

Add reporting to Jenkins

This section guides you through adding Perfecto Smart Reporting to Jenkins.

Add a link to your report from within Jenkins

You can add a dynamically generated link to the report for a given Jenkins job by using the [Groovy Postbuild Plugin](#).

The script should include the following code:

```
jobName = manager.build.getProject().getName()
buildNumber = manager.build.getNumber()

summary = manager.createSummary("graph.gif")
summary.appendText("<a href=\"https://<CQL_NAME>.reporting.perfectomobile.com/library?jobName[0]=${jobName}&jobNumber[0]=${buildNumber}\">Perfecto Test Report</a>", false)
```

See the details of the [reporting server](#).

Group tests by job and build in the Report Library

To view tests grouped by continuous integration system job and build in the Report Library, you need to provide the job name and build number into reporting as system variables. This enables you to group your test executions in reporting first by job name and then by build number, providing you a complete overview of the build quality.

On this page:

- [Add a link to your report from within Jenkins](#)
- [Group tests by job and build in the Report Library](#)
- [Supply Maven or Ivy parameters](#)
- [Set up Slack notifications](#)

The screenshot shows the Perfecto Report Library interface. The main area displays a summary for the group 'AppiumTests-Android-branch / 11' for the time frame 'Jun 15 - 19'. The summary shows a total success rate with a donut chart indicating 310 tests, 39 failed, and 271 passed. Below the summary is a table with columns for Report Name, Status, Platform (Device, Browser, OS, Resolution), and Time. The table lists several test runs on Google Nexus 9 devices. The right-hand sidebar contains filters for Time Frame (Custom, 15/06/16 to 19/06/16), Group By (Job), Then by (Job number), and Filter (Failed, Passed, Unknown).

Supply Maven or Ivy parameters

If you are running your tests using Maven or Ivy, add the following JVM parameters and (optionally) the Report-tags parameter:

```
-Dreportium-job-name=${JOB_NAME} -Dreportium-job-number=${BUILD_NUMBER} -Dreportium-job-branch=${GIT_BRANCH}
-Dreportium-tags=${myTag}
```

You need to supply the **myTag** value. Jenkins automatically supplies the **JOB_NAME**, **BUILD_NUMBER**, and **GIT_BRANCH**.

To add CustomField pairs, use the following format:

```
-DReportiumCustomFields=customField1=value1,customField2=value2
```

Build	
Maven Version	Maven-3.3.3
Root POM	E2E-Test/pom.xml
Goals and options	clean test -Dreportium-job-name=\${JOB_NAME} -Dreportium-job-number=\${BUILD_NUMBER}

This video shows how to add the Jenkins Job Name and Build Number parameters.

Set up Slack notifications

Using the [Slack plugin for Jenkins](#), you can configure Jenkins to send you a notification when the tests complete and include the link to the report for easy access.

To do this:

1. Go to the **Post-build Actions** section of the project configuration in Jenkins.
2. In the **Slack Notifications** sub-section, do the following:
 - a. Find and select the **Include Custom Message** checkbox.
Note: You may need to click **Advanced** to access this field.
 - b. In the **Custom Message** field, add the following:

View your Perfecto report at: [https://<CQL_NAME>.reporting.perfectomobile.com/library?jobName\[0\]=\\${JOB_NAME}&jobNumber\[0\]=\\${BUILD_NUMBER}&tags\[0\]=\\${myTag}](https://<CQL_NAME>.reporting.perfectomobile.com/library?jobName[0]=${JOB_NAME}&jobNumber[0]=${BUILD_NUMBER}&tags[0]=${myTag})

where:

<CQL_NAME> is the prefix of your Perfecto CQ Lab. For example, if you access the Perfecto Lab at <https://demo.perfectomobile.com>, then **<CQL_NAME>=demo**.



This video shows how to add Jenkins notifications for reporting to your Slack channel.