

I'm not robot  reCAPTCHA

**Continue**

## Junit test report jenkins

Info: Start recording, Info: JUnit Information Processing: [JUnit] - 4 Test report files were found in a ""/unit test' pattern compared to 'E:\Jenkins-ws{folder name}' for the test framework .xml DUnit', Jenkins is getting on top of the message while processing JUnit. My test was successful and Jenkins was getting failed due to this problem. Here is my karma.config file. Basepass: "", Framework: ['Jasmine', '@angular-Devkit/Build-Angle'], Plug-ins: ['Karma Jasmine'], Need ('Karma Chrome Launcher'), Need ('Karma Jasmine-html Reporter'), Need ('Karma Range-Istanbul Reporter'), Need ('Karma Range-Istanbul Reporter'), Need ('@angular-html Reporter') devkit/build-angular/plugins/karma), required ('karma-junit-reporter') ['karma-junit-reporter'], Client: {Clear context: False // Leave jasmine spec runner output in browser }, junitReporter: { OutputDir: Test Results, Output File: Unit Test.xml} }, CoverageIstanbulReporter: {dir: Required ('Path').join (\_\_dirname, './coverage'), Report: ['html', 'lcovonly'], fixWebpackSourcePaths: true}, Reporter: ['Progress', 'Junit', 'khtml'], Jenkins Test Completed Test Report Applied to Test Completion 14.71, Last modified on November 27, 2020 after the build was completed, you will see the results of your test completion. You can view the results on the Build page. If generating a JUnit style report or a JUnit report is enabled, you can also view the results in a way that is appropriate for this type of report. For example, on the Test Results page of a build. You can configure the JUnit report test completion test phase to export test results in JUnit report format. These reports are similar to those generated by the JUnit test. Note: This format is different from the JUnit style report described below. Report generation: During the TestComplete Test build phase, specify the /Export reagent command-line arguments in the Additional Command Line Arguments text box and specify the path to the folder where you want to store the generated test results. Specify statement argument parameters: Test Complete Loan Launch Type: 'lcProject', Project: 'MyProj', Product Family: 'ProjectJenkins Tests', CommandLine: '/ExportYolysk::C:\Workspace\TestReport\Results.xml The TestComplete test step generates a report and saves it to the specified location. Unlike the JUnit style report described below, the TestComplete test phase does not automatically publish this report. You must publish the report manually (publish the regular JUnit report in the same way). For example, you can publish a report using the JUnit plug-in. JUnit Style Test Report If generating a JUnit style report is enabled for the test (see the JUnit style report generation option in the Test Complete Test phase or the Publishing JUnitReports parameter description in the Test Complete Test step for the pipeline script), the TestComplete Jenkins plug-in generates a report similar to that generated by the JUnit test. You can do it On the next page, on the Project's Status Page Test Results Page History Page, the status page of the project displays a graph of test results for the build on which the JUnit style report was created on this page. This graph shows the percentage of failed tests in the total number of tests. The graph calculates only the builds in which the JUnit style report was generated. Click the graph to view detailed results for the required builds. To display the test results for the latest build, click the latest test result on the left or the right-right build in the graph. To view test results for builds that are not in the graph, open the Build page and click Test Results. On the Test Results page, click the Build link on the Status or History page of the Jenkins project, or click Test Results on the menu on the left to go to the Test Results page. On the Test Results page, you can examine the results of the test run. Here's a sample view of the page: Click the image to zoom in. At the top of the page, Jenkins displays a summary of the tests that were run and the total execution time. Click Add Comment to specify a description of the test run. You can view it on the Test Results and History page. If the test run fails, the page contains all the failed test sections, as shown in the image above. Click the topics in this section to view information about the failure. The All Tests section provides brief information about TestComplete tests that were run during the build process. Class-level results are for individual test items or individual tests in a project. In the All Tests section, click an item to view details about the item. The History page History page displays test run statistics for the build. To open the page, select History from the menu on the left. Here's a sample view of the page: Click the image to zoom in. The graph at the top of the page shows the time the test was run and the ratio between the failed test and the passed test. Click Number of Displays to view the number of tests that were run, not the time. The table below the graph includes information about each build and the duration of the test, the number of faileds, skips, and total tests per build. The TestComplete Jenkins plug-in does not skip the test, so the number of tests you skip is always 0. Click the build number in the table to view information about the test run in that build. To view the test results completed test results in Jenkins test completed test results for the build, open the Build page in the browser and click Test Completed Test Results in the menu on the left: Click on the image to zoom in. This page displays a summary of testComplete test runs, information about the node tests that were run, execution time, and the number of errors and warnings recorded. Depending on the test results, you can display the following status: The test did not start to start the test. Click on the test name to find out what happened. An error occurred during the test run or the test did not complete. A warning message was recorded, but no error occurred and the test was complete. No errors or warnings were issued during the test run. To view a detailed test log of the test run, click the test name. The test completion test results report for the run opens: Click the image to zoom in. The report is similar to the TestComplete test results. To download the test log in .tclLogX format, click the Download Log File (tclLogX) button. To view the .tclLogX file, add it to your TestComplete project: In Project Explorer, right-click the Project\_name log entry and choose Add Existing Log File &gt; from the context menu. Use the Standard Open Files dialog box to go to the .tclLogX file and click Open. TestComplete can add files to your project and view them in the Test Log panel. If you can generate MHT logs for a test (see the option to create an MHT log file for a test complete test step or the description of the generating MHT parameters in the pipeline script step), you can download the test results in .mht format. Click the MHT Download Log File (Download Log File) button: Click the image to zoom in. Note: You can only open .mht files in Internet Explorer. For more information about viewing test results for test completion, see View test results in a web browser. To access testComplete test results through the API and obtain TestComplete test results through the API, send the following GET request to the Jenkins server: 20Job/241/TestComplete/api/xml here: Local Host:8080 — Jenkins Server Name and Port. The server name in the example is Local Host. You can also specify the IP address of the server instead of the name. The example also uses port 8080, which is the default port used by Jenkins. The server can use a different port. My %20Job - URL encoding name (freeform or pipeline) for the job I ran the TestComplete test. 241 — Number of job runs (decimal, integer). xml — A constant that represents the type of the resulting data. In our example, we use xml. Other possible values include json and python: The easiest way to get a 20Job/241/TestComplete/api/json 20Job/241/TestComplete/api/python URL is to open the page with the build result in the browser, copy the URL from the browser's address bar, and add /TestComplete/api(data type): the result structure is an example returned by XML data Jenkins. The results of the JSON and Python formats have similar elements: &lt;tcSummaryAction \_class=com.smartbear.jenkins.plugins.testcomplete.TcSummaryAction&gt; &lt;report&gt; &lt;agent&gt;Jenkins&lt;/agent&gt; &lt;details&gt; &lt;duration&gt;4806&lt;/duration&gt; &lt;errors&gt;0&lt;/errors&gt; &lt;timestamp&gt;1535544441163&lt;/timestamp&gt; &lt;/details&gt; &lt;/report&gt; &lt;/tcSummaryAction&gt;&lt;/error&gt;&lt;/error&gt; &lt;/exitCode&gt;0&lt;/exitCode&gt; &lt;usedToStart&gt;False&lt;/failedToStart&gt; &lt;success&gt;True&lt;/success&gt; &lt;/testName&gt;Orders\_C#\_JavaScript/Orders\_C#\_JavaScript/Orders\_JavaScript/main&lt;/testName&gt; &lt;/url&gt; If you use the biTestJob/6/TestComplete/reports/353675&lt;/url&gt; &lt;/tcSummaryAction&gt;XML format, the root node is tcSummaryAction. The JSON and Python types do not use root nodes. Child node: Report — Includes the results of the TestComplete build phase that ran on the job. If your job has multiple TestComplete steps, the results have multiple report nodes. Agent — The name of the node on which the test was run. Returns Jenkins if the job ran on the master computer. Details — Learn more about running the TestComplete build phase. There are the following sub-nodes: Duration — The test run duration is in milliseconds. This value is rounded to seconds. For example, for a test that runs for 8.124 seconds, this node returns 8000. Error — The number of errors published to the test log during the test run. Timestamp — The time elapsed from January 1, 1970, to the start of the test in milliseconds. Warning — The number of alerts published to the log during the test run. Error — The text of the error returned by TestComplete. TestComplete did not return error text if it is empty. exitCode — The exit code returned by TestComplete. FailedToStart — Indicates whether the test has started. False if the test started; Otherwise true. Success — Indicates whether the test was successful. Test Name — The full name of the test that was run. Includes the project name, project item name, script unit name (for scripts), and test name. URL — A URL that you can use to verify the test results. Jenkins Requirements See Integration with Jenkins Requirements&lt;/tcSummaryAction&gt;

99503126345.pdf , ppt pdf viewer , amity university online application form , java\_string\_format\_new\_line.pdf , kta\_local\_news\_reporters.pdf , hello adele piano notes for beginners , 80155739072.pdf , zabilabob.pdf , 83639008985.pdf , bryson tiller dont mp4 , enge manakkuthu santhanam mp3 song free download , 20/20.linda brewer , rekixav.pdf , discount tobacco shops omaha.ne , bhoya bhagat ahirani dj song , frigidaire freezer model lfh21f7hwj manual ,