

Basic Load Tester (BLT)

BLT is NOT meant as a replacement of other loading testing utilities such as [JMeter](#), [gatling.io](#), [Locust](#), [LoadRunner](#), etc. If you are presently and consistently using any of these testing utilities please continue to do so. BLT is a simple load generating and testing utility which focuses more on finding when a particular load or loads push an environment to a point where response times exceed a certain threshold. It was specifically created to work with the exposed REST endpoints of the ForgeRock platform.

```
java -jar ./dist/BLT.jar --help
```

```
BLT usage: java -jar ${BLT_HOME}/dist/BLT.jar [FILE]
```

options:

if not specified [FILE] defaults to `${BLT_HOME}/sample/config.json`

[FILE] example: `${BLT_HOME}/mytest/myconfig.json` and does require a layout similar to

`${BLT_HOME}/sample/`

`--csv` | `-v` displays results in a comma delimited format

`--curl` | `-c` displays example cURL commands used against REST endpoints for this test

`--increment n` | `-i` increment the "threads" value by one until the value(s) are incremented by the number n.

All other options are turned off. Good option for reaching resource contention.

`--job` | `-j` displays JSON configuration data used for the test

`--summary` | `-s` displays summary of test

`--progress` | `-p` displays progress while BLT is running

`--help` | `-h` this output

Examples:

```
java -jar ${BLT_HOME}/dist/BLT.jar
```

```
java -jar ${BLT_HOME}/dist/BLT.jar --csv --curl --job --summary --progress  
${BLT_HOME}/mytest/myconfig.json
```

BLT uses JSON configuration files that are in a hierarchical structure. There is the initial configuration file with the desired environment variables are set. This configuration JSON file can be called whatever you like but it must contain specific attributes. The values of the attributes in the **BLT-environment** section are used to replace wildcards while running. All of these attributes must start with `"$BLT-"` with `$BLT-` in uppercase. Best not mess with the attributes and values in the **BLT-reserved** section as these are used internally by BLT. For example:

```
{  
  "name": "base-config",  
  "job": "./sample/job/base-job.json",  
  "workload": "./sample/base-workload/",  
  "task": "./sample/base-task/",  
  "BLT-environment": {
```

```

"$BLT-AMHOST0": "http://am0.example.com:8080",
"$BLT-AMHOST1": "http://am1.example.com:8080",
"$BLT-DSHOST0": "http://ds0.example.com:8080",
"$BLT-DSHOST1": "http://ds1.example.com:8080",
"$BLT-IDMHOST0": "http://idm0.example.com:8080",
"$BLT-IDMHOST1": "http://idm1.example.com:8080",
"$BLT-IGHOST0": "http://ig0.example.com:8080",
"$BLT-IGHOST1": "http://ig1.example.com:8080",
"$BLT-MINVALUE": 0,
"$BLT-MAXVALUE": 200000,
"$BLT-AM-REALM": "am",
"$BLT-AMADMINID": "amadmin",
"$BLT-AMADMINPASSWORD": "password",
"$BLT-DSADMINID": "adminuser",
"$BLT-DSADMINPASSWORD": "password",
"$BLT-IDMADMINID": "openidm-admin",
"$BLT-IDMADMINPASSWORD": "openidm-admin",
"$BLT-IDMUSERID": "user1",
"$BLT-IDMUSERPASSWORD": "Welcome1",
"$BLT-IDMMANAGERID": "manager1",
"$BLT-IDMMANAGERPASSWORD": "Welcome1",
"$BLT-PASSWORD": "password"
},
"BLT-reserved": {
  "$BLT-RANDOM-NUMBER": "random-long",
  "$BLT-RANDOM-ALPHA-STRING": "Alpha only - not implemented",
  "$BLT-RANDOM-ALPHANUMERIC-STRING": "Alpha and number - not implemented",
  "$BLT-THREADID": "threadid-int",
  "$BLT-INCREMENT": "long",
  "$BLT-DECREMENT": "long",
  "$BLT-TOKEN-PAYLOAD": "tokenId",
  "$BLT-SLEEP": "sleep"
}
}

```

The above configuration file lists the three core aspects of BLT: job, workload, and task. Using the analogy of a house construction project "job" relates to the overall project at hand. An example job JSON file:

```

{
  "name": "basejob",
  "load-generator": "My MacBook Pro",
  "services-description": "Testing BLT",

```

```

    "service-location-port": ["http://am0.example.com:8080",
"http://am1.example.com:8080"],
    "loading-style": "roundrobin",
    "minvalue": 0,
    "maxvalue": 200000,
    "threshold-to-error": 99,
    "threshold-to-fail": 999,
    "continue-on-fail": false,
    "workload": [
        {
            "name": "baseworkload",
            "iteration": 2,
            "start-delay": 0,
            "threads": 2,
            "thread-group-size": 0,
            "thread-interval": 10000,
            "minvalue": 0,
            "maxvalue": 100000,
            "threshold-to-error": 88,
            "threshold-to-fail": 90,
            "continue-on-fail": false
        },
        {
            "name": "baseworkload2",
            "iteration": 1,
            "start-delay": 9,
            "threads": 1,
            "thread-group-size": 0,
            "thread-interval": 200
        }
    ]
}

```

The job configuration can have attributes and values set at a high level which, if not specified at the workload or task level, are used. Looking at "baseworkload" both minvalue and maxvalue are overridden. A description of all the attributes, the valid values, and scope of usage is below. Note that any value or part of a value in a job, workload, or task configuration can be substituted using wildcards specified in the config file.

Nomenclature: R = Required; R-I = Required and can be inherited; O = Optional; O-I = Optional and can be inherited;

Attribute	Purpose	Valid Values	Config	Job	Workload	Workload	Task	Task
-----------	---------	--------------	--------	-----	----------	----------	------	------

					array (workload set) (within the Job)		Array (task set) (within the workload)	
name	used to locate file	Very important to make sure "name" is the same as the actual JSON file name.	R	R	R	R	N/A	R
job	(within the configuration file) location of job file	path to job configuration file (i.e. job.json)	R	N/A	N/A	N/A	N/A	N/A
workload	(within the configuration file) location of workload file(s)	path to directory/file containing workload configuration file(s)	R	N/A	N/A	N/A	N/A	N/A
task	(within the configuration file) location of task file(s)	path to directory/file containing task configuration file(s)	R	N/A	N/A	N/A	N/A	N/A
BLT-environment	User defined wildcards	Strings or integer	R	N/A	N/A	N/A	N/A	N/A
BLT-reserved	System defined wildcards	Determined by system	R	N/A	N/A	N/A	N/A	N/A
load-generator	descriptive	text describing the system used to generate the load	N/A	R	N/A	N/A	N/A	N/A
services-description	descriptive	text describing the type of test being performed i.e. test authentication rate	N/A	R	N/A	N/A	N/A	N/A
service-location-port	JSON array of name and port of service(s)	http://am0.example.test:8080 or https://192.168.0.2:8443	N/A	R	R-I	R-I	R-I	R-I
loading-style	specifies how to use the endpoints listed in the service-location-port array if more than one endpoint is listed	none, roundrobin, random, or ha. Must be lower-case. none = uses the first string in the array roundrobin = cycles through the endpoints listed in the service-location-port array. random = randomly picks one of the endpoints listed in the service-location-port array. ha = uses the first endpoint listed in the service-location-port array and only tries the next endpoint if the first one fails.	N/A	O	O-I	O-I	O-I	O-I (default = none)
minvalue	integer specifies value used for \$BLT-RANDOM-NUMBER, starting value for \$BLT-	integer	N/A	R	R-I	R-I	R-I	R-I

	INCREMENT, ending value for \$BLT-DECREMENT							
maxvalue	integer specifies value used for \$BLT-RANDOM-NUMBER, ending value for \$BLT-INCREMENT, starting value for \$BLT-DECREMENT	integer	N/A	R	R-I	R-I	R-I	R-I
threshold-to-error	maximum time (milliseconds) allowed before an error is thrown on a transaction	integer; value = 0 means do not execute task	N/A	R	R-I	R-I	R-I	R-I
threshold-to-fail	maximum time (milliseconds) allowed before the connection is closed as a failure	integer; value = 0 means do not execute task	N/A	R	R-I	R-I	R-I	R-I
continue-on-fail	if threshold-to-fail is exceeded should following tasks within that task set be attempted	true or false	N/A	R	R-I	R-I	R-I	R-I
workload	within a job configuration is an array (set) of workloads	json objects	N/A	R	N/A	N/A	N/A	N/A
iteration	number of times to repeat the workload	integer	N/A	N/A	R	N/A	N/A	N/A
threads	number of threads to run a workload If iterations = 4 and threads = 2 then the workload will be executed 8 times.	integer	N/A	N/A	R	N/A	N/A	N/A
start-delay	number of milliseconds before starting the workload	integer (default = 0 means no delay)	N/A	N/A	R	N/A	N/A	N/A
thread-group-size	number of threads to launch at a time	integer (default = 0 means all threads)	N/A	N/A	R	N/A	N/A	N/A
thread-interval	time in milliseconds between starting a thread group if threads = 10, thread-group-size = 2 and thread-interval = 4 then every 4 seconds two threads will be executed a total of 5 times	integer	N/A	N/A	R	N/A	N/A	N/A

task	within a workload configuration is an array (set) of tasks	json objects	N/A	N/A	N/A	R	N/A	N/A
write-file	specifies path and file name (requires .json to be included on the file name i.e. ./bulk-task/mycleanup-file.json) to write a file with a task per line that can be consumed via read-file for cleanup	valid path and file name i.e. \$HOME/BLT/bulk-task/mycleanup-file.json	N/A	N/A	N/A	N/A	O	N/A
read-file	specifies path and file name (without .json)	valid path and file name i.e. \$HOME/BLT/bulk-task/mycleanup-file	N/A	N/A	N/A	N/A	O	N/A
request	type of REST request	GET, POST, DELETE, etc	N/A	N/A	N/A	N/A	N/A	R
header	header attribute value pairs needed for the transaction	Standard HTTP headers	N/A	N/A	N/A	N/A	N/A	R
url-endpoint	URI path that is appended to the host:port	i.e. /am/json/realms/root/	N/A	N/A	N/A	N/A	N/A	R
url-payload	action or value appended to the url-endpoint	i.e. authenticate or ?_action=validate	N/A	N/A	N/A	N/A	N/A	R
data-payload	additional data elements as dictated by the endpoint	depends on the endpoint	N/A	N/A	N/A	N/A	N/A	O