Statistics

Table of contents

1 Standard statistics

Details of the statistics provided by The Grinder can be found in the documentation of the Statistics (../../g3/script-javadoc/net/grinder/script/Statistics.html) interface. Scripts can use this interface to:

- Query whether a test was successful
- Obtain statistic values, such as the test time of the last test
- Modify or set a test's statistics before they are sent to the log and the console
- Report custom statistics
- Register additional views of standard and custom statistics

2 Distribution of statistics

All the statistics displayed in the console are aggregates (totals or averages) of a number of tests received in the appropriate period. The reason for this is efficiency. The Grinder would not perform or scale if every data point was transferred back to the console.

The only place per-test statistics are available is in the process `data_*` files.

3 Querying and updating statistics

A script can query the statistics about the last completed test using grinder.statistics.forLastTest (../../g3/script-javadoc/net/grinder/script/Statistics.html#getForLastTest()) . Script code instrumented by a test can access information about the statistics for the test (which may be incomplete) using grinder.statistics.forCurrentTest (../../g3/script-javadoc/net/grinder/script/Statistics.html#getForCurrentTest()) . For details of the query and update methods, see StatisticsForTest (../../g3/script-javadoc/net/grinder/script/Statistics.StatisticsForTest.html) . Refer to the documentation of the Statistics (../../g3/script-javadoc/net/grinder/script/Statistics.html) interface for other details.

An example script (../g3/script-gallery.html#statistics.py) demonstrating these APIs can be found in the Script Gallery.

4 Registering new expressions

Custom statistic expressions can be added to console views and the worker process summary tables (found in the `out_*` log files) using the registerSummaryExpression (../../g3/script-javadoc/net/grinder/script/Statistics.html#registerSummaryExpression(java.lang.String, java.lang.String)) method.

Custom expressions can be added to worker process `data_*` using the registerDataLogExpression (../../g3/script-javadoc/net/grinder/script/Statistics.html#registerDataLogExpression(java.lang.String, java.lang.String)) method.

Both methods take a *displayName* and an *expression* as parameters.

The *displayName* is the label used for the expression. For expressions displayed in the console, this string is converted to a key for an internationalised resource bundle look up by prefixing the string with `statistic.` and replacing any whitespace with underscores; if no value for the key exists, the raw display name string is used.

Expressions are composed of statistic names (see Statistics (../../g3/script-javadoc/net/grinder/script/Statistics.html)) in a simple post-fix format using the symbols +, -, /

and `*`, which have their usual meanings, in conjunction with simple statistic names or sub-expressions. Precedence is controlled by grouping expressions in parentheses. For example, the error rate is `(* (/ errors period) 1000)` errors per second. The symbol `sqrt` can be used to calculate the square root of an expression.

Sample statistics, such as `timedTests`, must be introduced with one of `sum`, `count`, or `variance`, depending on the attribute of interest. For example, the statistic `expression (/ (sum timedTests) (count timedTests))` gives the mean test time in milliseconds.