

Non-Functional Testing

Jmeter Testing

JMeter is a software that can perform load test, performance-oriented business (functional) test, regression test, etc., on different protocols or technologies.

It simulates a group of users sending requests to a target server, and returns statistics that show the performance/functionality of the target server/application via tables, graphs, etc.

We tested scalability of our system by hitting requests in large amount and doing crash test.

Running tests for

- a. Filter places:

The screenshot shows the JMeter HTTP Request configuration window. The 'Name' field is set to 'Filter Places'. The 'Comments' field is empty. The 'Basic' tab is selected, showing the 'Web Server' section with 'Protocol (http):' set to 'http', 'Server Name or IP:' set to 'localhost', and 'Port Number:' set to '4000'. The 'HTTP Request' section shows the 'Method' set to 'GET' and the 'Path' set to '/places/?country=USA'. The 'Content encoding' field is empty. Below this, there are checkboxes for 'Redirect Automatically' (unchecked), 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data' (unchecked), and 'Browser-compatible headers' (unchecked). The 'Parameters' tab is selected, showing a table for 'Send Parameters With the Request'.

Name	Value	URL Encode?	Content-Type	Include Equals?
------	-------	-------------	--------------	-----------------

b. Places page:

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

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

HTTP Request

GET Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

c. Specific place request:

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

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

HTTP Request

GET Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

Test 1:

a. Test Parameters

- 10 users
- 1 sec time interval between two consecutive requests
- 10 requests per users

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

☒ Continue
 ☐ Start Next Thread Loop
 ☐ Stop Thread
 ☐ Stop Test
 ☐ Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: ☐ Infinite

☒ Same user on each iteration
☐ Delay Thread creation until needed
☐ Specify Thread Lifetime

Duration (seconds):

Startup delay (seconds):

b. Results obtained

View Results Tree

Name:

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Search: ☐ Case sensitive ☐ Regular exp.

Text

☒ Sampler result
 ☐ Request
 ☒ Response data

Response Body Response headers

Find ☐ Case sensitive ☐ Regular exp.

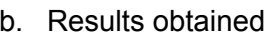
```
[{"_id":"644428f6ad596a355cbeb3e","owner":"642bd48992b05a33f2bceb3a","title":"as","address":"asd","photos":["d3d0cb60f6addc1e462770667079716d.jpeg"],"description":"asd","perks":[],"extralinfo":"asd","checkin":"null","checkout":"null","maxGuests":"1","price":"1000","_v":"0"},{"_id":"6446f44c8f1ce5129b1d162c3","owner":"6446b89c0b1ac52336a86526","title":"Burj Khalifa","address":"Dubai UAE","city":"Dubai","state":"Dubai","country":"Dubai","photos":["173XCVGFkord39om1TIBjaMLct6CY8XOI"],"description":"The best place to stay!","perks":["wifi","parking","tv","club","laundry"],"extralinfo":"","checkin":"17:21","checkout":"18:21","maxGuests":"15","price":"10000","_v":"0"},{"_id":"644901cf2ac9759853d53a06","owner":"6446b89c0b1ac52336a86526","title":"Better Inn","address":"hopperscotch","city":"washington","state":"Washington DC","country":"USA","photos":["15q4W97HrKn0QNIW_3td9S1VlbDmfz5TH"],"description":"kbenjkn","perks":["wifi","parking","pets","tv","laundry","club"],"extralinfo":"","checkin":"19:19","checkout":"20:19","maxGuests":"20","price":"5000","_v":"0"}]
```

☐ Scroll automatically?

Test 2:

a. Test parameters

- 100 users
- 2 sec time interval between two consecutive requests
- 100 requests per users



Test 3:

- a. Test parameters
 - 1000 users
 - 3 sec time interval between two consecutive requests
 - 100 requests per users

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Search: ☐ Case sensitive ☐ Regular exp.

Text

Specific place request
Filter Places
Specific place request
Specific place request
Filter Places
Specific place request
places page
places page
places page
Specific place request
Specific place request
places page
Filter Places
Filter Places
places page
Filter Places
Specific place request
Filter Places
places page
Specific place request
Filter Places
Filter Places
Filter Places
places page
places page
Filter Places
Filter Places
Filter Places
Filter Places
Scroll automatically?

Sampler result Request Response data

Response Body Response headers

Find ☐ Case sensitive ☐ Regular exp.

```
java.net.SocketException: Socket closed
    at java.net.SocketInputStream.socketRead0(Native Method)
    at java.net.SocketInputStream.socketRead(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:137)
    at org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:153)
    at org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:280)
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:138)
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:56)
    at org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:259)
    at org.apache.http.impl.DefaultHttpClientConnection.receiveResponseHeader(DefaultHttpClientConnection.java:163)
    at org.apache.http.impl.conn.CPooledProxy.receiveResponseHeader(CPooledProxy.java:157)
    at org.apache.http.protocol.HttpRequestExecutor.doReceiveResponse(HttpRequestExecutor.java:273)
    at org.apache.http.protocol.HttpRequestExecutor.execute(HttpRequestExecutor.java:125)
    at org.apache.http.impl.execchain.MainClientExec.execute(MainClientExec.java:272)
    at org.apache.http.impl.execchain.ProtocolExec.execute(ProtocolExec.java:186)
    at org.apache.http.impl.execchain.RetryExec.execute(RetryExec.java:89)
    at org.apache.http.impl.execchain.RedirectExec.execute(RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute(InternalHttpClient.java:185)
    at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:83)
    at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.executeRequest(HTTPHC4Impl.java:939)
    at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.sample(HTTPHC4Impl.java:650)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerProxy.sample(HTTPSamplerProxy.java:56)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1301)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1290)
    at org.apache.jmeter.threads.JMeterThread.doSampling(JMeterThread.java:651)
    at org.apache.jmeter.threads.JMeterThread.executeSamplePackage(JMeterThread.java:570)
    at org.apache.jmeter.threads.JMeterThread.processSampler(JMeterThread.java:501)
    at org.apache.jmeter.threads.JMeterThread.run(JMeterThread.java:268)
    at java.lang.Thread.run(Unknown Source)
```

Test 4:

a. Test Parameters

- 1000 users
- 0.1 sec time interval between two consecutive requests
- 1000 requests per users

Thread Group

Name: Test grp 1

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users): 1000

Ramp-up period (seconds): 0.1

Loop Count: ☐ Infinite 1000

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

b. Results

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
places page	133109	6646	544	8065	456.35	0.25%	50.1/sec	68.32	6.10	1395.9
Filter Places	132779	6657	570	7952	393.36	0.26%	50.0/sec	36.17	6.72	740.7
Specific place request	132434	6663	566	7650	353.82	0.25%	49.9/sec	35.98	7.29	738.5
TOTAL	398322	6655	544	8065	403.54	0.25%	150.0/sec	140.45	20.11	958.9

View Results Tree

Name:

Comments:

Write results to file / Read from file

Filename: ☐ Errors ☐ Successes

Search: ☐ Case sensitive ☐ Regular exp.

Text

- Specific place request
- Specific place request
- Specific place request
- Filter Places
- places page
- Specific place request
- Specific place request
- places page
- Filter Places
- Filter Places
- Specific place request
- Filter Places
- Filter Places
- places page
- Specific place request
- Specific place request
- places page
- Specific place request
- Specific place request
- places page
- Filter Places
- Filter Places
- Specific place request
- places page
- places page
- places page
- Filter Places
- Specific place request
- Filter Places
- places page
- places page
- Filter Places
- Specific place request

Sampler result Request Response data

Response Body Response headers

Find ☐ Case sensitive ☐ Regular exp.

```
java.net.SocketException: Socket closed
    at java.net.SocketInputStream.socketRead0(Native Method)
    at java.net.SocketInputStream.socketRead(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at org.apache.http.impl.io.SessionInputBufferImpl.streamRead(SessionInputBufferImpl.java:137)
    at org.apache.http.impl.io.SessionInputBufferImpl.fillBuffer(SessionInputBufferImpl.java:153)
    at org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:280)
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:138)
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:36)
    at org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:259)
    at org.apache.http.impl.DefaultBHttpClientConnection.receiveResponseHeader(DefaultBHttpClientConnection.java:163)
    at org.apache.http.impl.conn.CPoolProxy.receiveResponseHeader(CPoolProxy.java:157)
    at org.apache.http.protocol.HttpRequestExecutor.doReceiveResponse(HttpRequestExecutor.java:273)
    at org.apache.http.protocol.HttpRequestExecutor.execute(HttpRequestExecutor.java:125)
    at org.apache.http.impl.execchain.MainClientExec.execute(MainClientExec.java:272)
    at org.apache.http.impl.execchain.ProtocolExec.execute(ProtocolExec.java:186)
    at org.apache.http.impl.execchain.RetryExec.execute(RetryExec.java:89)
    at org.apache.http.impl.execchain.RedirectExec.execute(RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute(InternalHttpClient.java:185)
    at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:83)
    at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.executeRequest(HTTPHC4Impl.java:939)
    at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.sample(HTTPHC4Impl.java:650)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerProxy.sample(HTTPSamplerProxy.java:66)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1301)
    at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1290)
    at org.apache.jmeter.threads.JMeterThread.doSampling(JMeterThread.java:651)
    at org.apache.jmeter.threads.JMeterThread.executeSamplePackage(JMeterThread.java:570)
    at org.apache.jmeter.threads.JMeterThread.processSampler(JMeterThread.java:501)
    at org.apache.jmeter.threads.JMeterThread.run(JMeterThread.java:268)
    at java.lang.Thread.run(Unknown Source)
```

☐ Scroll automatically?