# **Processor in JMeter: PreProcessor & PostProcessor**

A processor is used to modify the Samplers in their scope.

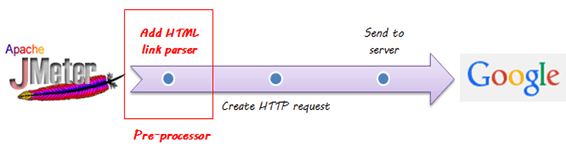
There are 2 Types of processors:

1. Pre-processor
2. Post-processor

## **Pre-processor**

Pre-processor executes some action **before** making Sampler Request.

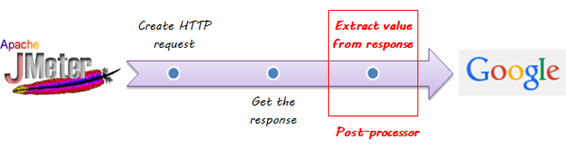
Consider a simple example: let’s say you wanted JMeter to “spider” through the website under test, **parse**link(check all links on the page) and **return** the HTML. You would add some action such as “HTML link parser” to your controller before creating an HTTP request.



## **Post-processor**

Post-processor executes some action after making a Sampler Request.

Consider a simple example: JMeter sends an HTTP request to the web server under test (etc [www.google.com](https://www.google.com/)) and get the response. You want JMeter to stop the testif the server response is an error. You can use the post-processor to do above task as follows:

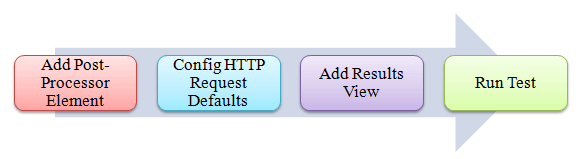


## **Post Processor Example**

This tutorial will show you step-by-step instructions on how to use Post-processor in JMeter. Let start with the simple test script.

1. JMeter sends an HTTP request to the web server under test [www.google.com](https://www.google.com/).
2. JMeter gets a response from the Google server.
3. If server response is **an error**, JMeter will **stop** the test.
4. If server response **OK** (no error), JMeter will **continue** the test.

Here is the **roadmap** of this example:



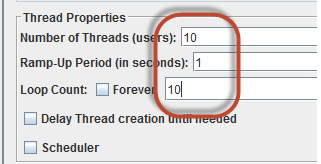
**Pre-condition:**

We **re-use** the Step 1 and Step 2 in article [JMeter Performance Testing](https://www.guru99.com/jmeter-performance-testing.html).

## **Step 1) Add Thread Group**

Right click on the[Test Plan](https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html)and add a new thread group: **Add**-> **Threads (Users)** -> **Thread Group**

But in Thread Group control panel, enter Thread Properties as follows:



This setting lets JMeter create **10** user request to [http://www.google.com](https://www.google.com/) **10** times.

## **Step 2) Add JMeter elements**

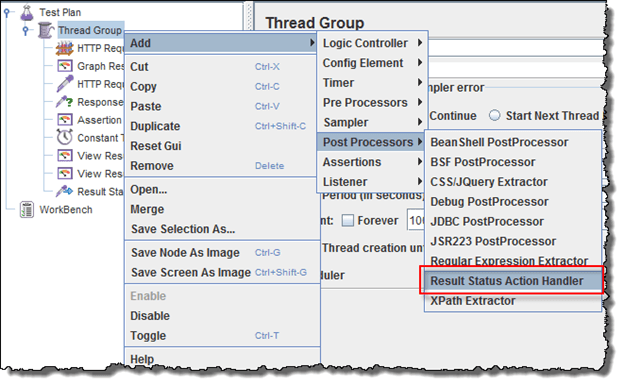
* Add HTTP request default
* Add HTTP request

We still make JMeter send request [http://www.google.com](https://www.google.com/) to Google server.

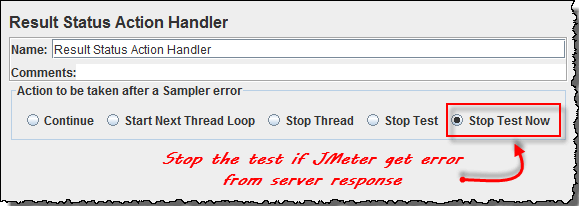
## **Step 3) Add Post-Processor Element**

Right Click **Thread Group**->**Add**->**Post Processor**->**Result Status Action Handler**

**Result Status Action Handler** allows the user to stop the thread or the whole test if the user request failed.

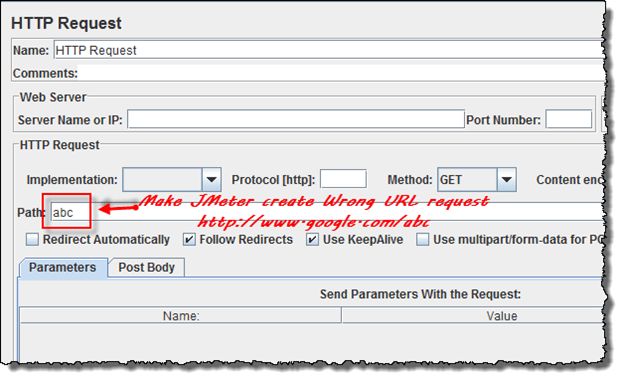


In Result Status Action Handle Pane, choose **Stop Test Now**. This selection will stop the test if JMeter get the error from server response.



## **Step 4) Config the HTTP Request**

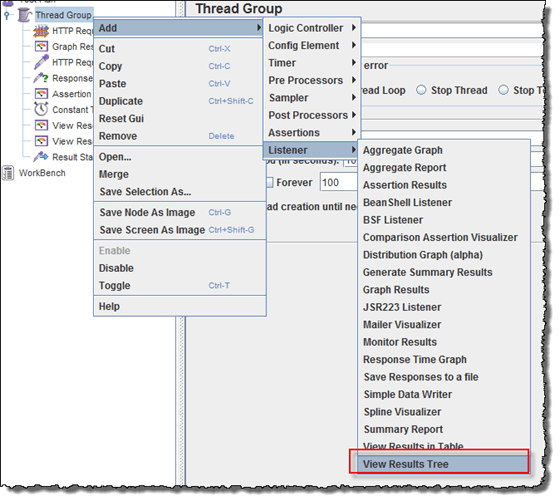
Open the HTTP Request Panel. Enter **“abc”** to the Path field.



When you enter **“abc”** to the path, JMeter will create a URL request to Google server: http://www.google.com/abc. This URL doesn’t exist on Google server. It is **wrong** URL request so Google server will return an error.

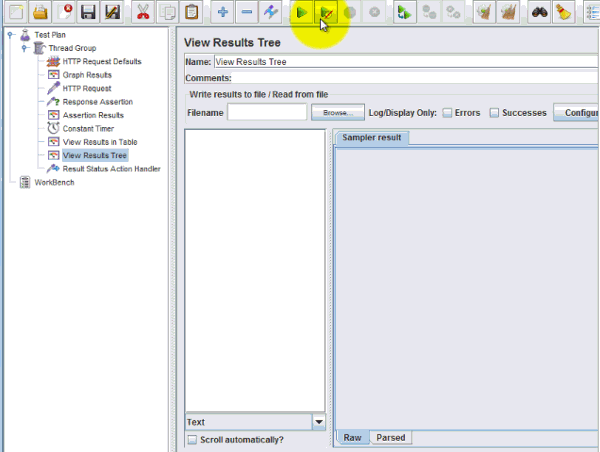
## **Step 5) Add View Result Tree**

Right Click **Thread Group**->**Add**-> **Listener**-> **View Result Tree**

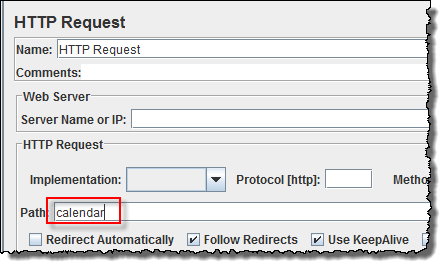


## **Step 6) Run Test**

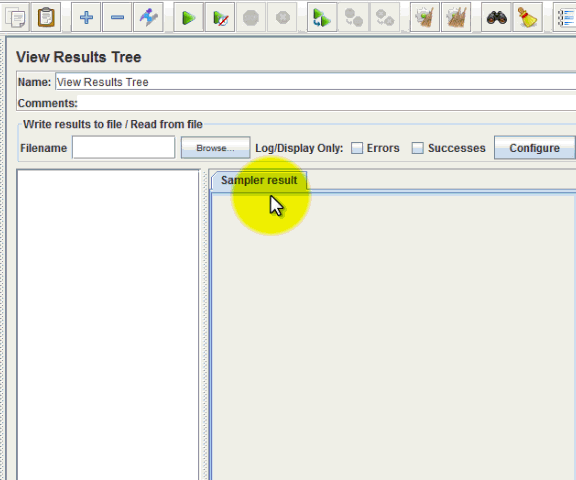
Select View Result Tree, press Run button on Menu bar. You will see the **error** response from Google server and the test will stop **with out** completing 100 threads.



Now return to step 4, open the HTTP Request pane, enter “**calendar**” to the pane. It makes JMeter create URL request <https://calendar.google.com/calendar/u/0/r> to the Google server. This is **correct** URL request so Google server will return OK (no error).



Select View Result Tree, press Run button on Menu bar. You will see the **OK** response from Google server and the test will continue until all 100 threads are complete.



## **Troubleshooting**

If you face the issue while running the above scenario … do the following:

1. Check whether you are connecting to the internet via a proxy. If yes, remove the proxy.
2. Open a new instance of Jmeter
3. Open the [ProcessorTestPlan.jmx](https://drive.google.com/uc?export=download&id=0B_vqvT0ovzHccFZlUG96ejE3YWM) in Jmeter
4. Double-click on Thread Group -> View Results Tree
5. Run the Test