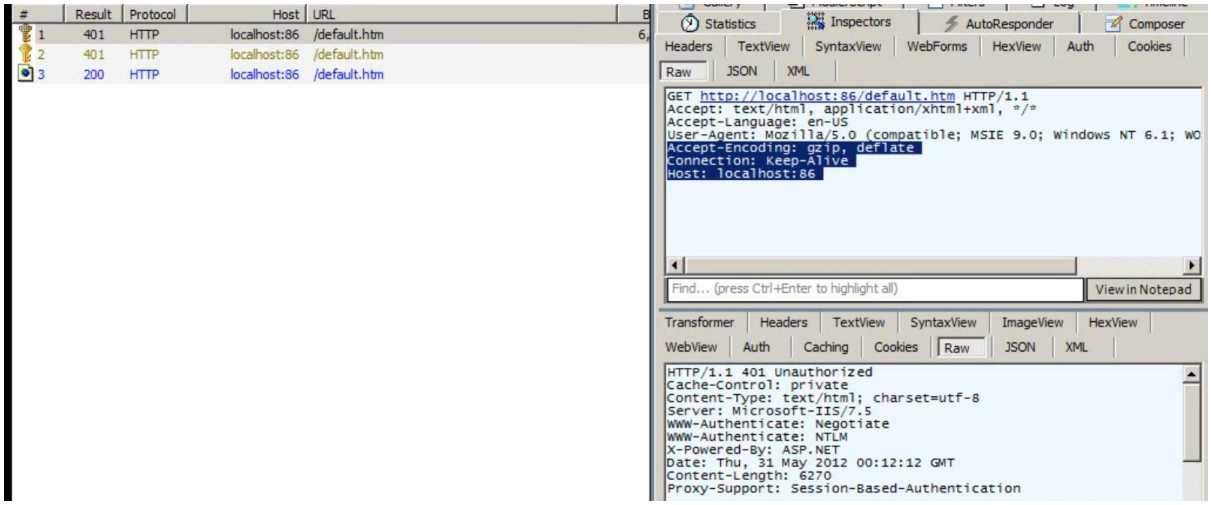
Note: When you uninstall fiddler from a machine, be sure to also remove the Fiddler SSL Certificates from the personal store of the current user.

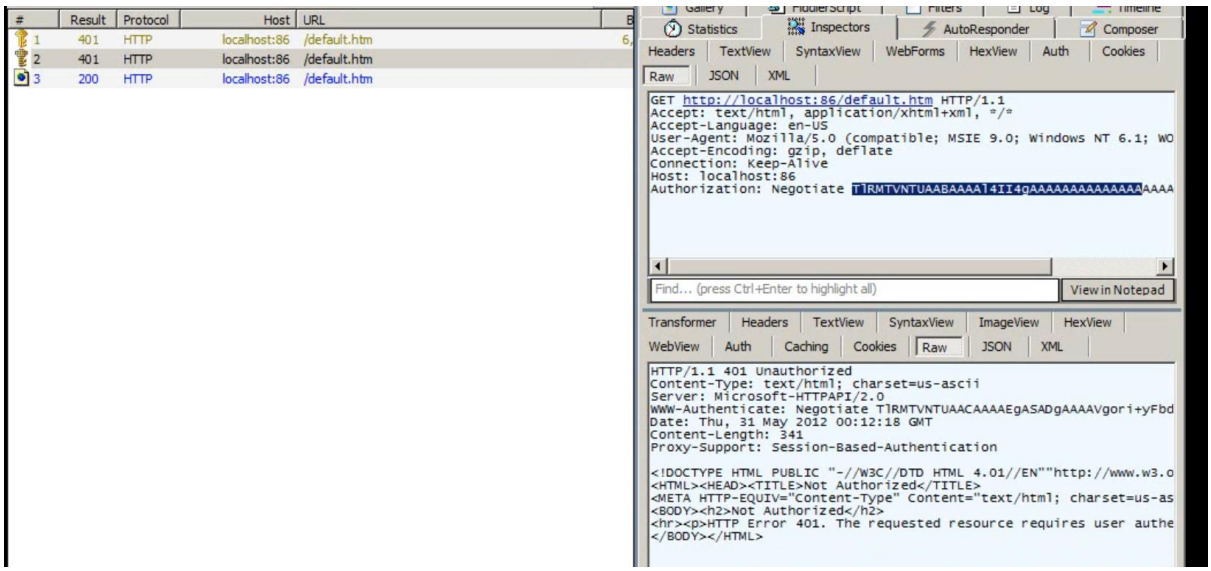
- Run certmgr.msc
- Go to the Personal store for the current user and remove the certificates issued by "DO_NOT_TRUST_FiddlerRoot"

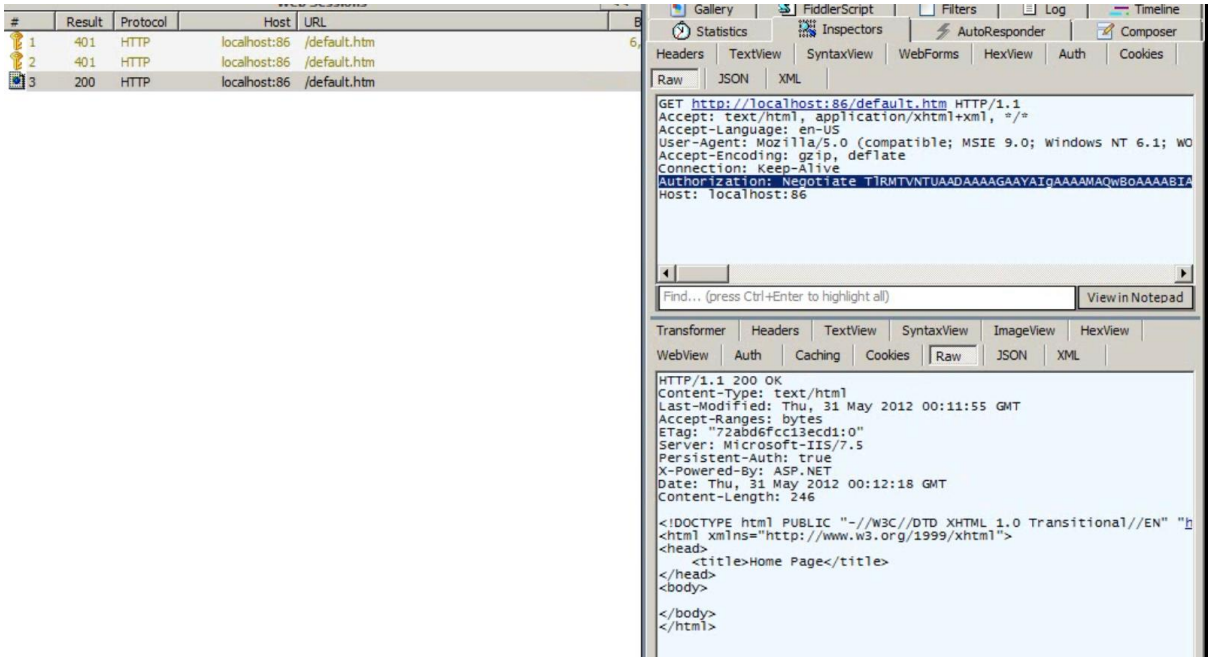


Common HTTP Responses

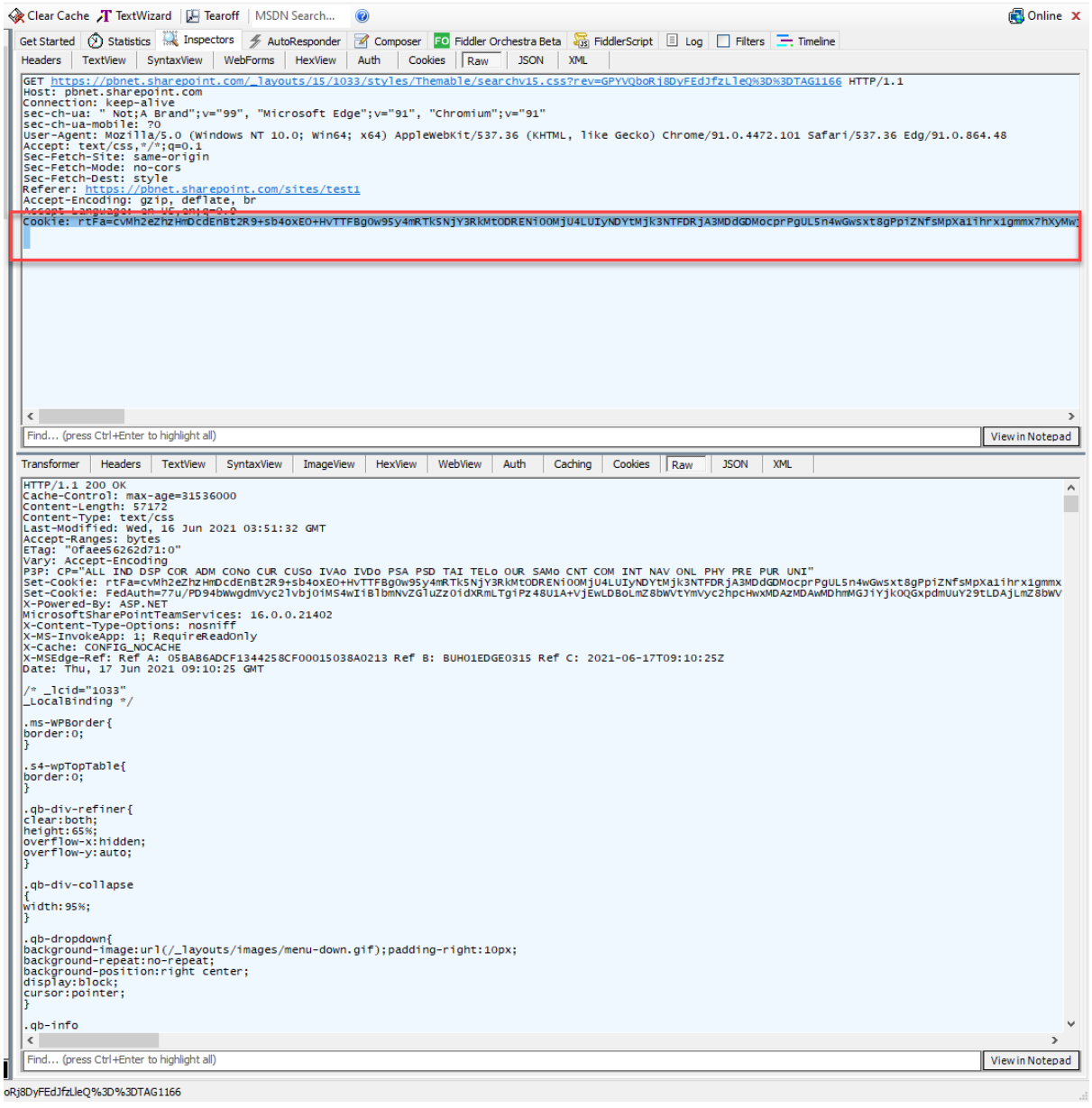
Windows Authentication



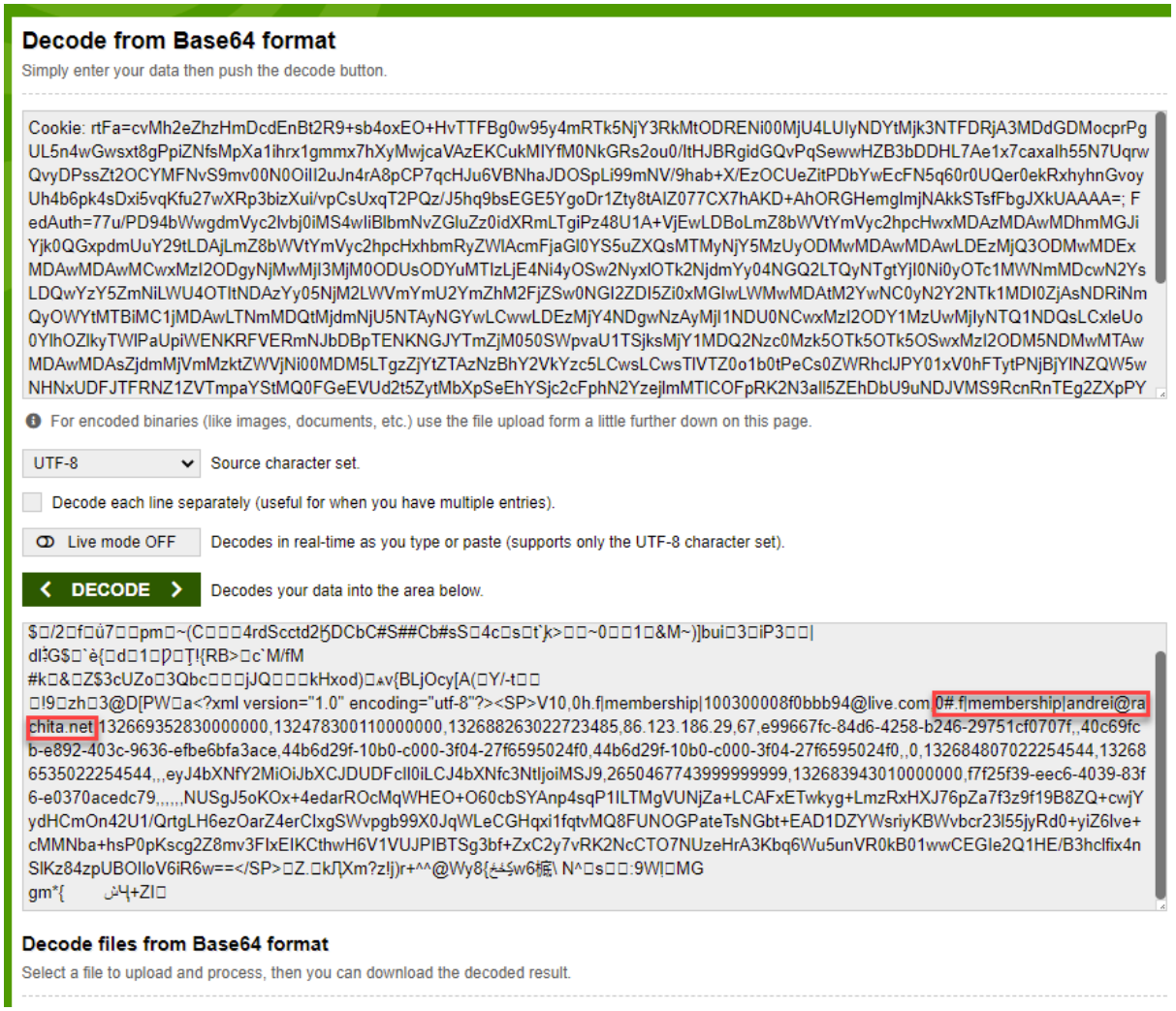



Result
Protocol
Host
URL
1 401 HTTP localhost:86 /default.htm
2 401 HTTP localhost:86 /default.htm
3 200 HTTP localhost:86 /default.htm
Gallery
FiddlerScript
Filters
Log
Timeline
Statistics
Inspectors
AutoResponder
Composer
Headers
TextView
SyntaxView
WebForms
HexView
Auth
Cookies
Raw
JSON
XML
GET http://localhost:86/default.htm HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WO
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Authorization: Negotiate TlRMTVNTUAADAAAAGAAYAIgAAAAMAQwBOAAAABIA
Host: localhost:86
Find... (press Ctrl+Enter to highlight all)
View in Notepad
Transformer
Headers
TextView
SyntaxView
ImageView
HexView
WebView
Auth
Caching
Cookies
Raw
JSON
XML
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Thu, 31 May 2012 00:11:55 GMT
Accept-Ranges: bytes
ETag: "72abd6fcc13ecd1:0"
Server: Microsoft-IIS/7.5
Persistent-Auth: true
X-Powered-By: ASP.NET
Date: Thu, 31 May 2012 00:12:18 GMT
Content-Length: 246
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "h
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Home Page</title>
</head>
<body>
</body>
</html>

SharePoint Online Authentication cookie



You can use a tool like the Base64 decoder: [Base64 Decode and Encode - Online] and decrypt the cookie:



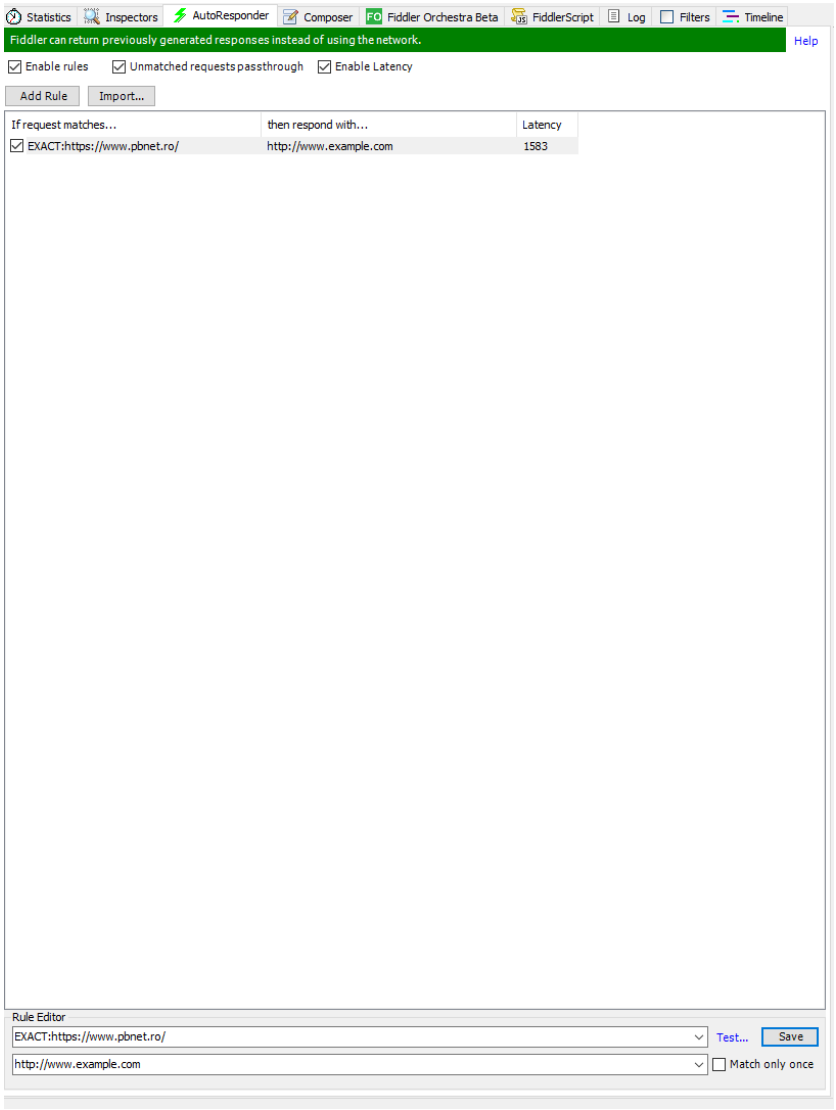
Notice that you are able to see the user who authenticated against the SharePoint site.

3.7 The Fiddler Autoresponder

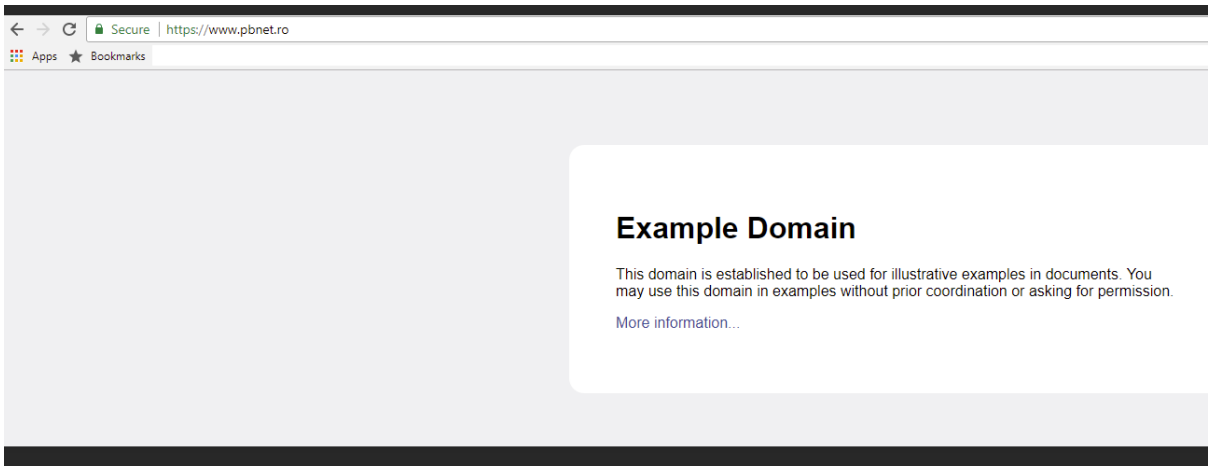
Fiddler Autoresponder

Being a proxy, Fiddler can return information without actually contacting the web server.

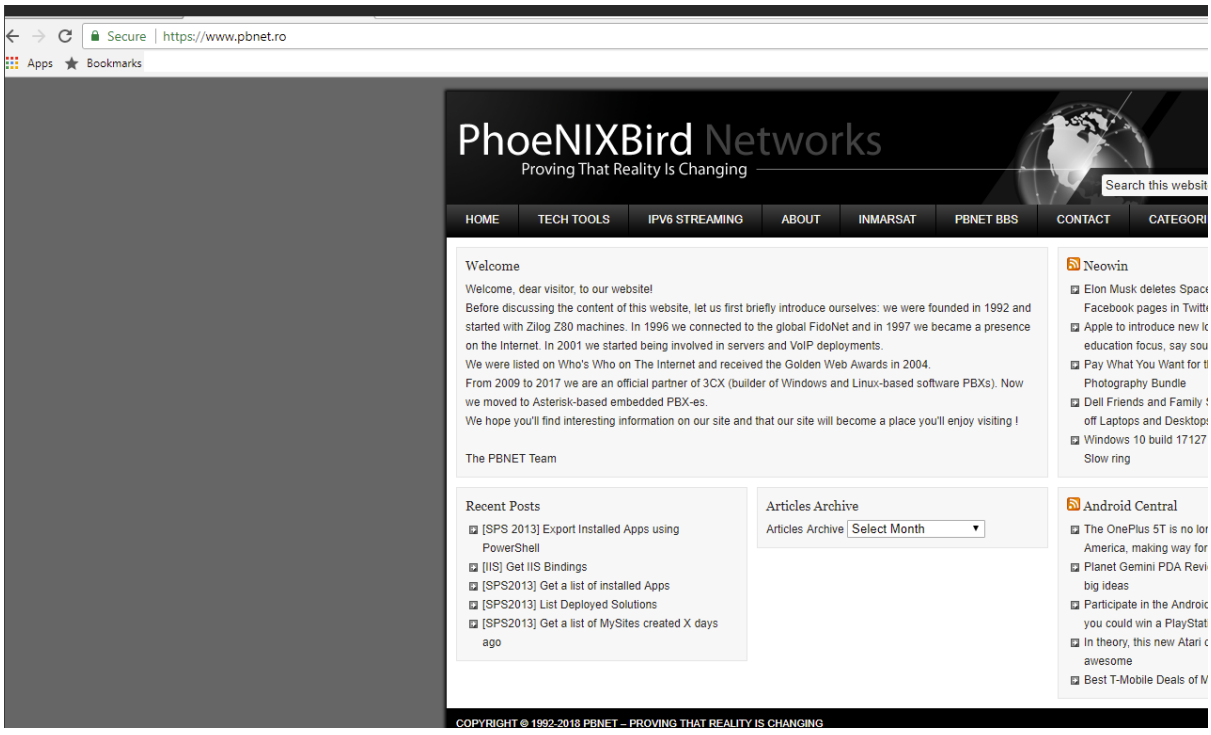
Notice how I'm redirecting the traffic.



And the result



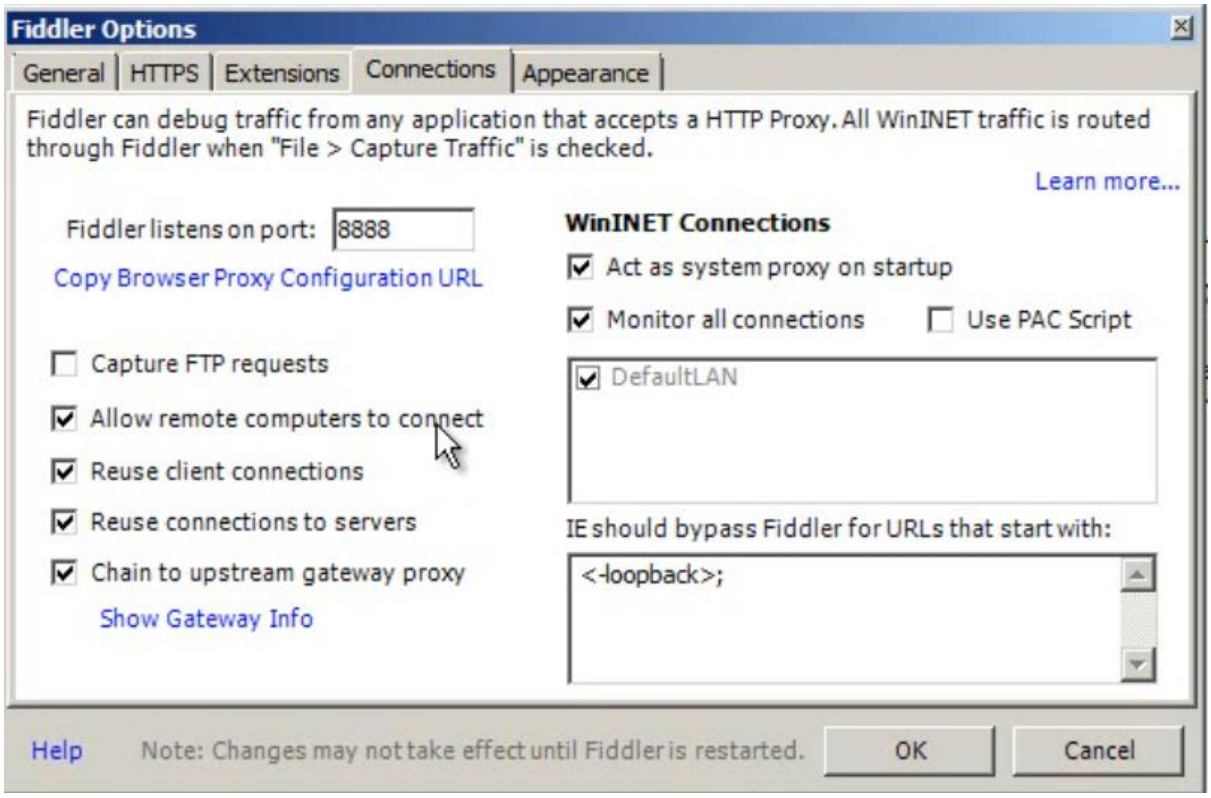
Instead of the original content:

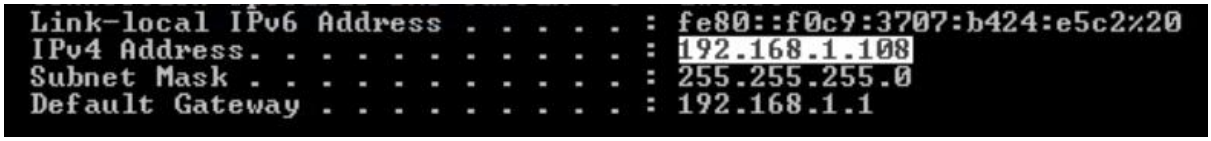


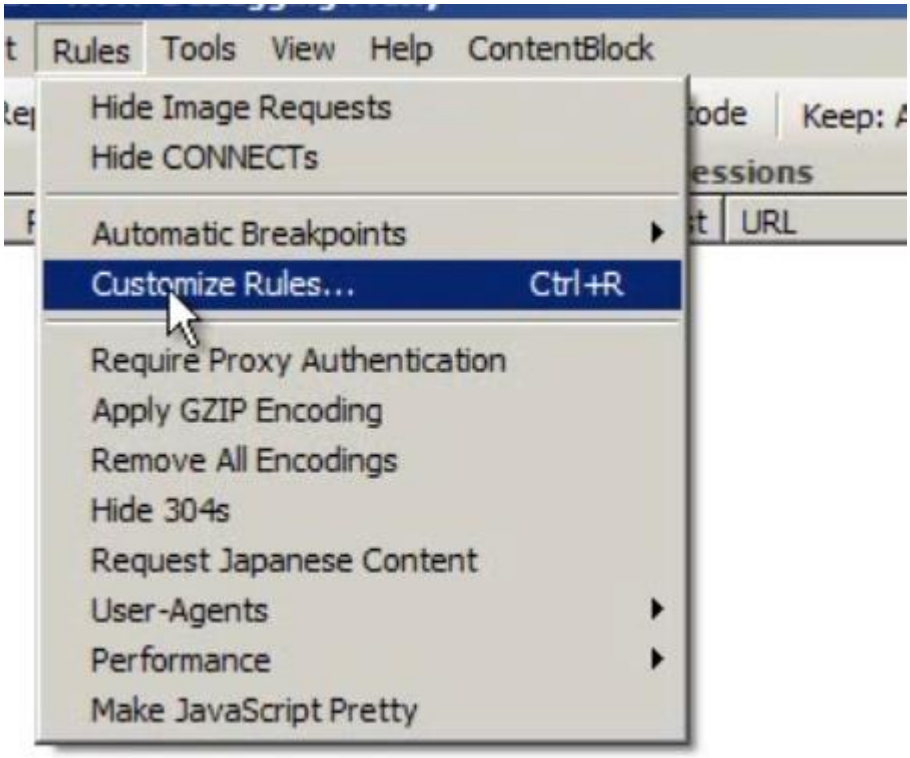
3.8 Fiddler used to capture traffic from phones, tablets, or other platforms.

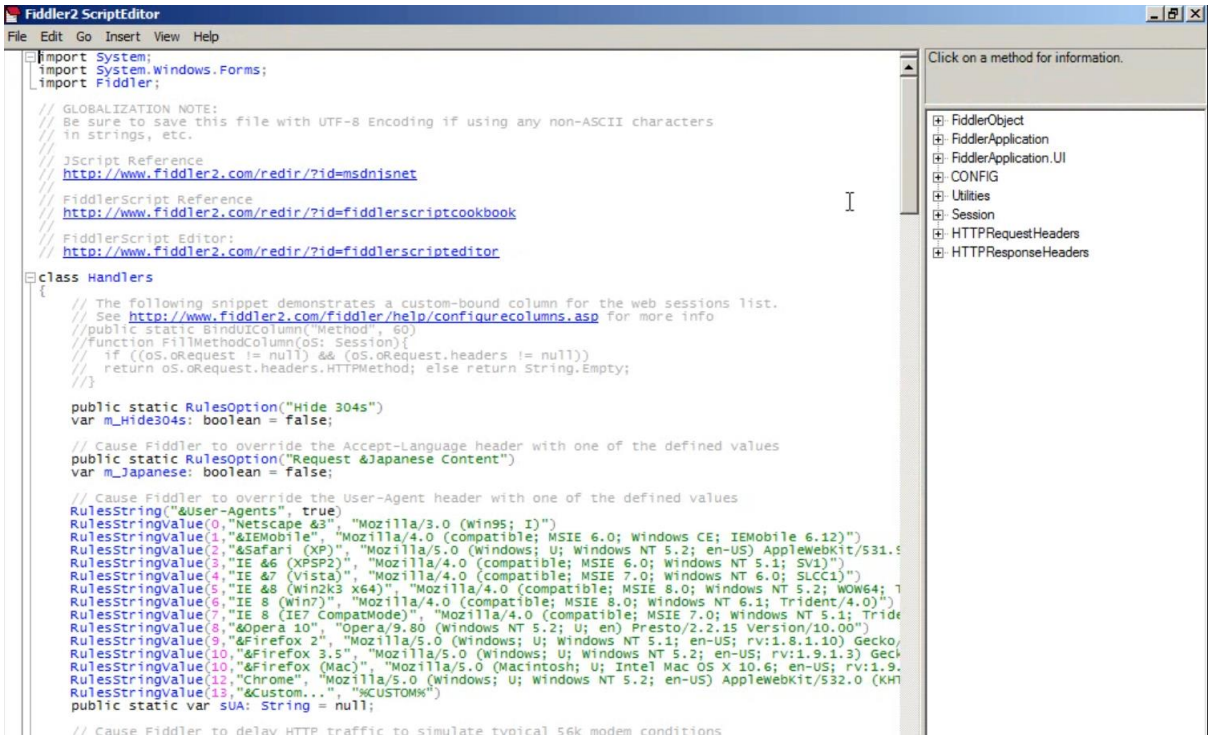
Capture Traffic from phones, tablets, or other platforms

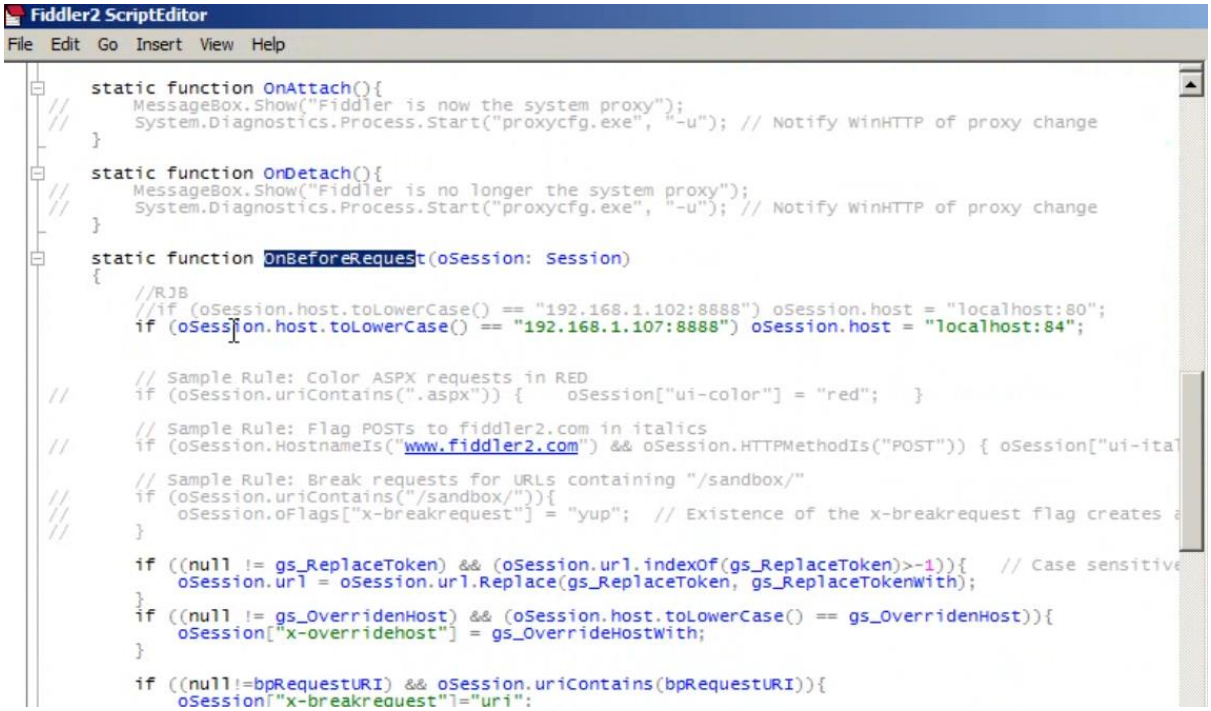
- You can set up Fiddler as a reverse proxy.
  - Run Fiddler on a Windows machine with the default port 8888 (you might also have to enable that port in the firewall)
  - Set Fiddler: Tools → Fiddler Options → Connections → Allow remote computers to connect.
  - Determine the IP of the Windows machine (ipconfig /all)
  - Determine any rules (Rules → Customize Rules)
    - Map incoming IP address to the original web server (local or remote)



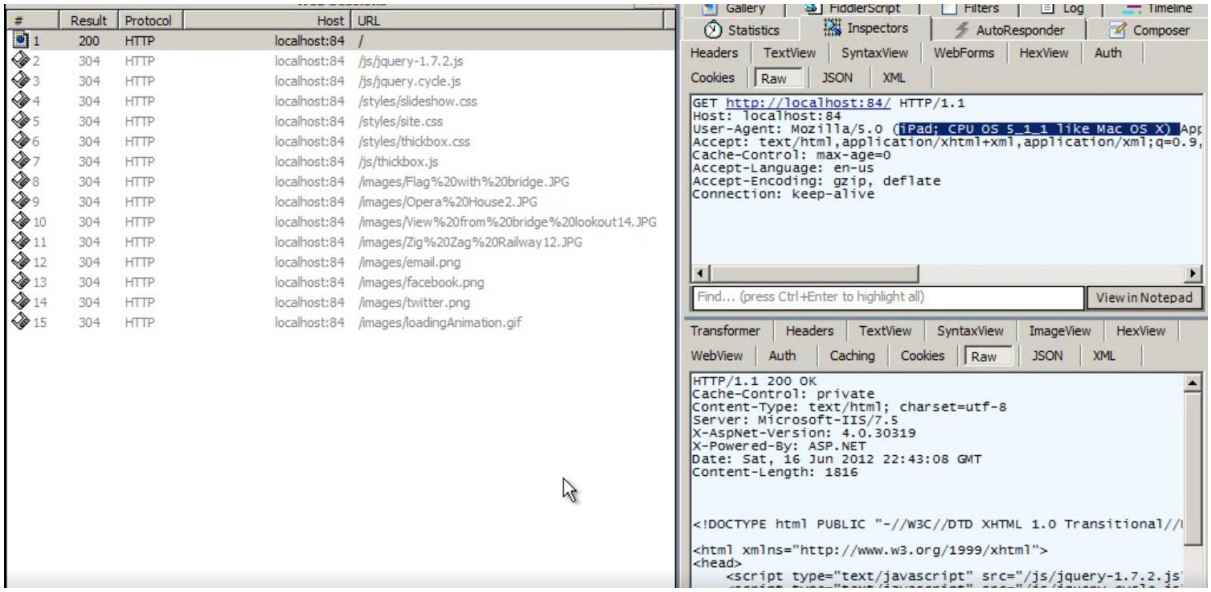



Rules
Tools
View
Help
ContentBlock
Hide Image Requests
Hide CONNECTs
Automatic Breakpoints
Customize Rules... Ctrl+R
Require Proxy Authentication
Apply GZIP Encoding
Remove All Encodings
Hide 304s
Request Japanese Content
User-Agents
Performance
Make JavaScript Pretty
Keep:
URL


Fiddler2 ScriptEditor
File Edit Go Insert View Help
Click on a method for information.
FiddlerObject
FiddlerApplication
FiddlerApplication.UI
CONFIG
Utilities
Session
HTTPRequestHeaders
HTTPResponseHeaders

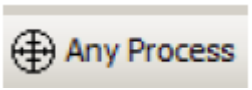


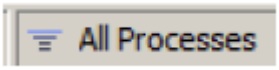
And the results:



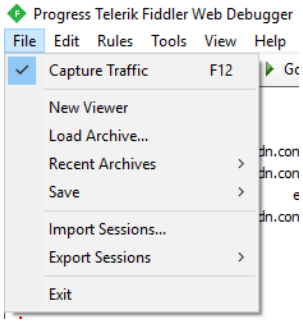
3.9 Common mistakes when using Fiddler

Common Mistakes

- Limited processes
    - Check toolbar 

    - Check status bar 

- Capturing
    - File Menu

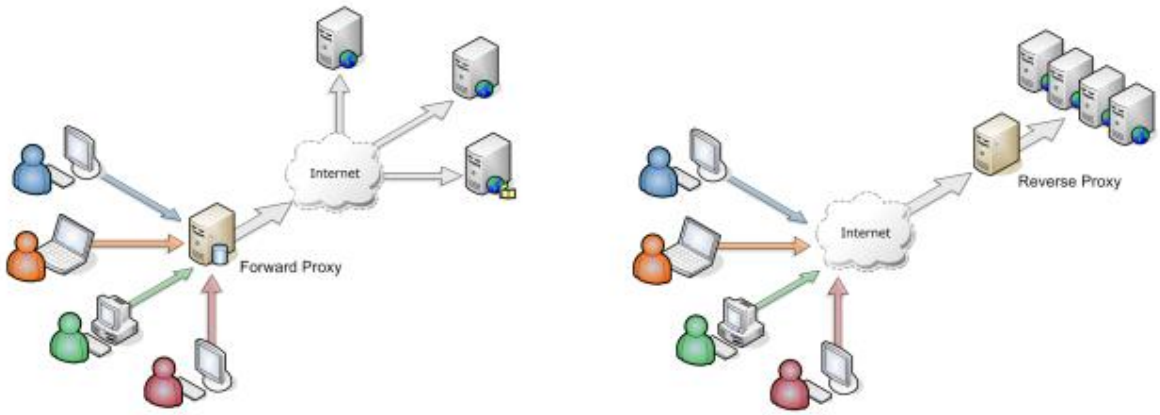


- No proxy set
- Browser does not automatically support using “localhost” or “127.0.0.1”
    - In this case use: ipv4.fiddler or ipv6.fiddler

4. Annexes

4.1 What is a proxy and a reverse proxy?

Proxy (Forward Proxy) Vs. Reverse Proxy



Most discussion of web proxies refers to the type of proxy known as a "forward proxy."

The proxy event in this case is that the "forward proxy" retrieves data from another web site on behalf of the original requestee.

Forward proxies are typically used internally by large organizations, such as universities and enterprises, to:

- Block employees from visiting certain websites.
- Monitor employee online activity
- Block malicious traffic from reaching an origin server.
- Improve the user experience by caching external site content.

A reverse proxy server is an intermediate connection point positioned at a network's edge. It receives initial HTTP connection requests, acting like the actual endpoint.

Essentially your network's traffic cop, the reverse proxy serves as a gateway between users and your application origin server. In so doing it handles all policy management and traffic routing.

A reverse proxy operates by:

- Receiving a user connection request
- Completing a TCP three-way handshake, terminating the initial connection
- Connecting with the origin server and forwarding the original request

4.2 HTTP Status Codes

- **1×× Informational**
- **100** Continue
- **101** Switching Protocols
- **102** Processing
- **2×× Success**
- **200** OK
- **201** Created
- **202** Accepted
- **203** Non-authoritative Information
- **204** No Content
- **205** Reset Content
- **206** Partial Content
- **207** Multi-Status
- **208** Already Reported
- **226** IM Used
- **3×× Redirection**
- **300** Multiple Choices
- **301** Moved Permanently
- **302** Found
- **303** See Other
- **304** Not Modified
- **305** Use Proxy
- **307** Temporary Redirect
- **308** Permanent Redirect
- **4×× Client Error**
- **400** Bad Request
- **401** Unauthorized
- **402** Payment Required
- **403** Forbidden
- **404** Not Found
- **405** Method Not Allowed
- **406** Not Acceptable
- **407** Proxy Authentication Required
- **408** Request Timeout

- **409** Conflict
- **410** Gone
- **411** Length Required
- **412** Precondition Failed
- **413** Payload Too Large
- **414** Request-URI Too Long
- **415** Unsupported Media Type
- **416** Requested Range Not Satisfiable
- **417** Expectation Failed
- **418** I'm a teapot
- **421** Misdirected Request
- **422** Unprocessable Entity
- **423** Locked
- **424** Failed Dependency
- **426** Upgrade Required
- **428** Precondition Required
- **429** Too Many Requests
- **431** Request Header Fields Too Large
- **444** Connection Closed Without Response
- **451** Unavailable For Legal Reasons
- **499** Client Closed Request
- **5×× Server Error**
- **500** Internal Server Error
- **501** Not Implemented
- **502** Bad Gateway
- **503** Service Unavailable
- **504** Gateway Timeout
- **505** HTTP Version Not Supported
- **506** Variant Also Negotiates
- **507** Insufficient Storage
- **508** Loop Detected
- **510** Not Extended
- **511** Network Authentication Required
- **599** Network Connect Timeout Error

4.3 HTTP Verbs

HTTP defines a set of request methods to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as HTTP verbs.

RFC 2616

OPTIONS

GET

HEAD

POST

PUT

DELETE

TRACE

CONNECT

RFC 2518

PROPFIND

PROPPATCH

MKCOL

COPY

MOVE

LOCK

UNLOCK

RFC 3253

VERSION-CONTROL

REPORT

CHECKOUT

CHECKIN

UNCHECKOUT

MKWORKSPACE

UPDATE

LABEL

MERGE

BASELINE-CONTROL

MKACTIVITY

RFC 3648

ORDERPATCH

RFC 3744

ACL

draft-dusseault-http-patch

PATCH

draft-reschke-webdav-search

SEARCH

4.4 WebDAV Methods

WebDAV (Web Distributed Authoring and Versioning) is an extension of the Hypertext Transfer Protocol (HTTP) that allows clients to perform remote Web content authoring operations.

BCOPY Method
BDELETE Method
BMOVE Method
BPROPFIND Method
BPROPPATCH Method
COPY Method
DELETE Method
LOCK Method
MKCOL Method
MOVE Method
NOTIFY Method
POLL Method
PROPFIND Method
PROPPATCH Method
SEARCH Method
SUBSCRIBE Method
UNLOCK Method
UNSUBSCRIBE Method
X-MS-ENUMATTS Method

4.5 Set an HTTP Proxy using PowerShell

Set HTTP Proxy via PowerShell

This function will edit the values at
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings" as below.

It enables the Proxy if disabled by default.

SYNTAX

```
Set-InternetProxy [-Proxy] <string[]> [[-acs] <string[]>]  [<CommonParameters>]
```

```
<#
.Synopsis
This function will set the proxy settings provided as input to the cmdlet.
.Description
This function will set the proxy server and (optional) Automatic configuration script.
.Parameter ProxyServer
This parameter is set as the proxy for the system.
Data from. This parameter is Mandatory
.Example
Setting proxy information
Set-InternetProxy -proxy "proxy:7890"
.Example
Setting proxy information and (optional) Automatic Configuration Script
Set-InternetProxy -proxy "proxy:7890" -acs "http://proxy:7892"
#>


Function Set-InternetProxy
{
   [CmdletBinding()]
   Param(


[Parameter(Mandatory=$True,ValueFromPipeline=$true,ValueFromPipelineByPropertyNam
e=$true)]
       [String[]]$Proxy,


[Parameter(Mandatory=$False,ValueFromPipeline=$true,ValueFromPipelineByPropertyNa
me=$true)]
       [AllowEmptyString()]
       [String[]]$acs

   )

   Begin
   {

          $regKey="HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings"

   }

   Process
   {

       Set-ItemProperty -path $regKey ProxyEnable -value 1
```

```
        Set-ItemProperty -path $regKey ProxyServer -value $proxy

        if($acs)
        {

                Set-ItemProperty -path $regKey AutoConfigURL -Value $acs
        }

    }

    End
    {

        Write-Output "Proxy is now enabled"

        Write-Output "Proxy Server : $proxy"

        if ($acs)
        {

            Write-Output "Automatic Configuration Script : $acs"

        }
        else
        {

            Write-Output "Automatic Configuration Script : Not Defined"

        }
    }
}
```

How to use the script:

```
<#
.Synopsis
This function will set the proxy settings provided as input to the cmdlet.
.Description
This function will set the proxy server and (optinal) Automatic configuration script.
.Parameter ProxyServer
This parameter is set as the proxy for the system.
Data from. This parameter is Mandatory
.Example
Setting proxy information
Set-InternetProxy -proxy "proxy:7890"
.Example
```

```
Setting proxy information and (optinal) Automatic Configuration Script
Set-InternetProxy -proxy "proxy:7890" -acs "http://proxy:7892"
#>
```

4.6 Set an HTTP Proxy using NETSH

Set a HTTP Proxy using NETSH

To navigate to the WinHTTP context, open an administrator Command Prompt window, type **netsh**, and then type **winhttp**.

```
C:\Windows\system32>netsh
netsh>winhttp
netsh winhttp>
```

You use the **set proxy** command to configure the proxy settings. You can type the command followed by a question mark to see the syntax for this command.

```
netsh winhttp>set proxy /?
```

This example specifies that HTTP servers and HTTPS servers are accessed through the proxy server proxy_server, except for host names that don't contain a period specified by the "<local>" argument.

```
netsh winhttp>set proxy proxy_server "<local>"
```

This example imports proxy information used by Internet Explorer by using the **import proxy** command.

```
netsh winhttp>import proxy source=ie
```

These examples use the **reset proxy** command to reset the WinHTTP proxy to DIRECT.

```
netsh winhttp>reset proxy
```

Even if you aren't running a proxy server, we recommend that you use Netsh.exe to check whether a previous proxy has been set. By running the tool without arguments, this example shows the current configuration.

```
netsh winhttp>show proxy
```

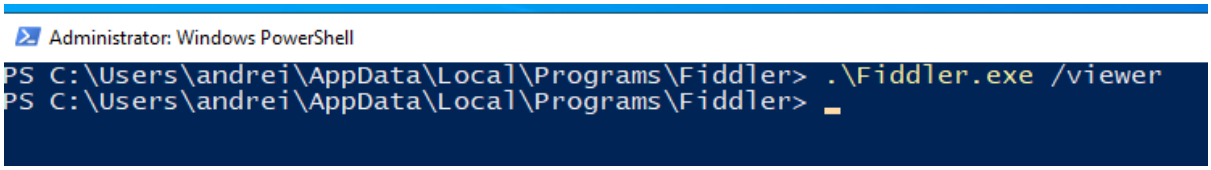
4.7 Examine a HAR trace with Fiddler.

HAR, short for HTTP Archive, is a format used for tracking information between a web browser and a website. A HAR file is primarily used for identifying performance issues, such as bottlenecks and slow load times, and page rendering problems.

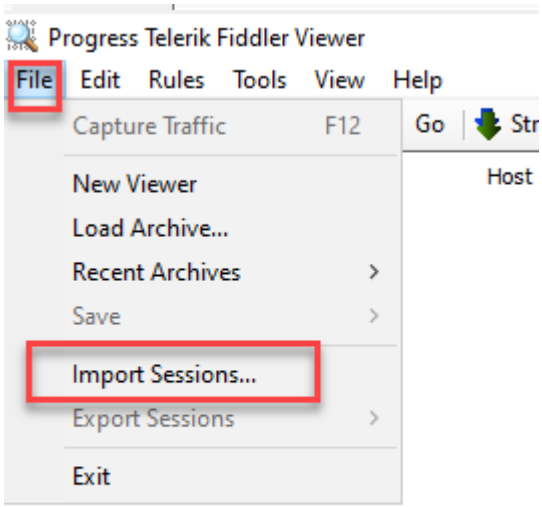
Fiddler can also be used to analyze HAR traces taken with Google Chrome or Microsoft Edge.

More info on capturing the trace can be found here: https://docs.microsoft.com/en-us/azure/azure-portal/capture-browser-trace

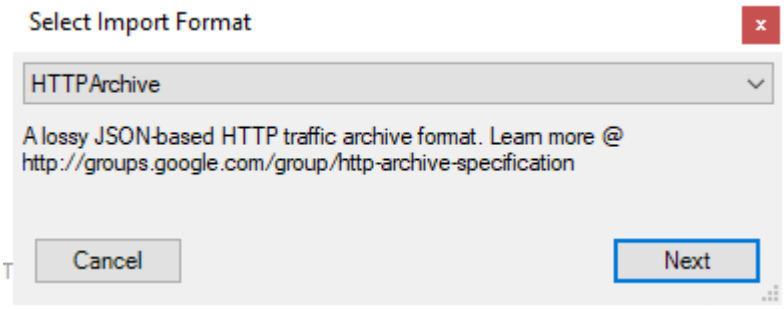
- Open Fiddler in viewer mode (running fiddler.exe /viewer)



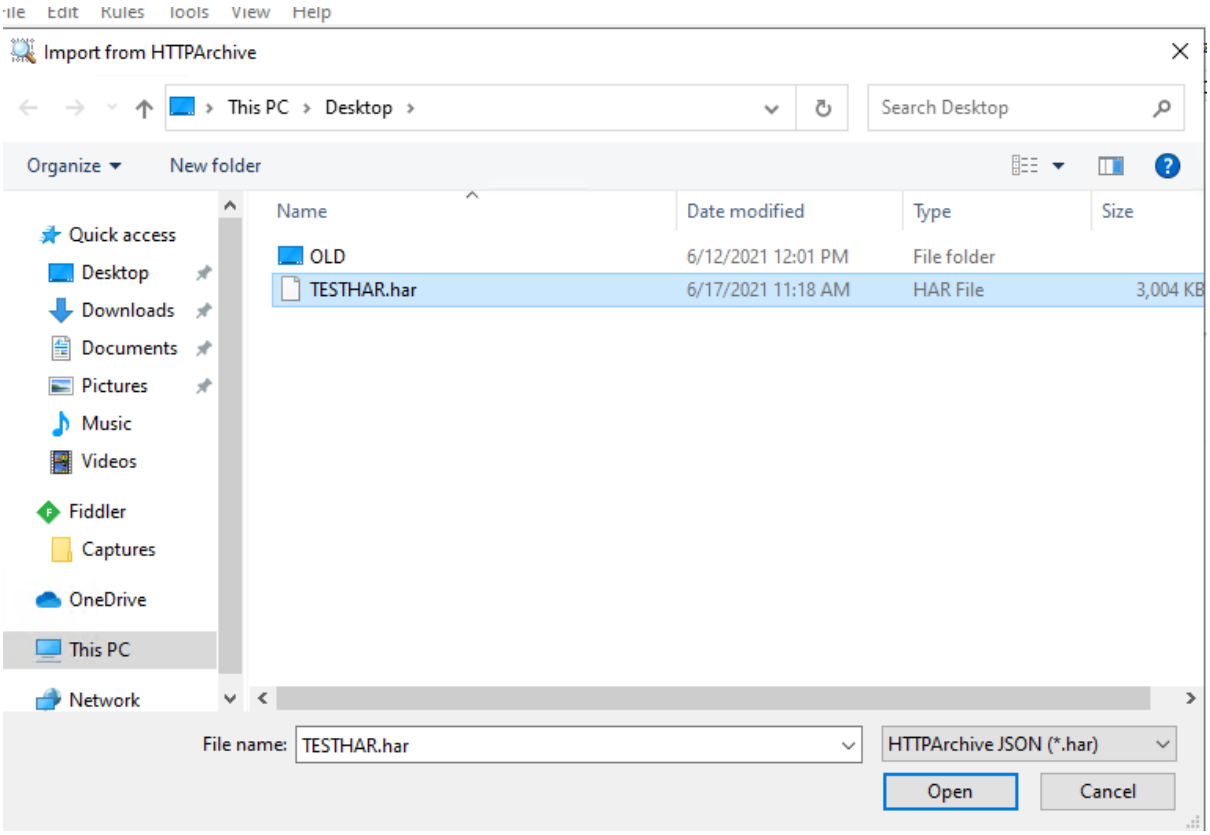
- Choose “File” and “Import Sessions”



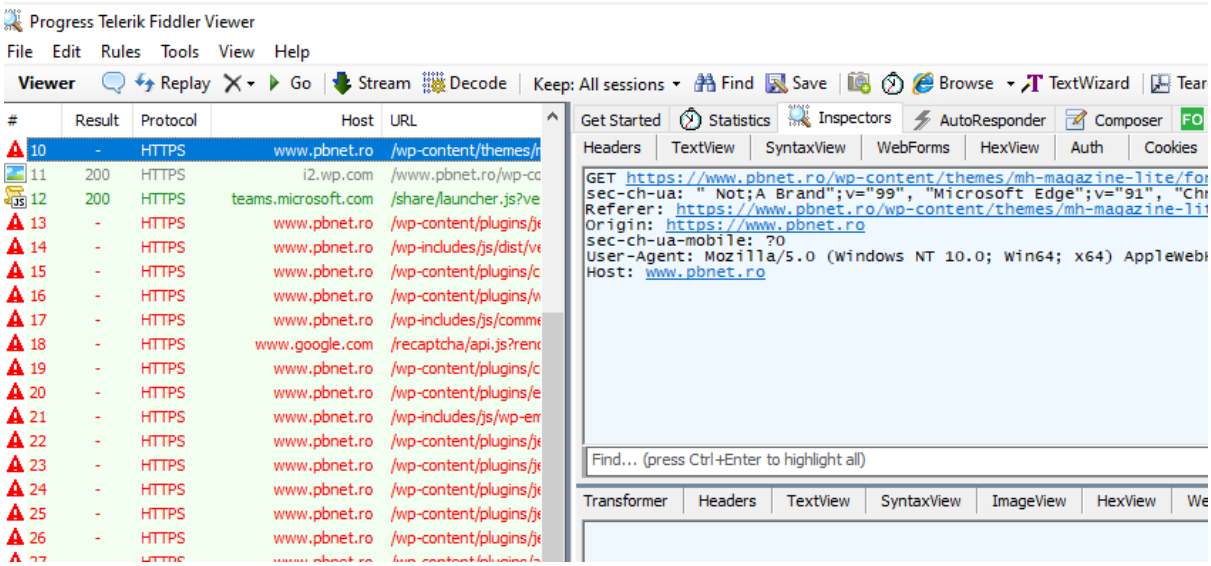
- From the format selector, chose: HTTP Archive



and select the HAR file.



Then Fiddler will open the trace:



4.8 Analyzing NETLOG export files with Fiddler

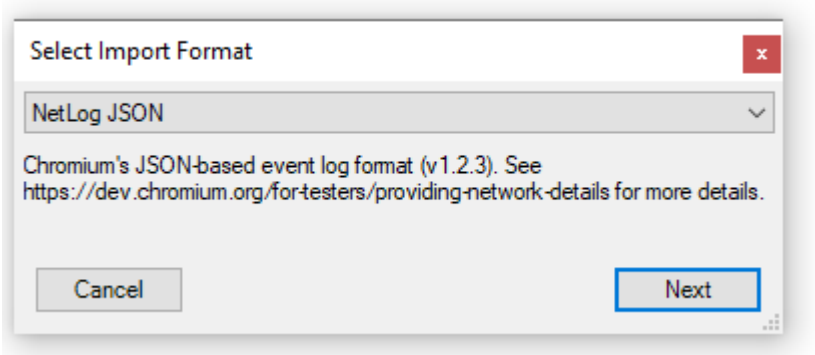
More info on capturing the Netlog dump can be found here: https://www.chromium.org/for-testers/providing-network-details

Note: in Microsoft Edge, you can use edge://net-export

Now you find yourself with a JSON file that you need to analyze.

So, here is a way to do it:

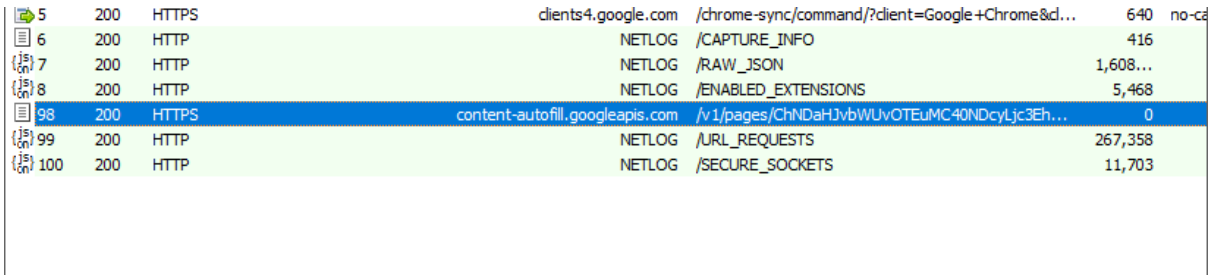
- Download and install the NetLog importer plugin for Fiddler from [**here**]
- Open Fiddler in viewer mode using: *Fiddler -viewer*
- From the File menu, choose **Import sessions** then select the import format "Netlog JSON"



- select the JSON file and click "Open"

Now you will be able to analyze the trace directly in Fiddler.

In addition to the requests and responses parsed from the log, there are several pseudo-Sessions with a fake host of NETLOG that represent metadata extracted from the log:



5	200	HTTPS	clients4.google.com	/chrome-sync/command/?client=Google+Chrome&cl...	640	no-ca
6	200	HTTP	NETLOG	/CAPTURE_INFO	416	
7	200	HTTP	NETLOG	/RAW_JSON	1,608...	
8	200	HTTP	NETLOG	/ENABLED_EXTENSIONS	5,468	
98	200	HTTPS	content-autofill.googleapis.com	/v1/pages/ChNDaHJvbWUvOTEuMC40NDcyLjc3Eh...	0	
99	200	HTTP	NETLOG	/URL_REQUESTS	267,358	
100	200	HTTP	NETLOG	/SECURE_SOCKETS	11,703	

These pseudo-sessions include:

- RAW_JSON contains the raw constants and event data. You probably will never want to examine this view.
- CAPTURE_INFO contains basic data about the date/time of the capture, what browser and OS version were used, and the command line arguments to the browser.
- ENABLED_EXTENSIONS contains the list of extensions that are enabled in this browser instance. This entry will be missing if the log was captured using the –log-net-log command line argument.
- URL_REQUESTS contains a dictionary mapping every event related to URL_REQUEST back to the URL Requests to which it belongs. This provides a different view of the events that were used in the parsing of the Web Sessions added to the traffic list.
- SECURE_SOCKETS contains a list of all the HTTPS sockets that were established for network requests, including the certificates sent by the server and the parameters requested of any client certificates.

Note: the certificates can be viewed by saving the string that begins with "-BEGIN CERTIFICATE" to a file with the .cer extension.

Please keep in mind that the NetLog format does not currently store the request body bytes.