# Automation Testing

TECHNOLOGY

# Configuration Elements

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
  cout << "Hello, world!" << endl;
  return 0;
}
```

# A Day in the Life of an Automation Test Engineer

Alex learned about the JMeter tool, which is used for a variety of non-functional testing.

Now, he wants to understand the JMeter configuration elements and how it allows users to create defaults and variables to be used by the samplers.

In this lesson, he will learn about how configuration elements are used to add or modify requests made by the samplers.

# Learning Objectives

By the end of this lesson, you will be able to:

- Learn about the configuration elements that are provided in JMeter
- Analyze the input fields of the random variable
- Apply counter in JMeter
- Understand the usage of CSV Data Set Config in JMeter

# Random Variables and Counters

# JMeter Test Plan

A JMeter Test Plan comprises at least one Thread Group.

# JMeter Test Plan

Within each Thread Group, a user may place a combination of one or more elements like Sampler, Logic Controller, Configuration Element, Listener, and Timer.

Each Sampler can be preceded by one or more pre-processor elements followed by a post-processor element and an assertion element.

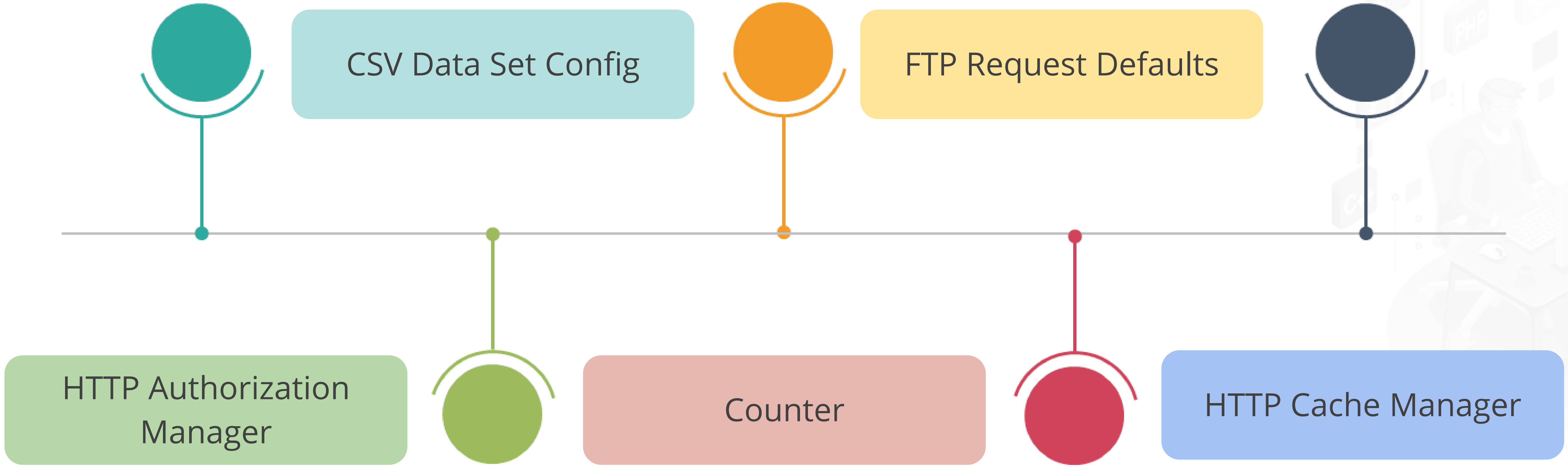
# Configuration Elements

Configuration Elements allow users to create defaults and variables to be used by samplers. These are used to add or modify requests made by Samplers.

They are executed before any samplers located in the same scope as them.

# Configuration Elements

JMeter provides these configuration elements:



CSV Data Set Config

FTP Request Defaults

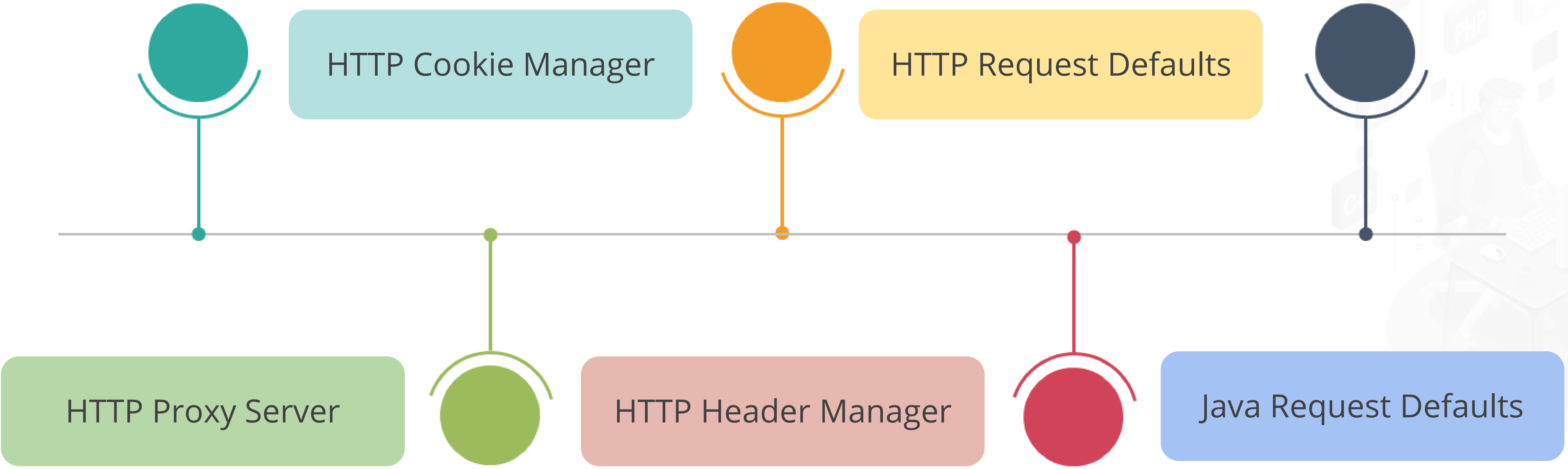
HTTP Authorization Manager

Counter

HTTP Cache Manager

# Configuration Elements

JMeter provides these configuration elements:



HTTP Cookie Manager

HTTP Request Defaults

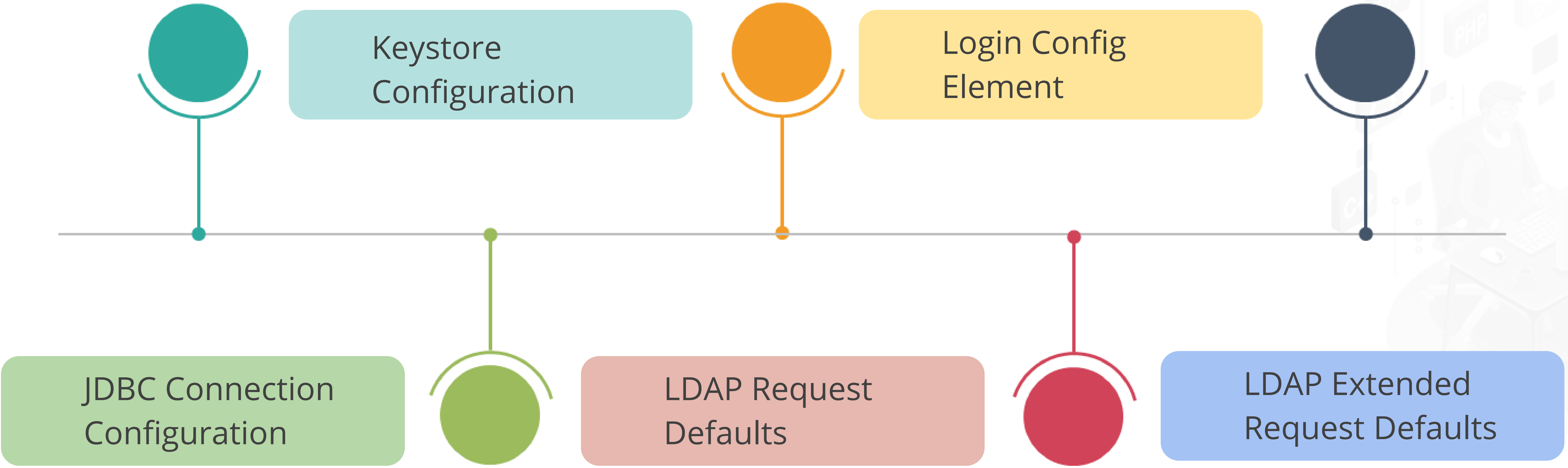
HTTP Proxy Server

HTTP Header Manager

Java Request Defaults

# Configuration Elements

JMeter provides these configuration elements:



Keystore Configuration

JDBC Connection Configuration

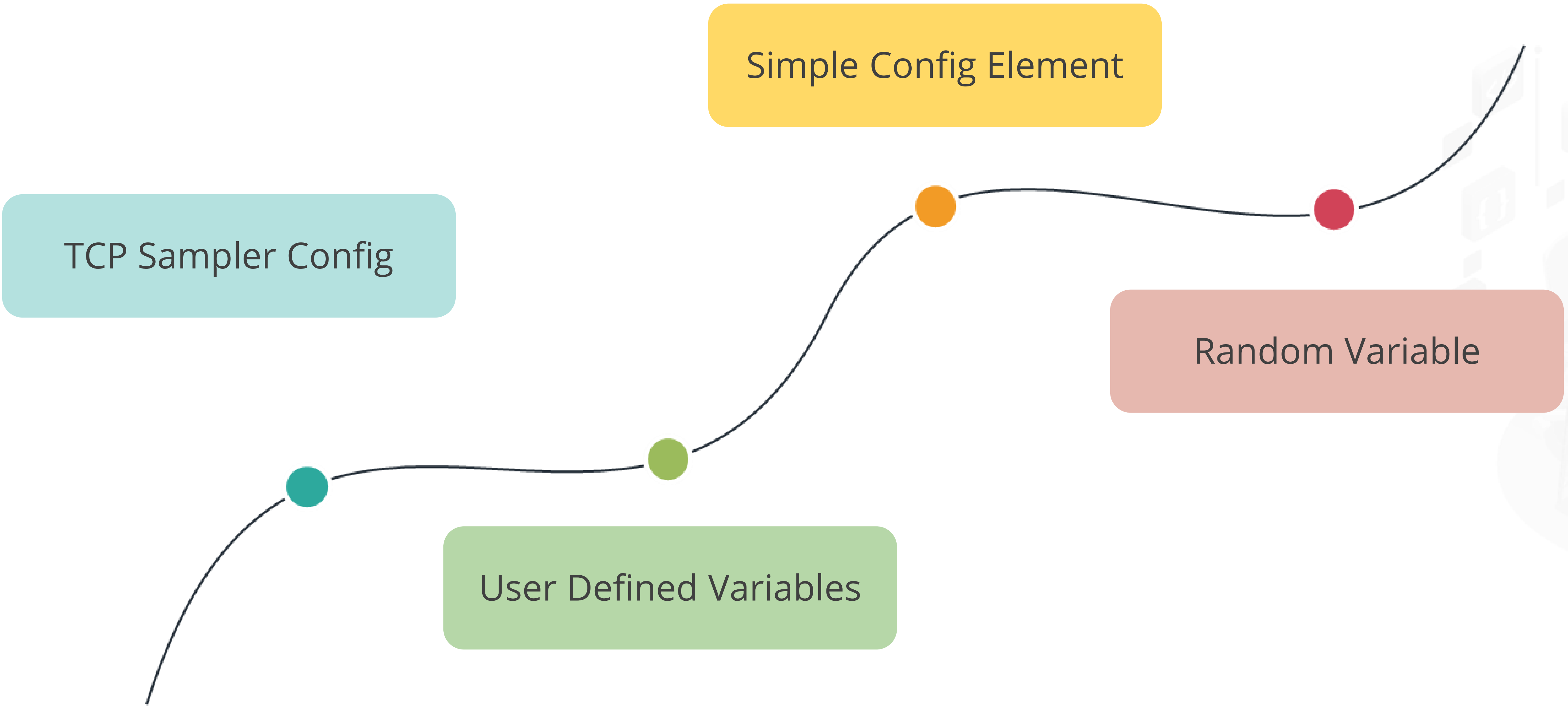
Login Config Element

LDAP Request Defaults

LDAP Extended Request Defaults

# Configuration Elements

JMeter provides these configuration elements:



TCP Sampler Config

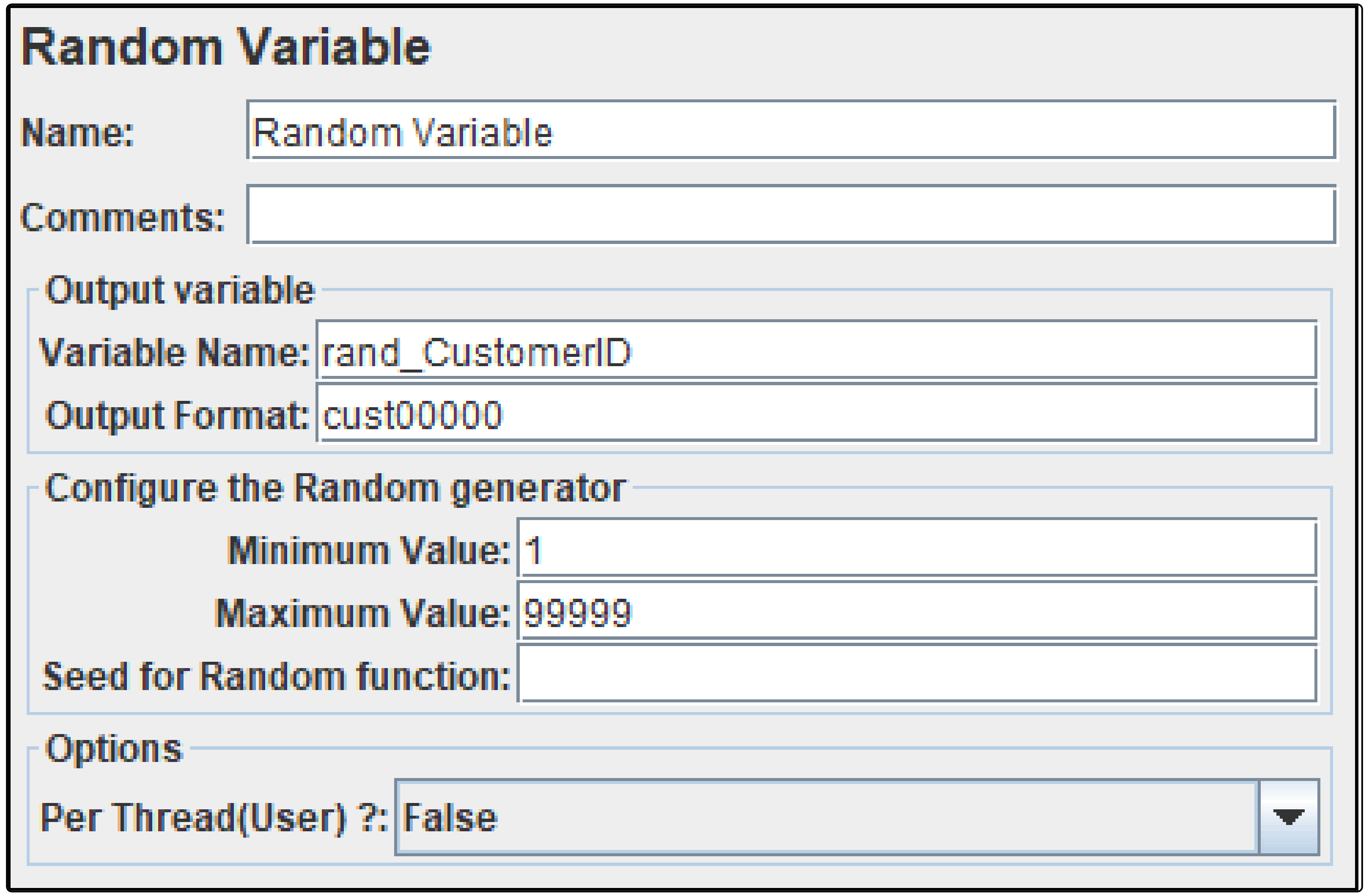
User Defined Variables

Simple Config Element

Random Variable

# Random Variable

A random variable configuration element is used to generate a random integer value between a specified range for each iteration.



Random Variable

Name: Random Variable

Comments:

Output variable

Variable Name: rand_CustomerID

Output Format: cust00000

Configure the Random generator

Minimum Value: 1

Maximum Value: 99999

Seed for Random function:

Options

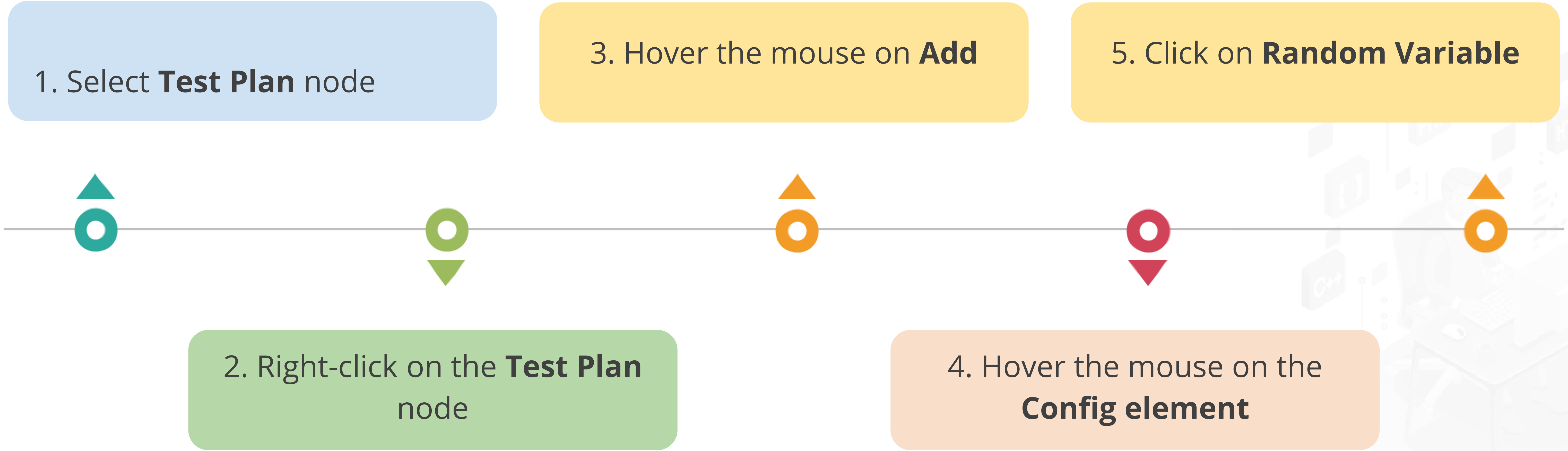
Per Thread(User) ?: False

The generated value can be concatenated with a string which is stored in a variable.

Image source: www.blazemeter.com

# Adding Random Variable in JMeter

Steps to be followed:



1. Select **Test Plan** node

2. Right-click on the **Test Plan** node

3. Hover the mouse on **Add**

4. Hover the mouse on the **Config element**

5. Click on **Random Variable**

# Input Fields of Random Variable

Random Variable has the following input fields:

- **Name**: To provide element name
- **Comments**: To provide arbitrary comments
- **Variable Name:** To store the generated random value
- **Output Format:** To specify the format of the string

# Input Fields of Random Variable

Random Variable has the following input fields:

- **Minimum Value:** The minimum value of the random number range
- **Maximum Value:** The maximum value of the random number range
- **Seed for random function:** The random number generator (Default is the current time in milliseconds)
- **Per Thread (User):** The generated value is true or false

# Counter in JMeter

In JMeter, a counter is an element that enables users to create incremental values.

As a result, the user may use the value from any location inside a thread group.

# Usage of JMeter Counter

JMeter counter is used in cases where users may require to build various types of advanced JMeter test plans.

These advanced test plans not only include replaying a recorded test scenario with an increased number of users but something more complex. In such cases, users need some form of a counter.

# Usage of JMeter Counter

The better way of using a counter is with a Loop Controller.

Example: A user needs to create five entities in a loop using the HTTP Request sampler and each entity name must be unique.

# Usage of JMeter Counter

In this example, the process is as follows:

**Step 1: Add a Loop Controller:** The user will add a Loop Controller and set the Loop Count to 5.

# Usage of JMeter Counter

**Step 2: Define the JMeter Counter:** The user will define a counter inside the Loop Controller and configure it.

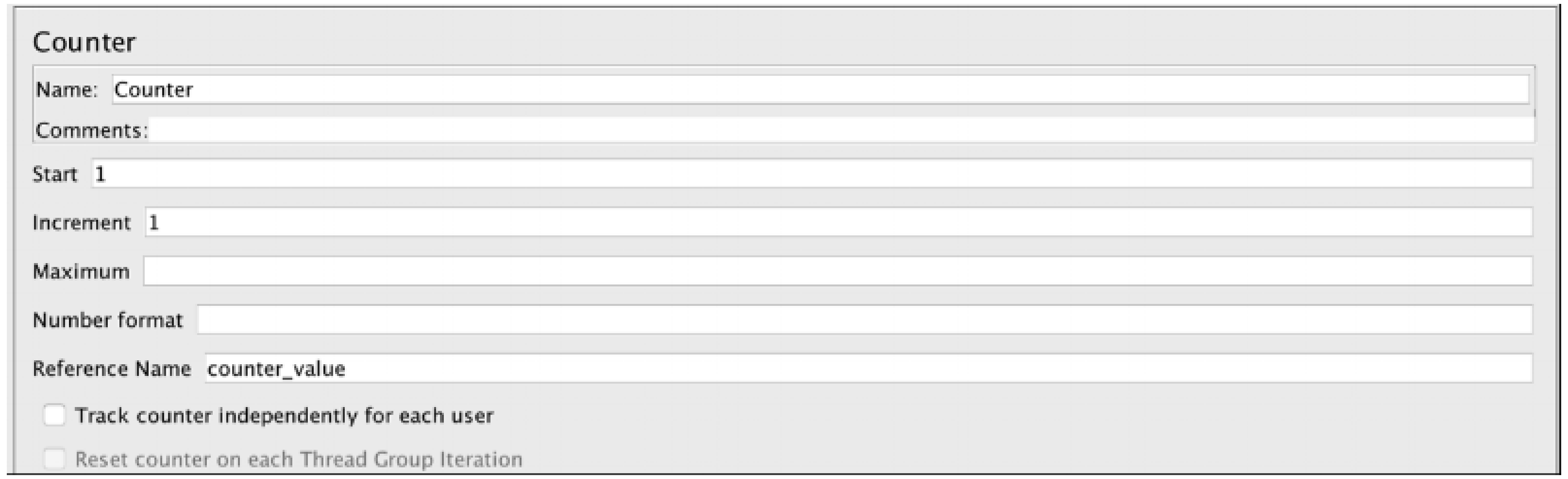
- **Start**: This is the initial counter value; the user should make it 1.
- **Increment**: This value will be added to the current Counter value once the Counter is hit. Here, the user should make it 1.
- **Maximum**: If it is left blank, the Counter value will increase infinitely.

# Usage of JMeter Counter

- **Number Format**: The user can change the output number format using this parameter.
- **Reference Name**: The user can define a JMeter Variable name that will be holding the current Counter value. Here, it is **counter_value**.

# Usage of JMeter Counter

The Counter configuration should look like this:



Counter

Name: Counter

Comments:

Start 1

Increment 1

Maximum

Number format

Reference Name counter_value

□ Track counter independently for each user

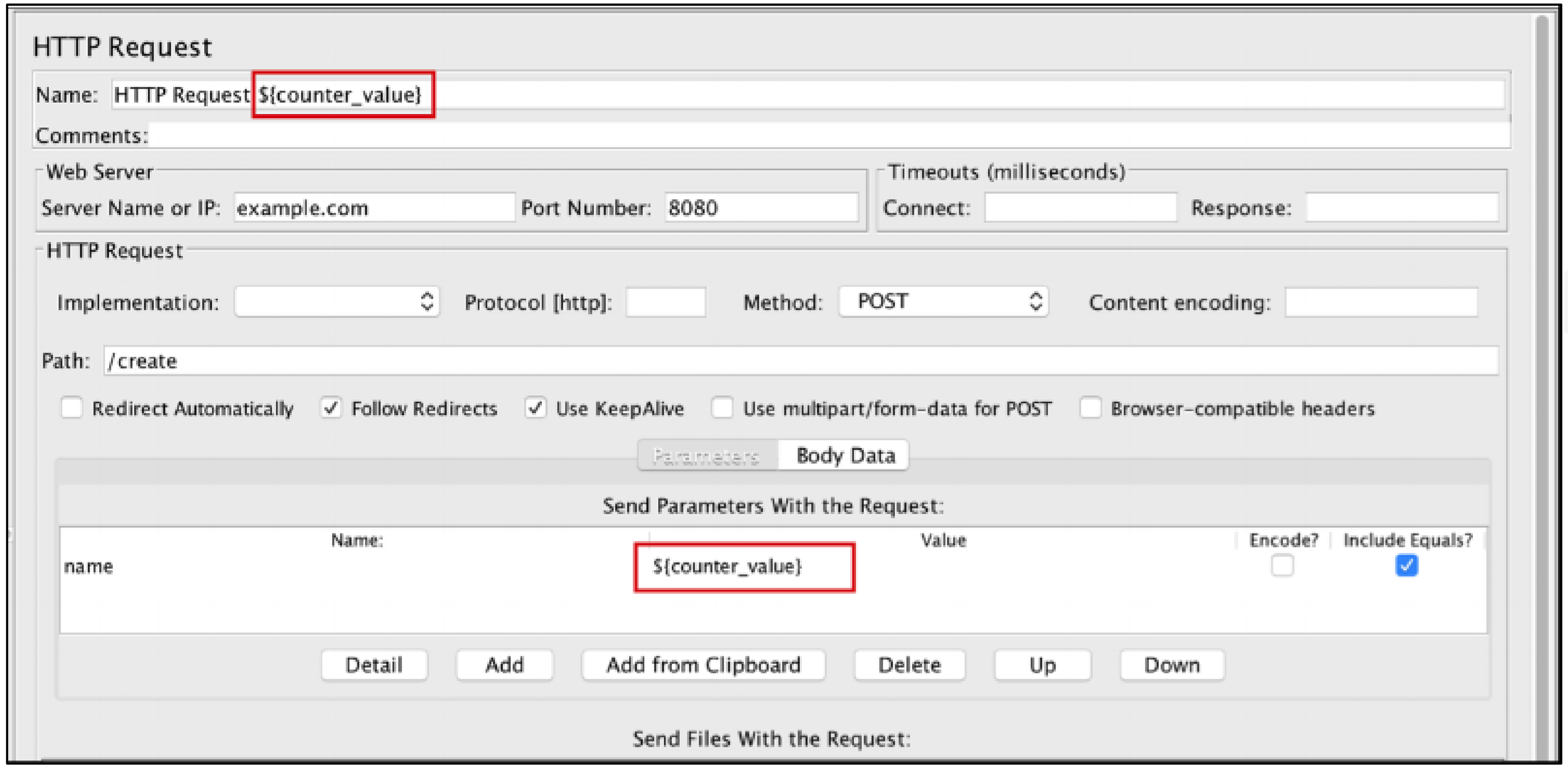
□ Reset counter on each Thread Group Iteration

Image source: www.blazemeter.com

# Usage of JMeter Counter

User should amend the HTTP Request sampler **name** parameter so that it would take the current Counter value and send it to the server.

The HTTP Request should be:



HTTP Request

Name: HTTP Request ${counter_value}

Comments:

Web Server

Server Name or IP: example.com Port Number: 8080

Timeouts (milliseconds)

Connect: Response:

HTTP Request

Implementation: Protocol [http]: Method: POST Content encoding:

Path: /create

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Send Parameters With the Request:

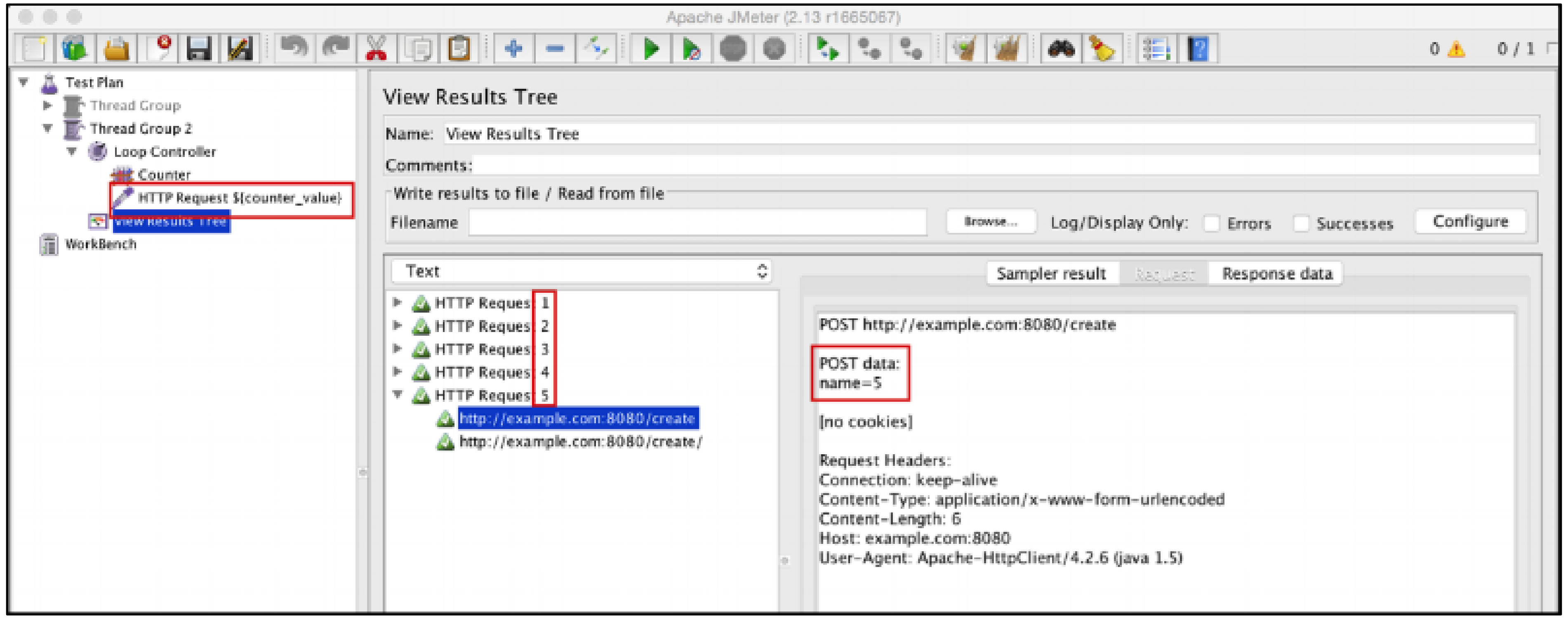
| Name: | Value | Encode? | Include Equals? |
|---|---|---|---|
| name | ${counter_value} | ☐ | ☑ |

Send Files With the Request:

Image source: www.blazemeter.com

# Usage of JMeter Counter

In the **View Results Tree listener** output, the results should be identical as if they were implemented by copy and paste.



Name: View Results Tree

Comments:

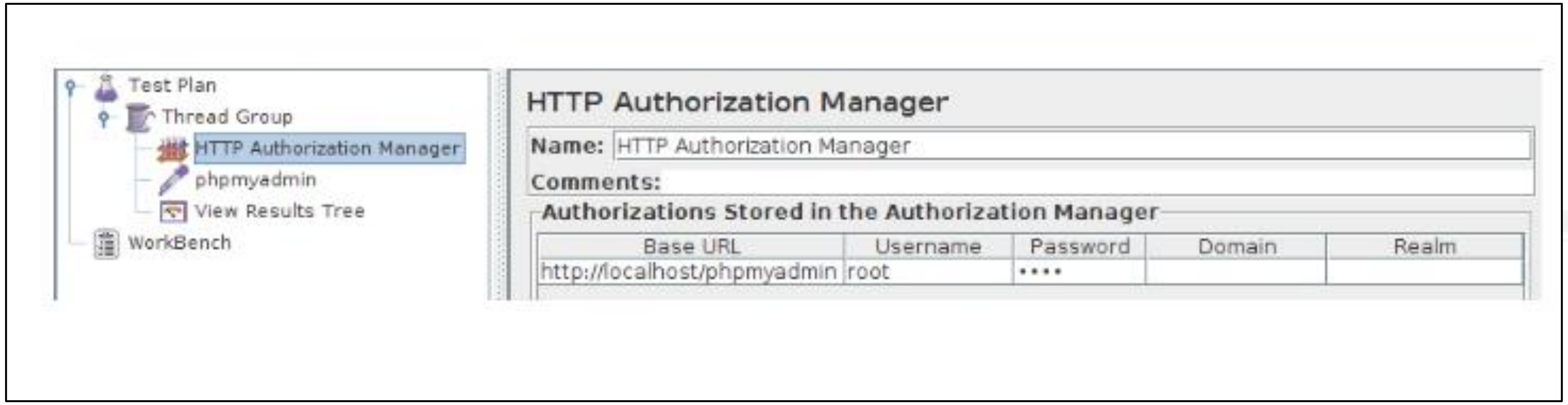
```
POST http://example.com:8080/create

POST data:
name=5

[no cookies]

Request Headers:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 6
Host: example.com:8080
User-Agent: Apache-HttpClient/4.2.6 (java 1.5)
```

Image source: www.blazemeter.com

## HTTP, UDV, FTP, Java, and JDBC

# HTTP

**HTTP Authorization Manager:** Among other configuration elements, users have the HTTP Authorization Manager.



HTTP Authorization Manager

Name: HTTP Authorization Manager

Comments:

Authorizations Stored in the Authorization Manager

| Base URL | Username | Password | Domain | Realm |
|---|---|---|---|---|
| http://localhost/phpmyadmin | root | •••• | | |

The Authorization Manager lets the user specify one or more user logins to web pages that are restricted using server authentication.

# HTTP

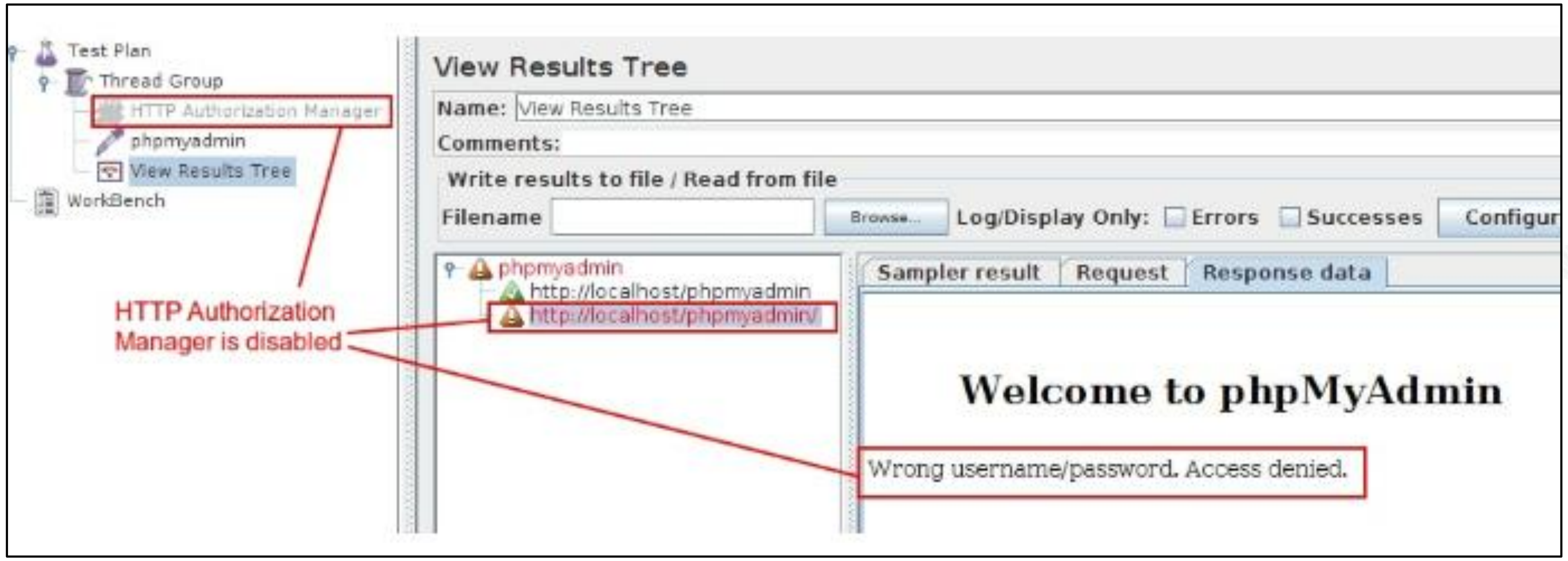
Users see authentication style when they attempt to access a restricted page, and the browser displays a login dialog box. JMeter transmits the login information when it encounters this type of page.

*Note*

Authorization headers are not shown in the View Results Tree listener, so users won't be able to check their values from the test script.

# HTTP

The three important fields are the base URL, username and password, and the HTTP authorization manager.



Name: View Results Tree

Comments:

Write results to file / Read from file

Welcome to phpMyAdmin

Wrong username/password. Access denied.

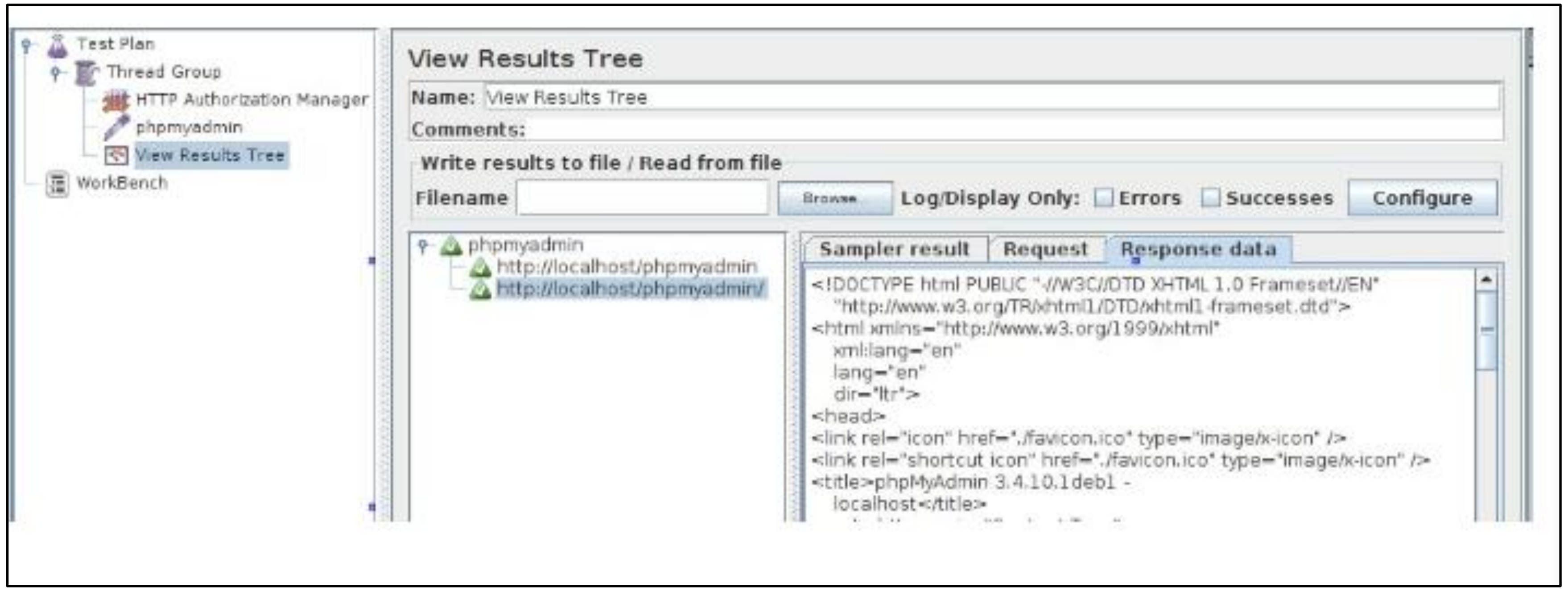
HTTP Authorization Manager is disabled

Image source: www.blazemeter.com

# HTTP

Since it is configured properly, the system does not provide access to the resources.

When the HTTP Authorization Manager is enabled, it works in this way:



View Results Tree

Name: View Results Tree

Comments:

Log/Display Only: □ Errors □ Successes

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xml:lang="en"
   lang="en"
   dir="ltr">
<head>
<link rel="icon" href="./favicon.ico" type="image/x-icon" />
<link rel="shortcut icon" href="./favicon.ico" type="image/x-icon" />
<title>phpMyAdmin 3.4.10.1deb1 -
   localhost</title>
```

Image source: www.blazemeter.com

Here, the server has given the authorization, and JMeter has received the HTML as a response.

# HTTP

**HTTP Cache Manager:** JMeter does not download static content until it is explicitly configured to do so.

If users have checked the **Retrieve All Embedded Resources** option, JMeter will download all static files during the execution.

# HTTP

It is required to include HTTP Cache Manager to achieve browser-like caching behavior.

The cache manager will save all these static files and not download them again from the server until modified.

# HTTP

**HTTP Cookie Manager:** It is used to store cookies that the targeted server sends in the response to the user's Http request.

Users can also add user-defined cookies, which will be shared with all the threads.

# HTTP

Cookies can be seen using the **View Results Tree** listener.

Such cookies usually have an expiration date.

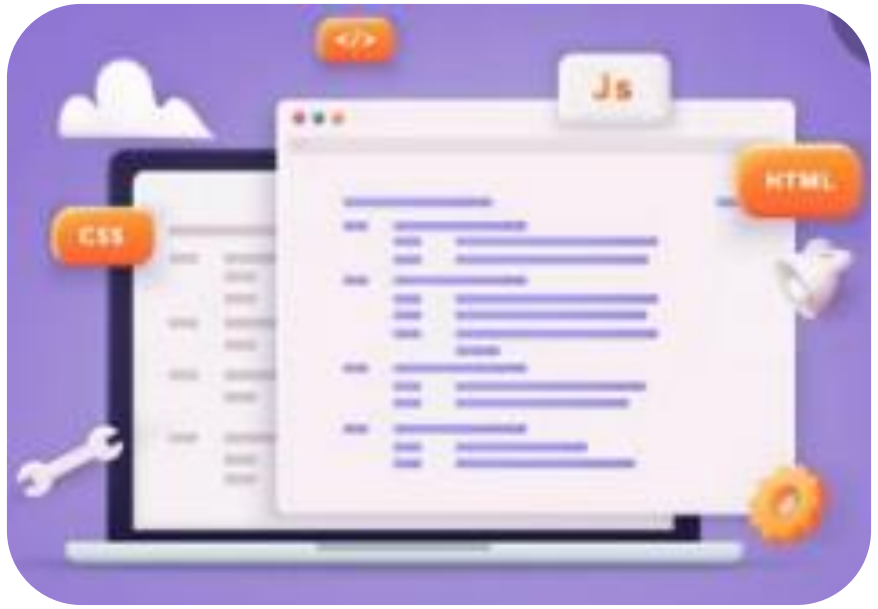
# HTTP

**HTTP Proxy Server:** Record Testing helps the tester to record & run their activity against the test target. It is a type of automated testing but for multiple users.

The Proxy Server allows JMeter to watch and record user activity while they are browsing web applications with a normal browser.

# HTTP

**HTTP Request Defaults:** HTTP Request Defaults Configuration Element lets users set default values to be used in HTTP Request Samplers.



# HTTP

**HTTP Header Manager:** The HTTP header manager in JMeter helps in containing and maintaining the HTTP headers that are sent to the server from the browser for scenario recording.

Every time the browser sends a request to a server, headers with additional information are attached to the request.

# User Defined Variables (UDV)

User defined variables element let users use default variables and values in the test plan.

If a user needs to use UDV in only one sampler, it can be defined under that sampler. If the user needs to use UDV in multiple parts, it must be defined at the start of the test plan.

# FTP Request Defaults

Apart from HTTP and HTTPS protocols, Apache JMeter also supports FTP protocols.

Load testing of FTP servers is important for determining their capacity, measuring throughput, and checking the impact of the severe load for a prolonged time while transferring the small, medium, and bulky files.

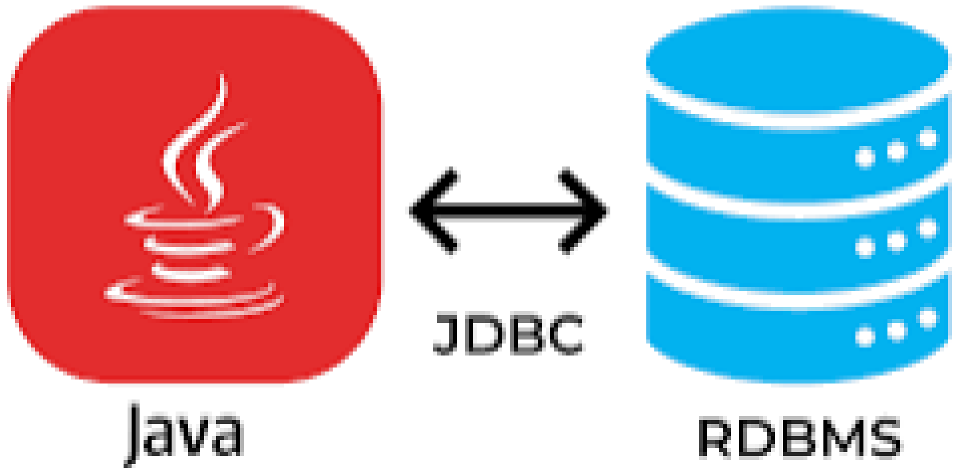
# JAVA Request Defaults

In JMeter, **Java Request Defaults** are used to set default parameters to pass them into Java Request.

Java Request Defaults config element is added under the test plan. The parameter values can also be overridden using Java Request.

# JDBC Connection Configuration

The JDBC Connection Configuration is used to configure JMeter connections to the database.



# JDBC Connection Configuration

When testing APIs, web services, or other system parts, users might need to record or retrieve data from a database.

The purpose of this interaction is to check the correct record of specific data in the DB or to prepare test data for the tests by adding specific records to the database.

TECHNOLOGY

# CSV Data Set Config

# Overview: CSV Data Set Config

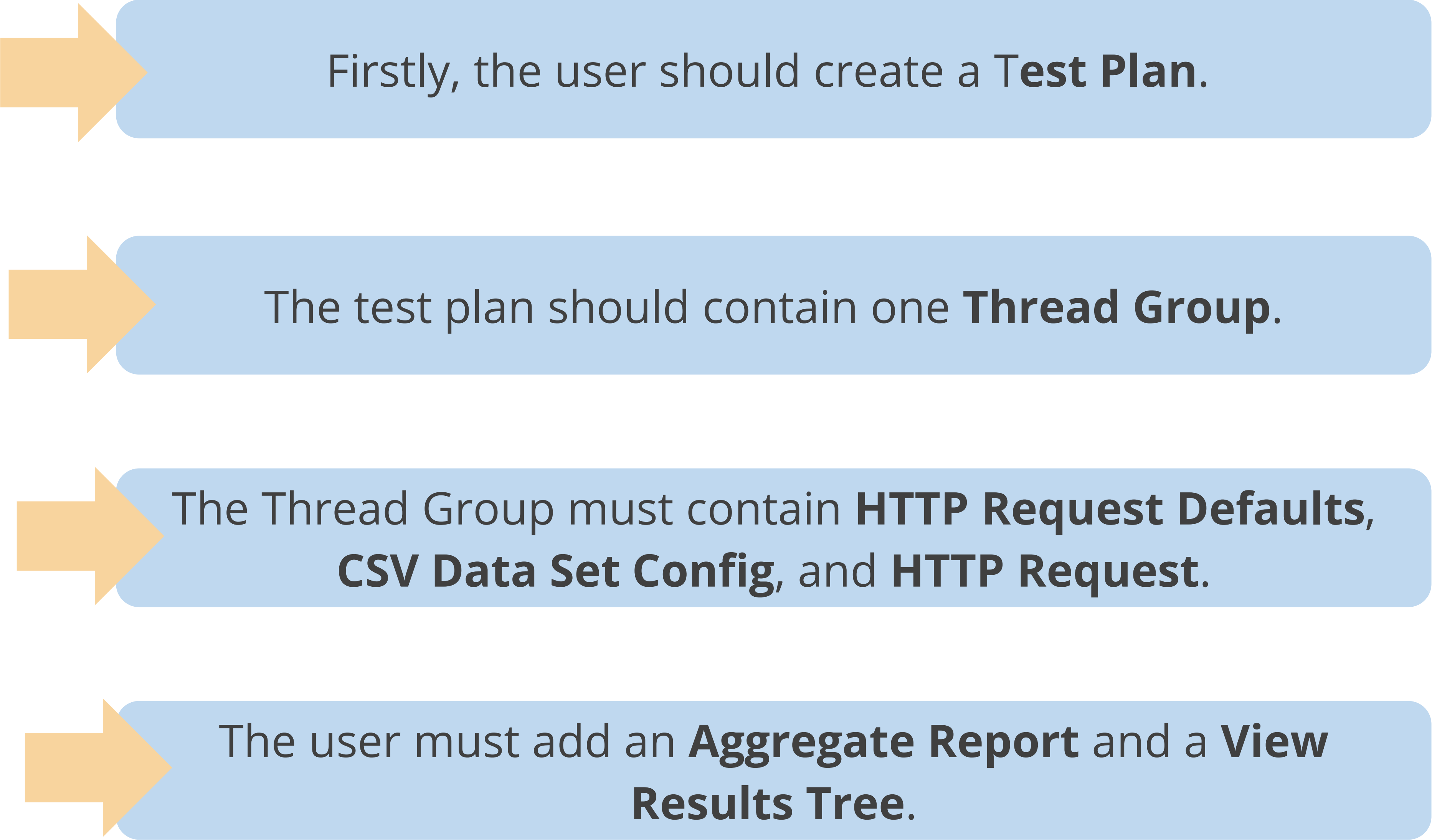
JMeter is an open-source load testing tool, which has an element that allows users to use external data sets in a CSV format.

This element is called the **CSV Data Set Config**.

The CSV Data Set Config is used to read lines from a file and split them into variables.

# Usage of CSV Data Set Config

Users need to follow a procedure to use the CSV data set config element while creating performance testing scripts in JMeter.



Firstly, the user should create a **Test Plan**.

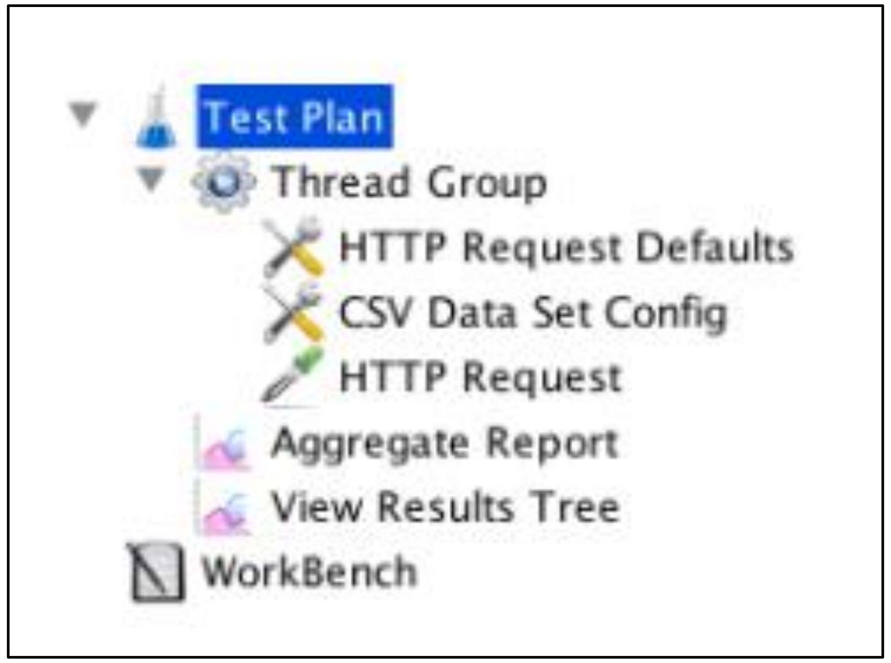
The test plan should contain one **Thread Group**.

The Thread Group must contain **HTTP Request Defaults**, **CSV Data Set Config**, and **HTTP Request**.

The user must add an **Aggregate Report** and a **View Results Tree**.

# Usage of CSV Data Set Config

After creating a test plan, it should look like this:

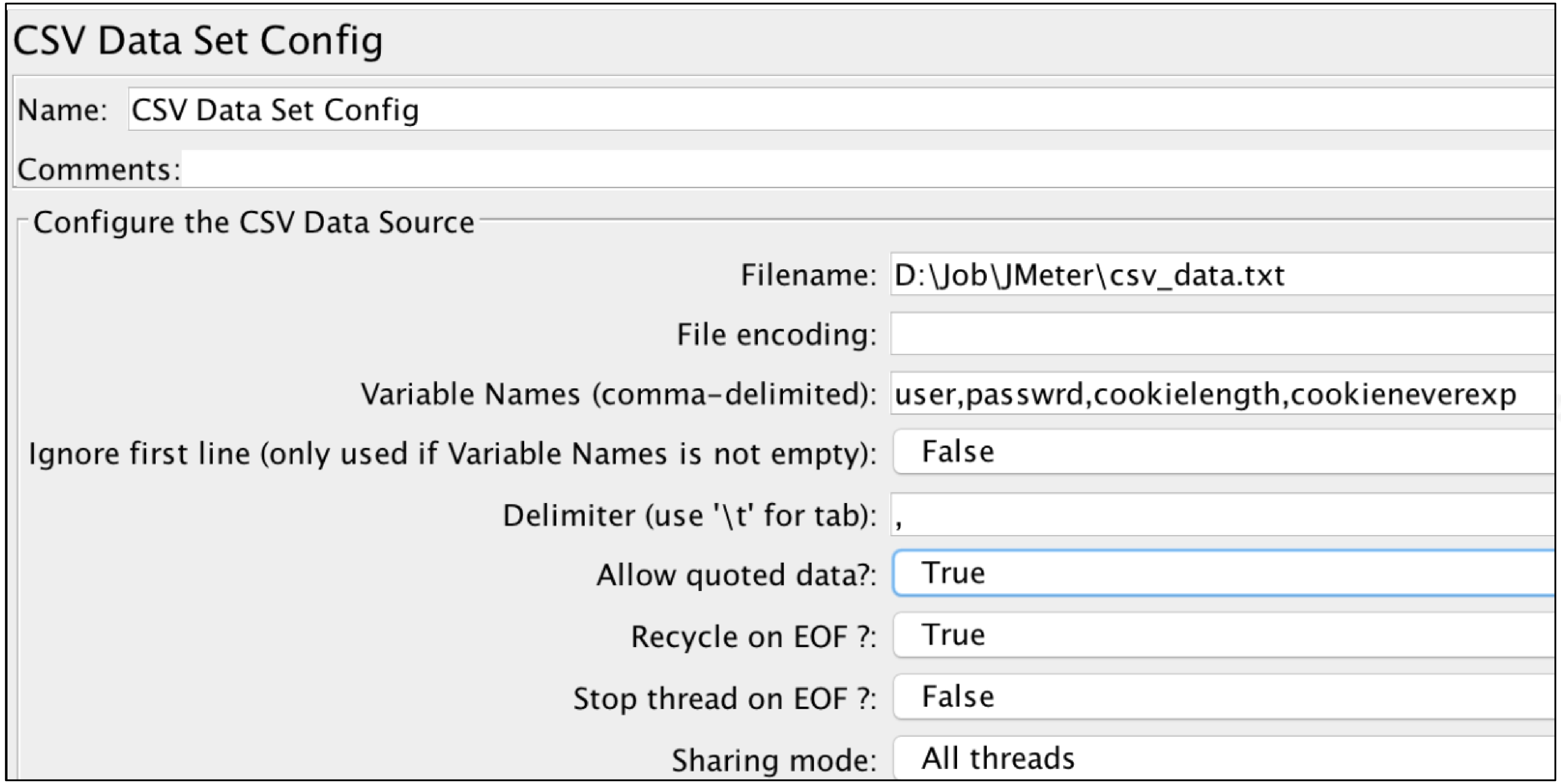


- Test Plan
  - Thread Group
    - HTTP Request Defaults
    - CSV Data Set Config
    - HTTP Request
  - Aggregate Report
  - View Results Tree
- WorkBench

Image source: www.blazemeter.com

# Configuring the CSV Data Set Config

The CSV Data Set Config configuration will look like this:



CSV Data Set Config

Name: CSV Data Set Config

Comments:

Configure the CSV Data Source

Filename: D:\Job\JMeter\csv_data.txt

File encoding:

Variable Names (comma-delimited): user,passwrd,cookielength,cookieneverexp

Ignore first line (only used if Variable Names is not empty): False

Delimiter (use '\t' for tab): ,

Allow quoted data?: True

Recycle on EOF ?: True

Stop thread on EOF ?: False

Sharing mode: All threads

# Configuring the CSV Data Set Config

In the image, there are several fields, out of which, at least three values should be present while others are optional:

- **Filename:** If the user's file is in the /bin directory, enter the filename only. If it's somewhere else, use the full path to the file.
- **Variable names:** Define the column name mapping as a list of comma-separated strings. To skip a column, add an extra comma with no name.

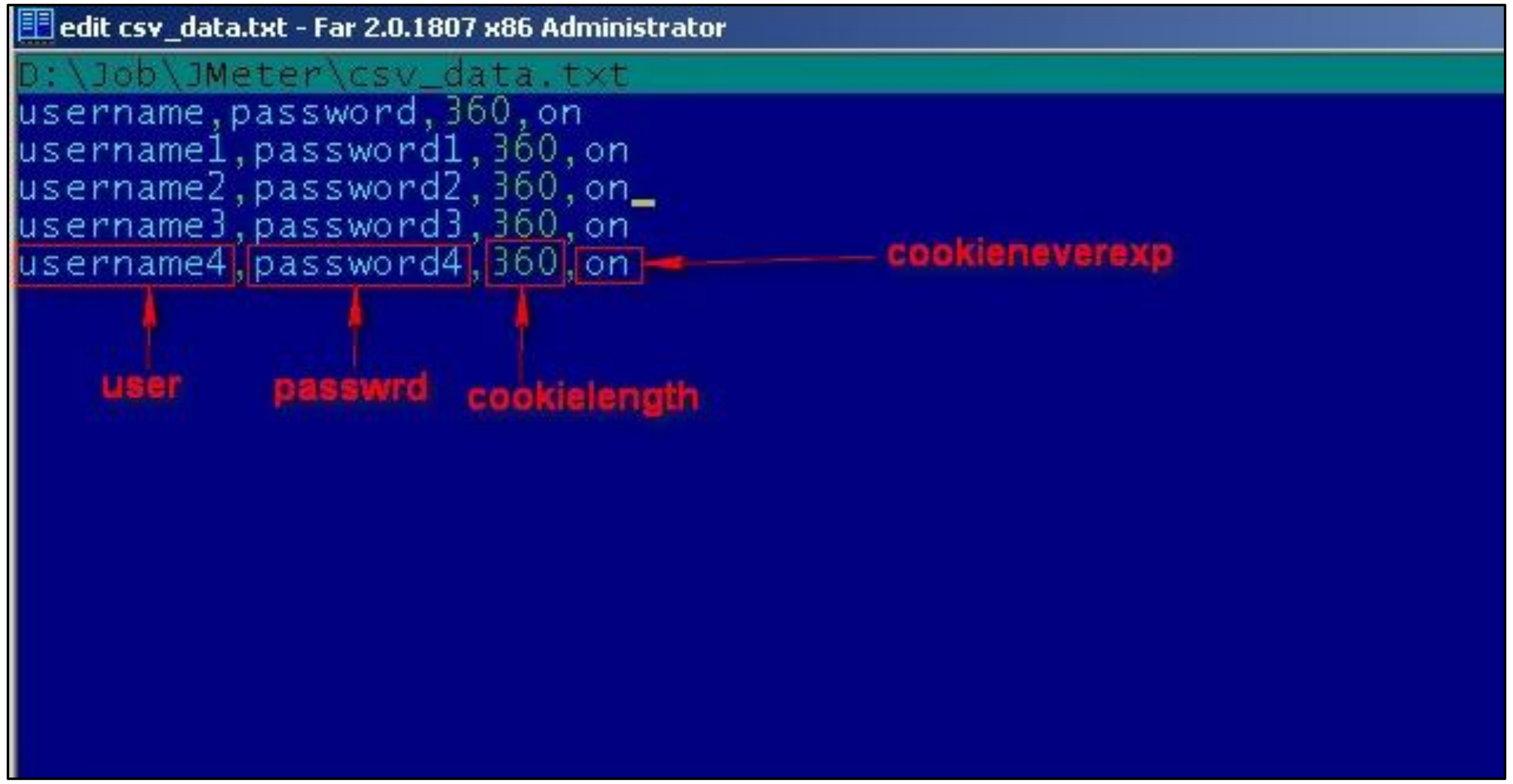
# Configuring the CSV Data Set Config

In the image, there are several fields, out of which, at least three values should be present while others are optional:

- **Delimiter:** A comma is the default delimiter, but if the file has tabs, enter \t here.
- **Ignore first line?** If the first row of the CSV file contains column names, the user can enable this option, but if the first row contains data, disable this option.
- **Allow quoted data?** If the user's column value contains commas, and commas are also used as delimiters, allow quoted values.

# Configuring the CSV Data Set Config

The user has sent four variables with the Login request. They are **user**, **passwrd**, **cookielength**, **cookieneverexp**.

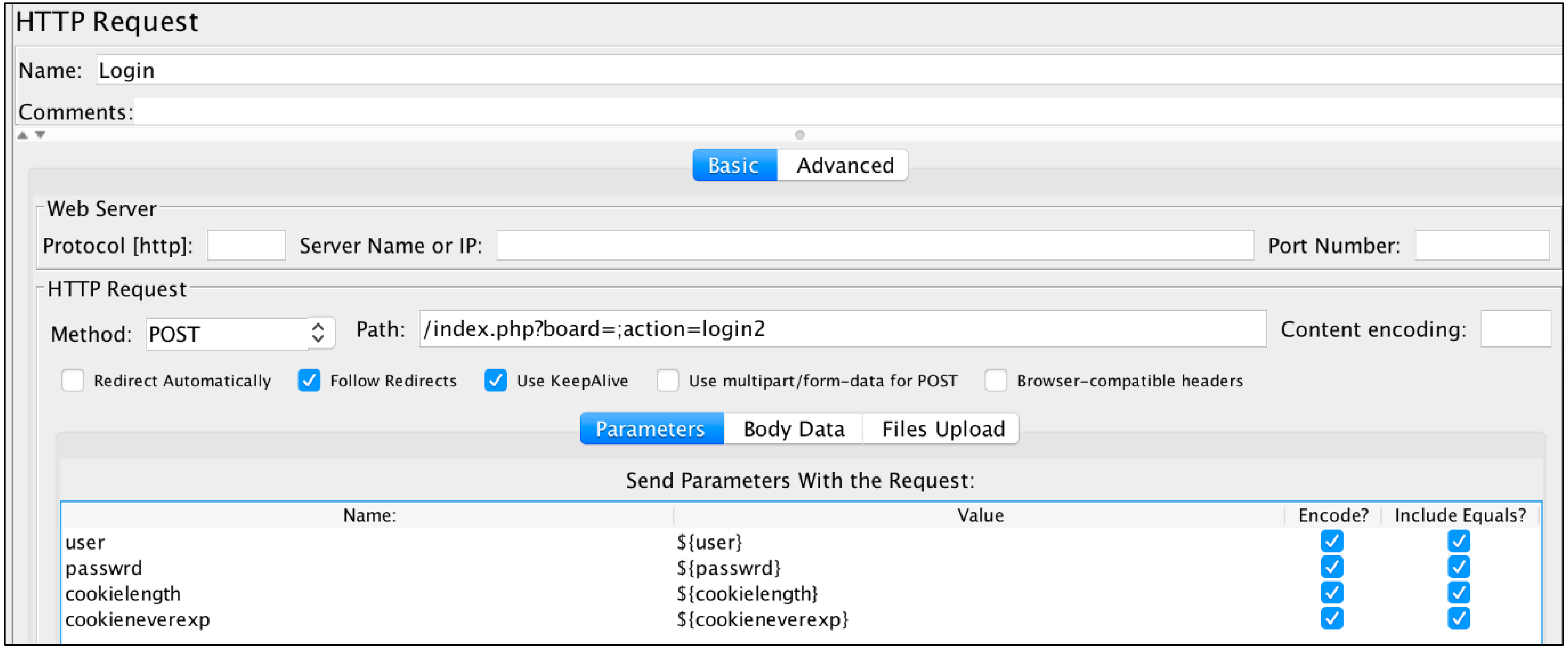


edit csv_data.txt - Far 2.0.1807 x86 Administrator

```
D:\Job\JMeter\csv_data.txt
username,password,360,on
username1,password1,360,on
username2,password2,360,on
username3,password3,360,on
username4,password4,360,on
```

# Configure the HTTP Request

In the case of Login request, it looks like this:



HTTP Request

Name: Login

Comments:

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

Method: POST Path: /index.php?board=;action=login2 Content encoding:

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data for POST □ Browser-compatible headers

Send Parameters With the Request:

| Name: | Value | Encode? | Include Equals? |
|---|---|---|---|
| user | ${user} | ☑ | ☑ |
| passwrd | ${passwrd} | ☑ | ☑ |
| cookielength | ${cookielength} | ☑ | ☑ |
| cookieneverexp | ${cookieneverexp} | ☑ | ☑ |

# Configure the HTTP Request

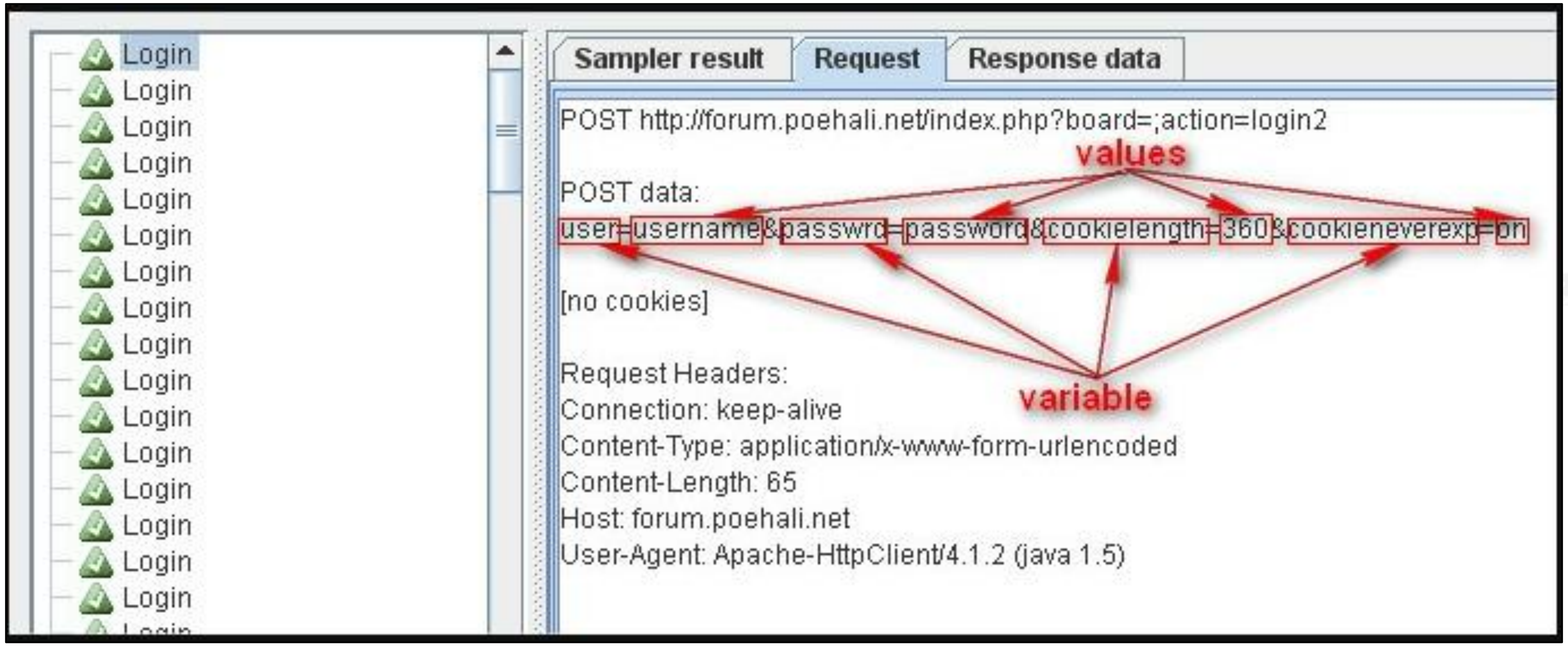
In the request, there are four variables with the same quantity as in CSV Data Set Config.

The value field has the same name as the variable in the 'CSV Data Set'. The construction ${....} means that it is a variable and not an absolute value.

# Running The Test Plan

When the user runs the test plan, JMeter will verify with **csv_data.txt** and change ${passwd}, and ${user} with their values.

The results will be displayed like this:



```
POST http://forum.poehali.net/index.php?board=;action=login2

POST data:
user=username&passwrd=password&cookielength=360&cookieneverexp=on

[no cookies]

Request Headers:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 65
Host: forum.poehali.net
User-Agent: Apache-HttpClient/4.1.2 (java 1.5)
```

# Running The Test Plan

Every CSV Data Set Config is visible to all Thread Groups by default.

If it is needed to use separate CSV Data Set Config elements for each Thread, the user can create the data files and set the CSV Data Set Config element from sharing mode to current thread.

# Key Takeaways

- Configuration elements allow users to create defaults and variables to be used by samplers.

- Random Variable config element is used to generate a random integer value between a specified range for each iteration.

- In JMeter, a counter is an element that enables users to create incremental values.

- JMeter is an open-source load testing tool, which has an element called CSV Data Set Config that allows users to use external data sets in a CSV format.