

Reporting Test Metadata

On this page:

- [Reporting additional test data](#)
 - [Number](#)
 - [String/Text](#)
 - [External links](#)
 - [Links to build artifacts](#)
 - [Image from artifacts directory](#)
- [Displaying additional test data](#)
 - [Additional test data for a failed test](#)
 - [Additional test data graph for numeric values](#)
 - [Additional data for successful tests](#)

Since TeamCity 2018.2, a test run in TeamCity can be associated with some additional information (metadata), complementing test status, execution time, and output. This information can be used to provide extra logs, screenshots, numeric values, tags etc.

You can now use service messages to report this kind of additional test data in TeamCity and then view it in the TeamCity Web UI.

Reporting additional test data

Additional test data is reported using the `testMetadata` service message, with the following attributes:

- `name` (optional) is a name of the metadata item, used to distinguish different metadata items for the same test and to show this item in UI. When omitted, TeamCity autogenerates the name, but does not show it in UI.
- `testName` (optional) is the name of the test for which this metadata is generated, excluding the suite name. If the service message is reported in the middle of the test or immediately after its finish, the `testName` attribute can be skipped. If the metadata is reported later, it should contain the full test name, including suites, packages and so on.
- `value` (mandatory) contains the metadata value associated with the test. The max value length is 1024.
- `type` (optional) defaults to `'text'` unless otherwise specified. The type affects how metadata is stored and how its value is shown in the TeamCity UI.

If the format of the service message is incorrect, a corresponding note about it is written into build log. The types of data which can be recognised by TeamCity are as follows:

Number

```
##teamcity[testMetadata testName='package.Test.testName' name='setUp time' type='number' value='434.5']
```

You can see a graph of changes for a numeric value, from build to build for the given test.

String/Text

```
##teamcity[testMetadata testName='package.Test.testName' name='some key' value='a text']
```

External links

```
##teamcity[testMetadata testName='test.name' name='JetBrains' type='link'  
value='https://jetbrains.com']
```

Links to build artifacts

```
##teamcity[testMetadata testName='test.name' type='artifact' value='path/to/catalina.out']
```

The path to the artifact should be relative to the [build artifacts directory](#), and can reference a file inside archive:

```
##teamcity[testMetadata testName='test.name' type='artifact'  
value='logs.zip!/testTyping/full-log.txt']
```

When showing links to artifacts, TeamCity shows both the `'name'` and the filename of the referenced image. If name was autogenerated, it is not shown.

Image from artifacts directory

```
##teamcity[testMetadata testName='test.name' type='image' value='path/to/screenshot.png']
```

The path to the image should be relative to the build artifacts directory. When showing images, TeamCity shows both the `'name'` and the filename of the referenced image.

Displaying additional test data

You can view additional test data in various places in the TeamCity Web UI.

Additional test data for a failed test

To view additional data for a test, expand the stacktrace: TeamCity shows it before test failure details in a separate Test Metadata section.

1 test failed (no new)

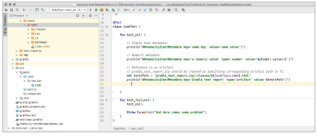
Gradle suite: Gradle test: foo (1)

CodeTest.test_failure | show details

Test metadata (4)

Gradle test report [foo.CodeTest.html]

Some screenshot [pictureForAttention.png]



a numeric value	481.141693
some key	some value

```

java.lang.Exception: And here comes some problem
    at foo.CodeTest.test_failure(CodeTest.kt:34)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:124)
    at org.testng.internal.Invoker.invokeMethod(Invoker.java:583)
    at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:719)
    at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:989)
    at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:125)
    at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
    at org.testng.TestRunner.privateRun(TestRunner.java:648)
    at org.testng.TestRunner.run(TestRunner.java:505)
    at org.testng.SuiteRunner.runTest(SuiteRunner.java:455)
    at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:450)

```

Additional test data graph for numeric values

To view the test data graph, expand the stacktrace: TeamCity shows it before test failure details in a separate Test Metadata section.

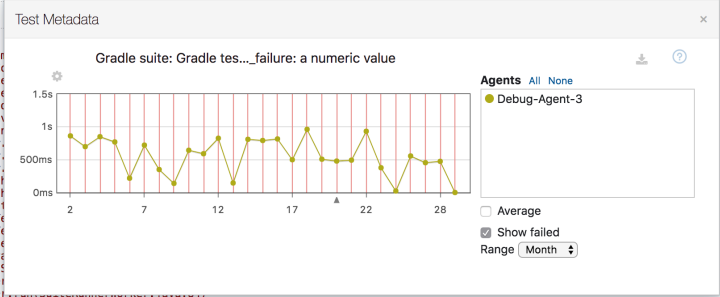
a numeric value 481.141693

some key

```

java.lang.Exception: And here comes some problem
    at foo.CodeTest.test_failure(CodeTest.kt:34)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:124)
    at org.testng.internal.Invoker.invokeMethod(Invoker.java:583)
    at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:719)
    at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:989)
    at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:125)
    at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
    at org.testng.TestRunner.privateRun(TestRunner.java:648)
    at org.testng.TestRunner.run(TestRunner.java:505)
    at org.testng.SuiteRunner.runTest(SuiteRunner.java:455)
    at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:450)
    at org.testng.TestNG.runSuitesSequentially(TestNG.java:1288)

```



Additional data for successful tests

To view additional data for a successful test, go to the Tests tab or the Test History tab: the OK status for a test is now clickable if additional test data is present:

Total test count: 2 (1 failed); total duration: < 1s

View: tests containing: with: any status Filter

Status Test

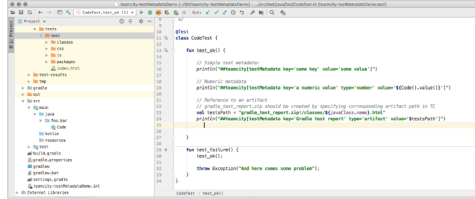
Failure CodeTest.test_failure | (Gradle suite: Gradle test.fo)

OK CodeTest.test_ok | (Gradle suite: Gradle test.fo)

Test metadata (4)

Gradle test report [foo.CodeTest.html]

Some screenshot [pictureForAttention.png]



a numeric value 938.369812

some key some value