

Build a Test Using a Recorder

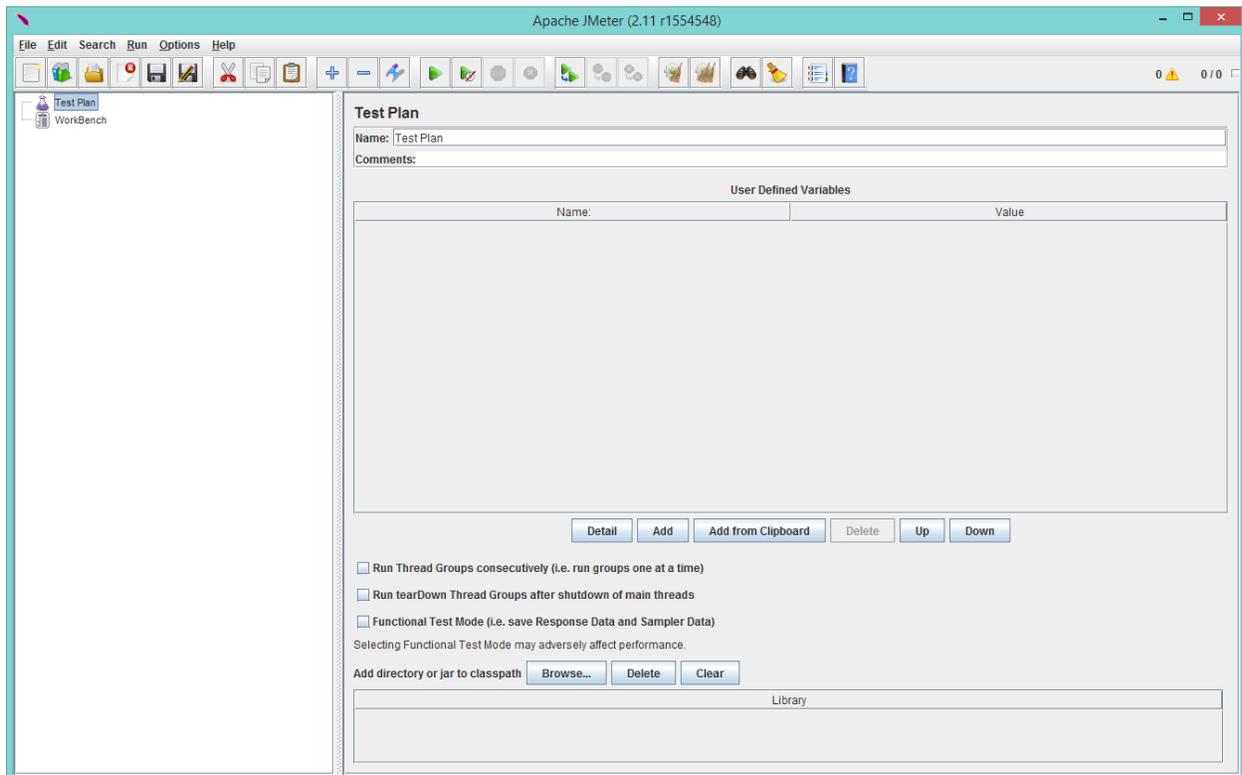
A JMeter Recorder lets you set up a simple test plan by using a browser to move through a website. The Recorder records the sequence of actions that are performed in a browser. This can make it easy to start building a set of tests since the initial moves of the test are automatically recorded as the user steps through a sequence of actions in a browser.

The JMeter Recorder creates within JMeter a proxy that will catch all of the actions that are performed in a Browser. In other words the actions that you perform in a browser will be sent by the browser to the JMeter proxy. JMeter records the action and then send the request on to the website which responds to your request.

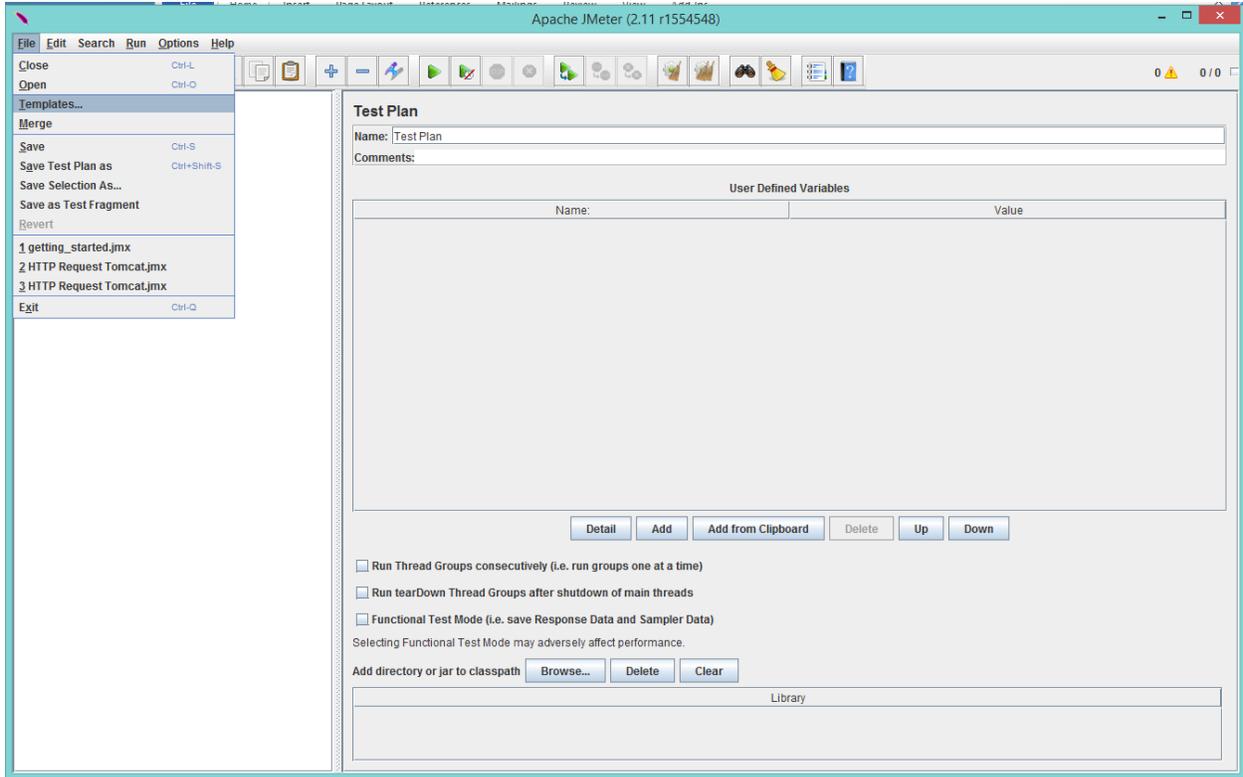
To do this JMeter must be set up to act as a proxy and the web browser you are using must be set up to use a proxy which will be JMeter on localhost as the host.

Set up JMeter with a Recorder (the proxy)

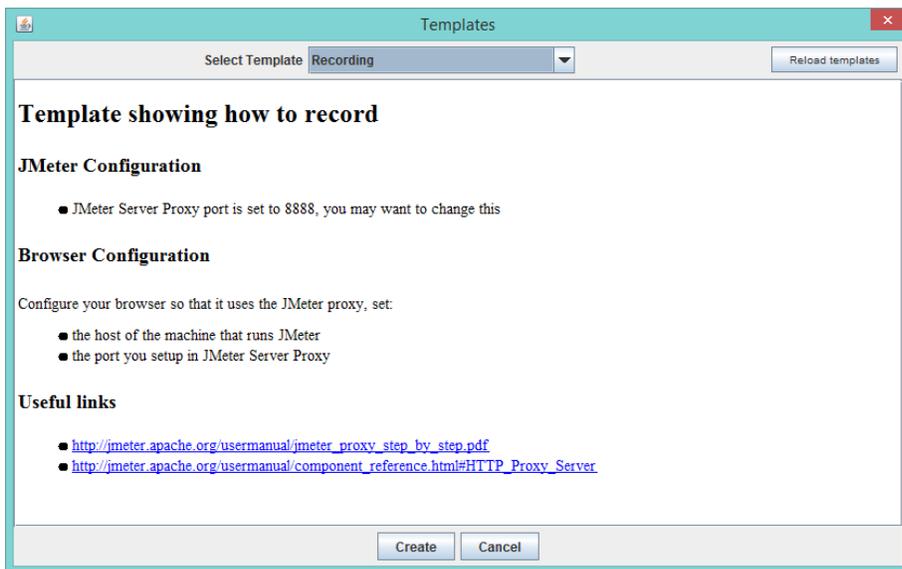
First a test plan will be create in JMeter that will intercept the browser's requests by acting as a proxy. Therefore, start up JMeter. You should see the window below



Next go to File/Templates as shown in the snapshot image on the next page.

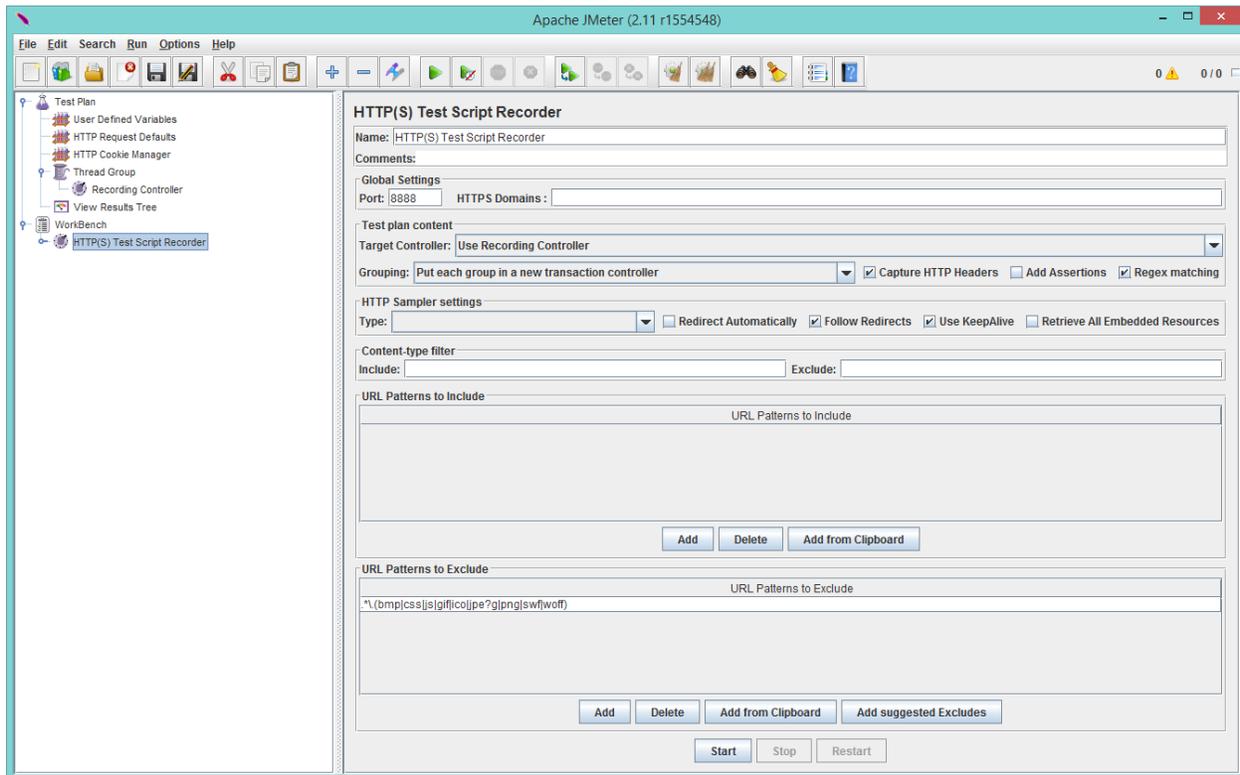


This gives the following Dialog where you can select the template at the top. The Recording template explains what the template will do and gives some useful links.



Now click on the “Create” button to create a new Recording test plan. Note that it is usual to change the port number that the proxy will listen on to a port not in use by any other program running on the computer. This will be done as we set up the test plan.

The test plan, once it is created, will look like the following image. Note that all of the parts of the plan have been expanded and the Test Script Recorder has been selected before the snapshot was taken.



The Test Script Recorder is where the JMeter proxy that will be started is defined.

Note that the template added the following to the test plan

- User Defined Variables – we will not need these
- HTTP Request Defaults – Default values for items such as server name or IP and Port can be defined for the test. If these are not specified in other elements then the defaults apply. Since we are just going to record what is done in the browser we do not need these.
- HTTP Cookie Manager – we will not need this for the work done in recording
- Thread Group – used to define the one thread that will have the recorder in it, the recorder records the actions done in the browser.
- Recording Controller – where the recorded samples will be saved
- View Results Tree – not used here since a test is not being run.
- HTTP(S) Test Script Recorder – Where the proxy that will record the browser actions is defined.

Some of the items placed into the test plan are put there so that the recorded session could be made into a test plan after recording is done.

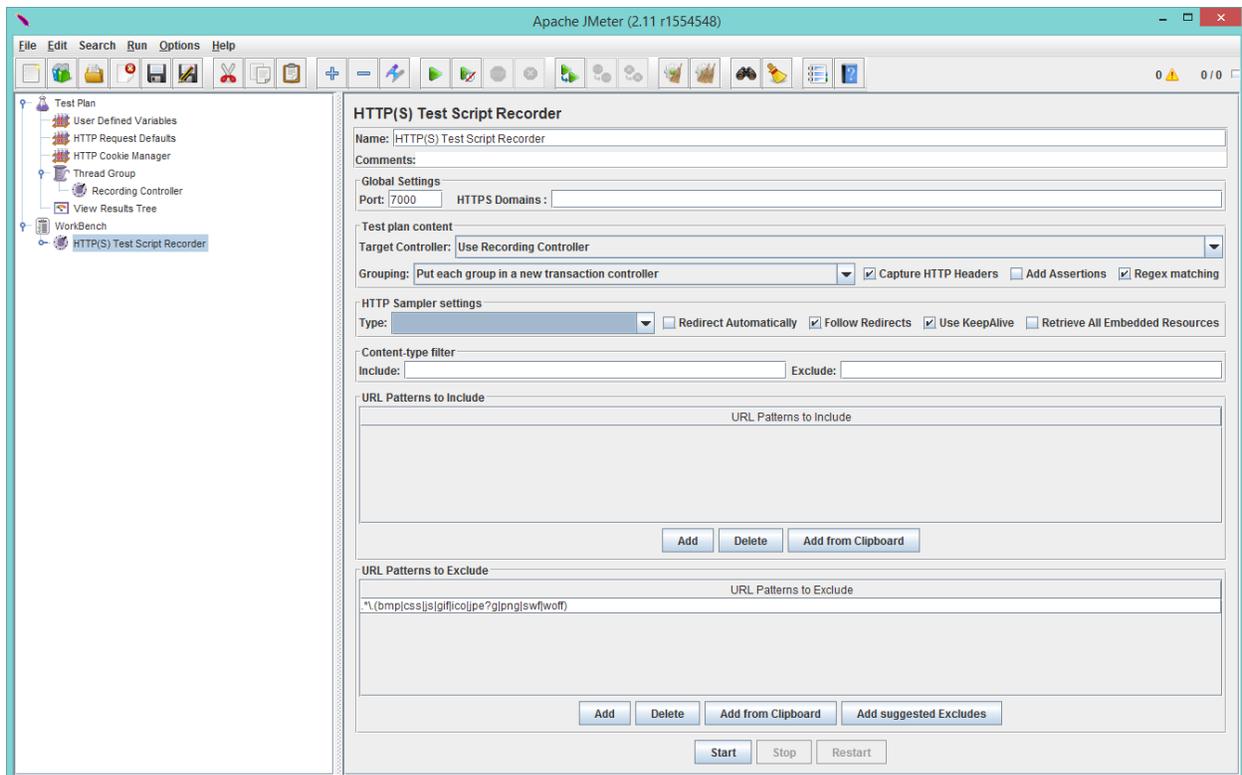
Now double click on the Workbench to open it and click on the HTTP(S) Test Script Recorder. You will get the prior image that shows the parameters/arguments for the Test Script Recorder. This is where the recording proxy in JMeter can be define.

If there is a desire to know the exact definitions for each of the parameters in the Test Script Recorder, they are given at the end of this document.

However, in this case you want to change the Global Setting Port to a port that is not being used on the computer by another program. Generally a port in the 7000s or 8000s is a safe bet for being unused. The command on windows, that can be run in a command prompt, to show the ports in use is `netstat -a`

The rest of the setting defaults are sensible and provide a good starting point.

JMeter Test Script Recorder now looks like:



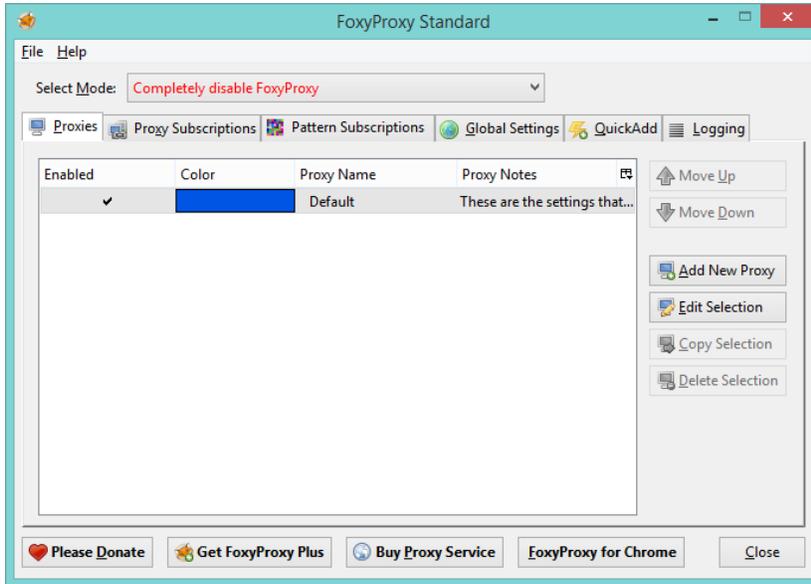
The Test Script Recorder will now start on port 7000 and monitor all requests coming from browsers that are set up with a proxy that is localhost port 7000.

Set up the Browser to use a Proxy

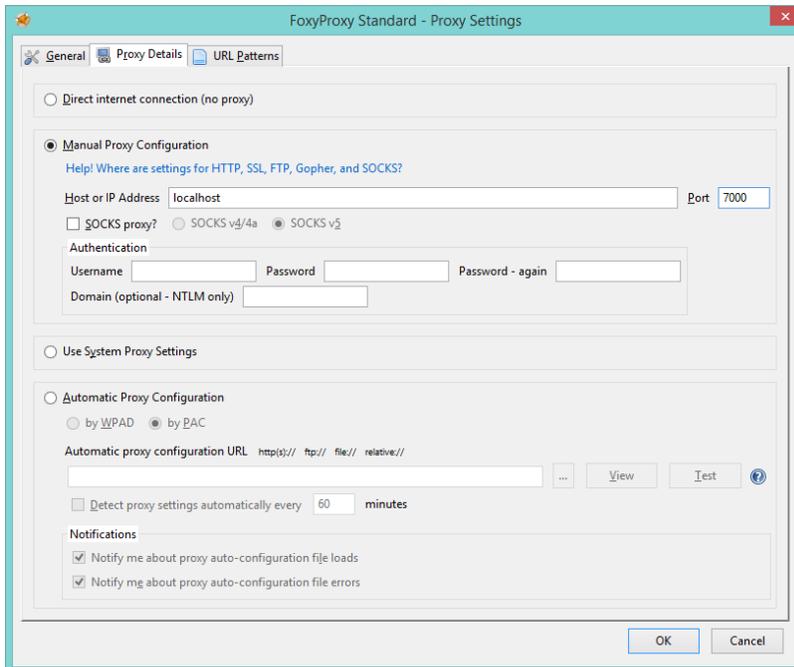
The Browser can be set up to use a proxy with either an add-on or by modifying its settings. Note that after setting up a proxy it is important to remember to remove the proxy after the recording is finished.

If you are using windows the easiest way to set the proxy is to use the Foxyproxy add-on for either IE, Firefox, or Chrome. Go to Foxyproxy.com and download the standard add-on (rather than the basic add-on). The following uses Foxyproxy on Firefox as an example.

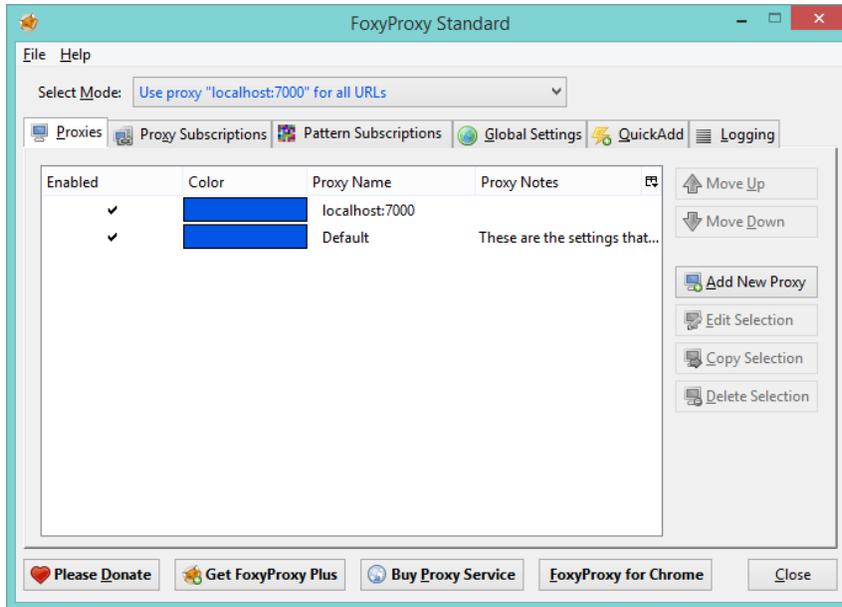
To define a proxy for the Browser using Foxyproxy go to Tools/Foxyproxy Standard. This gives in Firefox the Dialog:



Click on “Add New Proxy”. This gives the dialog:



Define the host as localhost and the port as the same number as used in JMeter for the Test Script Recorder (in the example here it was 7000). Clicking OK defines the proxy and allows you to choose it as shown below:



Click on close. The Recorder test plan in JMeter can now be started. In this case the Firefox browser will be used to move to a website and click on various links. As the actions are performed the Test Script Recorder records the actions made by the user. This forms the start of a test plan.

Parameters For Test Script Recorder

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Port	The port that the HTTP(S) Test Script Recorder listens to. 8080 is the default, but you can change it.	Yes
HTTPS Domains	List of domain (or host) names for HTTPS. Use this to pre-generate certificates for all servers you wish to record. For example, *.apache.org, *.incubator.apache.org Note that wildcard domains only apply to one level, i.e. podling.incubator.apache.org matches *.incubator.apache.org but not *.apache.org	No
Target	The controller where the proxy will store the generated samples. By	Yes

Controller	default, it will look for a Recording Controller and store them there wherever it is.	
Grouping	<p>Whether to group samplers for requests from a single "click" (requests received without significant time separation), and how to represent that grouping in the recording:</p> <ul style="list-style-type: none"> • Do not group samplers: store all recorded samplers sequentially, without any grouping. • Add separators between groups: add a controller named "---- -----" to create a visual separation between the groups. Otherwise the samplers are all stored sequentially. • Put each group in a new controller: create a new Simple Controller for each group, and store all samplers for that group in it. • Store 1st sampler of each group only: only the first request in each group will be recorded. The "Follow Redirects" and "Retrieve All Embedded Resources..." flags will be turned on in those samplers. • Put each group in a new transaction controller: create a new Transaction Controller for each group, and store all samplers for that group in it. <p>The property proxy.pause determines the minimum gap that JMeter needs between requests to treat them as separate "clicks". The default is 1000 (milliseconds) i.e. 1 second. If you are using grouping, please ensure that you leave the required gap between clicks.</p>	Yes
Capture HTTP Headers	Should headers be added to the plan? If specified, a Header Manager will be added to each HTTP Sampler. The Proxy server always removes Cookie and Authorization headers from the generated Header Managers. By default it also removes If-Modified-Since and If-None-Match headers. These are used to determine if the browser cache items are up to date; when recording one normally wants to download all the content. To change which additional headers are removed, define the JMeter property proxy.headers.remove as a comma-separated list of headers.	Yes
Add Assertions	Add a blank assertion to each sampler?	Yes
Regex Matching	Use Regex Matching when replacing variables? If checked replacement will use word boundaries, ie it will only replace word matching values of variable, not part of a word. A word boundary follows Perl5 definition and is equivalent to \b. More information below in the paragraph about "User Defined Variable replacement".	Yes
Type	Which type of sampler to generate (the Java default or HTTPClient)	Yes
Redirect	Set Redirect Automatically in the generated samplers?	Yes

Automatically		
Follow Redirects	Set Follow Redirects in the generated samplers? Note: see "Recording and redirects" section below for important information.	Yes
Use Keep-Alive	Set Use Keep-Alive in the generated samplers?	Yes
Retrieve all Embedded Resources	Set Retrieve all Embedded Resources in the generated samplers?	Yes
Content Type filter	Filter the requests based on the content-type - e.g. "text/html [;charset=utf-8]". The fields are regular expressions which are checked to see if they are contained in the content-type. [Does not have to match the entire field]. The include filter is checked first, then the exclude filter. Samples which are filtered out will not be stored. Note: this filtering is applied to the content type of the response	No
Patterns to Include	Regular expressions that are matched against the full URL that is sampled. Allows filtering of requests that are recorded. All requests pass through, but only those that meet the requirements of the Include/Exclude fields are <i>recorded</i> . If both Include and Exclude are left empty, then everything is recorded (which can result in dozens of samples recorded for each page, as images, stylesheets, etc are recorded). If there is at least one entry in the Include field, then only requests that match one or more Include patterns are recorded .	No
Patterns to Exclude	Regular expressions that are matched against the URL that is sampled. Any requests that match one or more Exclude pattern are <i>not</i> recorded .	No
Notify Child Listeners of filtered samplers	Notify Child Listeners of filtered samplers Any response that match one or more Exclude pattern is <i>not</i> delivered to Child Listeners (View Results Tree) .	No
Start Button	Start the proxy server. JMeter writes the following message to the console once the proxy server has started up and is ready to take requests: "Proxy up and running!".	N/A
Stop Button	Stop the proxy server.	N/A
Restart Button	Stops and restarts the proxy server. This is useful when you change/add/delete an include/exclude filter expression.	N/A