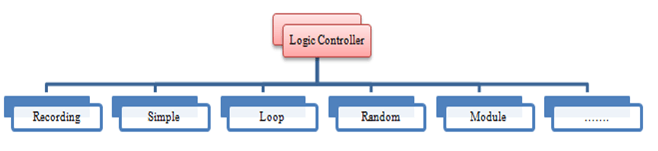
**What is the Logic Controller?**

Logic Controllers let you define the order of processing request in a Thread. It lets you control "when" to send a user request to a web server. For example, you can use Random Controllers to send HTTP requests to the server randomly

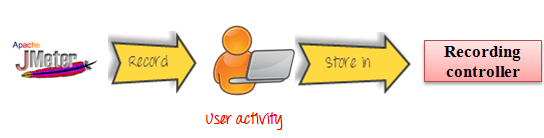
Logic Controllers determine **the order** in which user request is executed.

Some commonly used Logic controllers are below:

[](https://www.guru99.com/images/LogicControllerClassification.png)

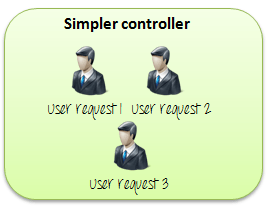
**Recording Controller:**

JMeter can **record** your[Testing](https://www.guru99.com/software-testing.html)steps; a recording controller is a **placeholder** to store these recording steps.

[](https://www.guru99.com/images/RecordingControllerJMeter.png)

**Simple Controller:**

Simple Controller is just a **container** for user request.

[](https://www.guru99.com/images/SimpleController.png)

**Loop Controller:**

Loop Controller makes the user request run **a specified number of times** or run **forever** as shown in figure:

[](https://www.guru99.com/images/LoopControllerJMeter.png)

**Random Controller:**

Random Controller makes all the user requests run in **the random** order in each loop period.

For example, you have 3 user requests to website [http://www.google.com](http://www.google.com/) in following order:

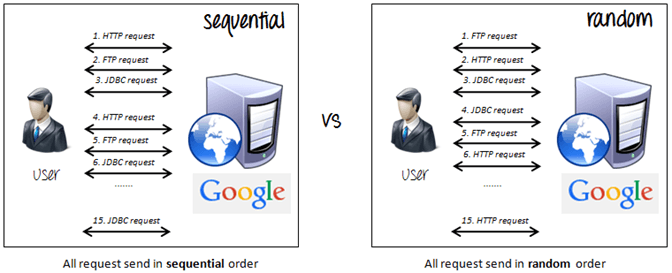
1. HTTP request
2. FTP request
3. JDBC request

These 3 requests should run 5 times. Total 15 user requests will be sent to Google server by JMeter.

In **sequential** order, requests are sent **sequentially** in following order:

HTTP request ->FTP request->JDBC request

for each loop.

[](https://www.guru99.com/images/SequentialVSRandomController.png)

In **random** order, requests are sent as **randomly,**

FTP request ->HTTP request->JDBC request

Or

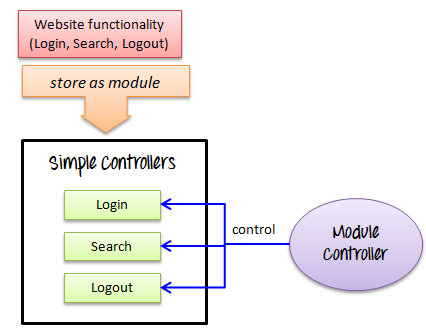
JDBC request ->FTP request->HTTP request

For each loop.

**Module Controller:**

The goal of Module Controller is to add modularity to JMeter.

The general idea is that web applications consist of small units of functionality (i.e. Logon, Create Account, Logoff...). This functionality can be stored in [Simple Controller](https://www.guru99.com/controllers-in-jmeter.html#SimpleController) as "modules".  Module Controller will choose which module needs to run.

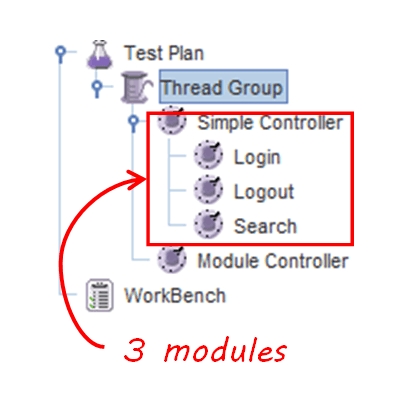
[](https://www.guru99.com/images/ModuleControllerJMeter.png)

Consider the following scenario -

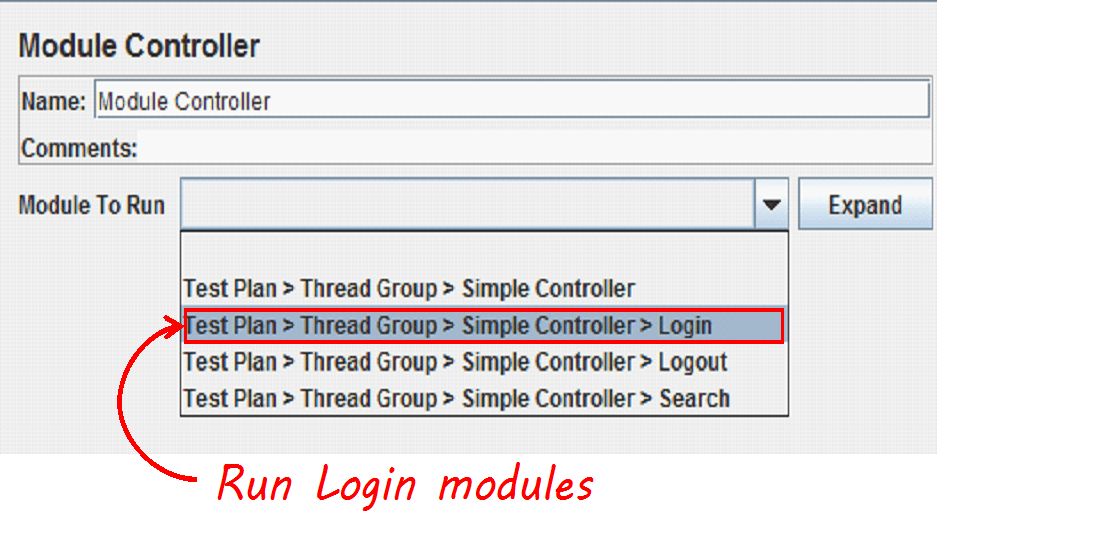
You want to simulate:

* 50 users **logging out**,
* 100 users **logging in**
* 30 users  **search**[www.google.com](http://www.google.com/)

You can use JMeter to create 3 modules. Each module simulates each user activity: Login, Logout, and Search.

[](https://www.guru99.com/images/TestNewPlanJMeterController.png)

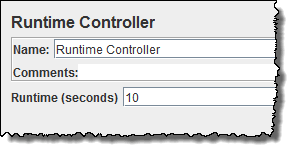
The Module controller chooses which module needs to run.

[](https://www.guru99.com/images/RunLoginModuleJMeter.png)

**Other Important Controllers:**

* **Interleave** Controller:  picks up and makes **one** of user request run in **each** loop of the thread.
* **Runtime** Controller: controls **how long** its children are allowed to run.

For example, if you specified Runtime Controller 10 seconds, JMeter will run your test for 10 seconds.

[](https://www.guru99.com/images/RunTimeControllerJMeter.png)

* **Transaction**Controller: measures the **overall time** taken to **finish** a test execution
* **Include**Controller: is designed to use an external test plan. This controller allows you to use multiple test plans in JMeter. See detail in [JMeter Performance Testing](https://www.guru99.com/jmeter-performance-testing.html).

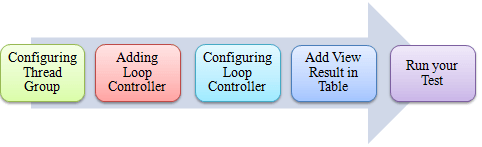
**Loop Controller Example**

This section shows you step-by-step instruction to add **Loop Controller**set to your current performance test plan.

The Loop Controller makes the samplers run as a certain number of times, in addition to the loop value you specified for the Thread Group. For example, if you

* Add one HTTP Request to a Loop Controller with a loop count 50
* Configure the Thread Group loop count to 2
* Then, JMeter will send a total of 50 \* 2 = 100 HTTP Requests.

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

[](https://www.guru99.com/images/RunTestPlanFlowForJMeter(1).png)

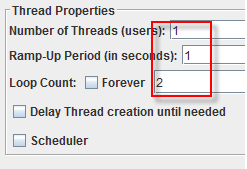
**Step 1) Configuring Thread Group**

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

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:

[](https://www.guru99.com/images/ThreadPropertyForJMeter.png)

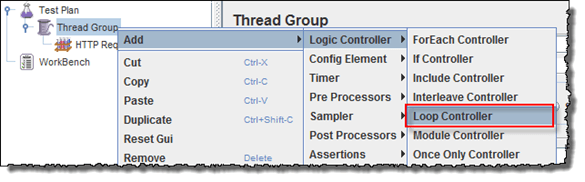
          It will make **one** user request to the web server google.com and run it **2** times.

1. **Add JMeter elements**

          Add HTTP request default to www.google.com.

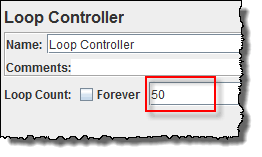
1. **Adding Loop Controller**

          Right Click Thread Group -> Logic Controller -> Loop Controller

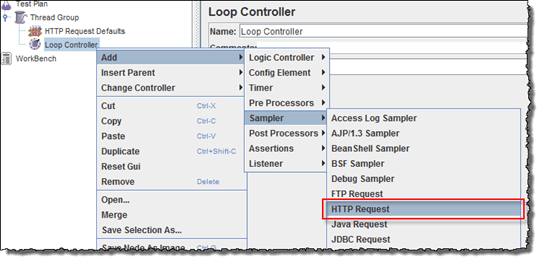
[](https://www.guru99.com/images/AddingLoopControllerJMeter.png)

**Step 2) Configuring Loop Controller**

Add value 50 to Loop Count field as below figure. It will make **one** user request to the web server [google.com](http://www.google.com/) run it **50** times, in addition to the loop value =2, you specified for the Thread Group above.So JMeter will send a total of **2 \* 50 = 100** HTTP Requests.

[](https://www.guru99.com/images/ConfiguringLoopController(1).png)

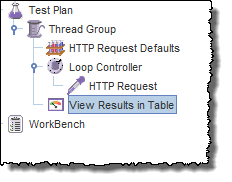
Right click Loop Controller, Add -> Sampler -> HTTP request

[](https://www.guru99.com/images/AddSamplerHTTPRequest.png)

**Step 3) Add View Results in Table**

We re-use Step 2 in [Timer](https://www.guru99.com/timers-jmeter.html) to add View Results in Table

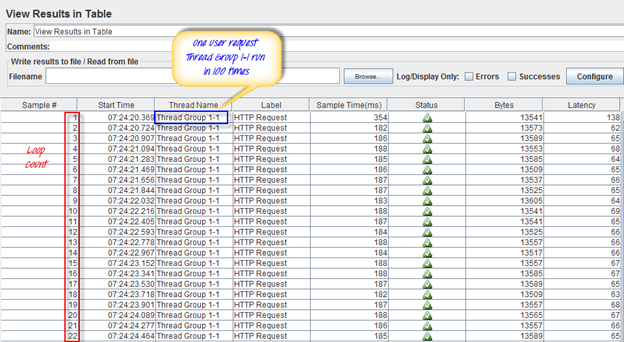
So the test plan is shown in below figure

[](https://www.guru99.com/images/ViewResultInTableJMeterController.png)

**Step 4) Run your test**

Now return View Results in Table, click Start button on the Menu bar (Ctrl+R) to run a test

As shown in the figure below, JMeter simulates **one user request,**which is sent 100 times, to the web server <http://www.google.com/>. The Test is stopped after a user request was sent in 100 times.

[](https://www.guru99.com/images/RunYourTestViewResultInTableController.png)