

Report Library

The Report Library presents a detailed overview of the tests run within a selected period of time, including:

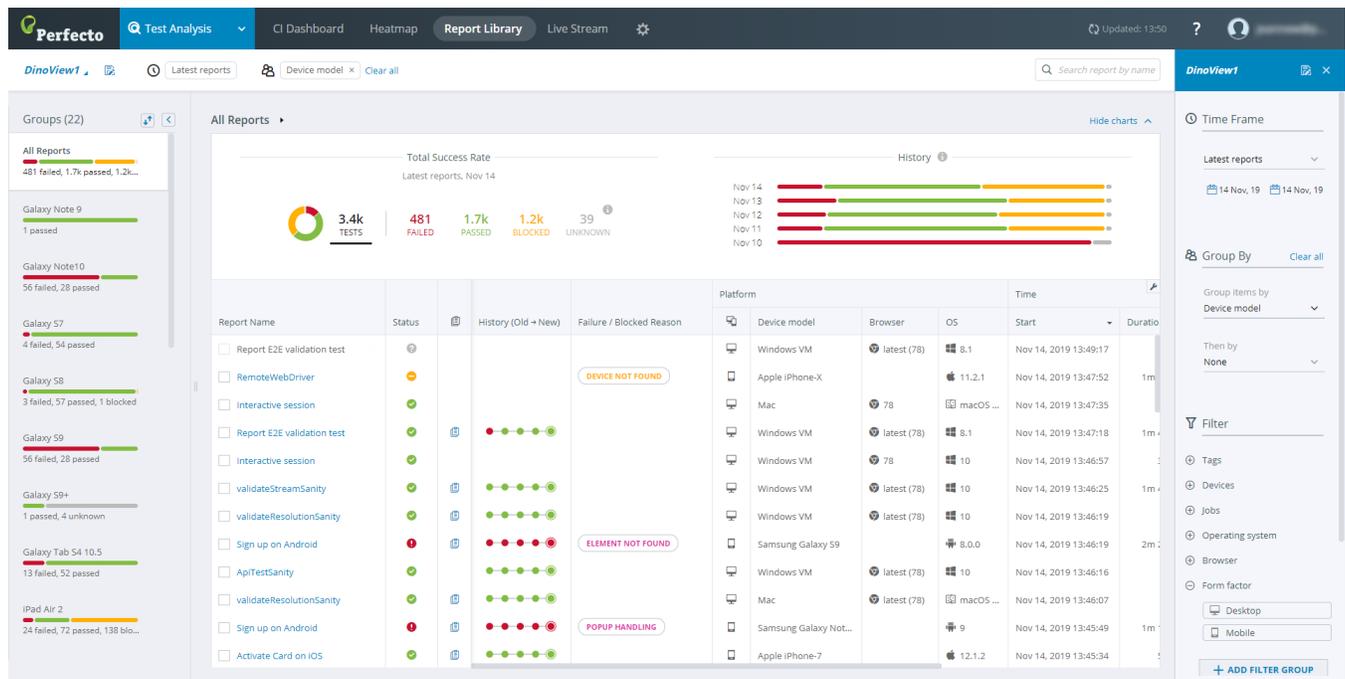
- Statistical information for all listed tests
- Historical breakdown of when tests failed
- Table of all selected tests, including information on the parameters of the test run

For information on how to filter and group the view, see [Focus on what's important to you](#).

This figure shows the **Report Library** grouped by device model.

On this page:

- [Overview](#)
- [Focus on what's important to you](#)
 - [Search reports by name](#)
 - [Filter by status](#)
 - [Configure columns](#)
 - [Filter or group by custom fields](#)
 - [Filter by tags](#)



Overview

The following sections explain how to read and use the different parts of the **Report Library** view.

Header bar

The header bar across the top of the display (just below the top-level **Test Analysis** tabs) displays:

- **Saved view selector** - If you have any pre-configured views, select to view the current data based on the configured selections.
- **Active filters** - Lists all of the currently selected filters and groupings for the list.
- **Search field** - Lets you search for keywords to display only tests with names that match the entered term.

Statistic charts

The section just below the header bar includes statistical views of the listed tests:

- **Statistical Overview** shows general statistics of the currently listed tests, split into the number of passed, failed, blocked, and other tests.
- **Run History** provides an overview of dates when tests were run. The view is color-coded to show the number of passed, failed, blocked, and other tests for a particular date.

Use the **Hide charts/Show charts** link above the charts to toggle the display.

Test details table

The table in the **Report Library** view lists the following information for all test reports.

Column	Sub-column	Description	Displayed by default
Report Name		The script name as supplied in the PerfectoReportiumClient <code>testStart()</code> method. The column includes a checkbox that you can use to select test reports for the Cross Report view .	Yes
Status		The final status reported by the specific test run. The test status is one of: Passed , Failed , Blocked , Unknown .	Yes
Cross Report		Indicates if the test report is part of a cross-report .	Yes
History		Indicates when this test run was executed relative to other runs of the same test. (See below for more detail.)	Yes
Failure /Blocked Reason		Indicates a reason detected, either automatically by Smart Reporting heuristics or by the testers (see Implement the reporting framework), why the test failed. For more information, see below .	Yes
Platform		Indicates if tests were executed on Mobile or Web devices. May include an indication of the number of devices executed on.	Yes
	Form factor	Icon that identifies if the testing device was a <i>Mobile</i> or <i>Desktop Web</i> device.	Yes
	Device model	List of devices used in the test, with each device separated by comma.	Yes
	Device ID	For mobile devices, provides the ID number of the testing device.	No
	Browser	Browser version for Web devices.	Yes

	OS	List of operating system versions, coordinated with devices list.	Yes
	Resolution	List of device resolutions, coordinated with devices list.	No
Job		Details of the CI Job reported for this test.	No
	Name	Job name as reported in the execution context.	No
	#Number	Job number as reported in the execution context.	No
	Branch	Job branch as reported in the execution context.	No
Time		Provides details of when test was run and its duration.	Yes
	Start	Start time of the test run.	Yes
	Duration	Duration time of the test.	Yes
Tags		Indicates the number of tags associated with the test run.	Yes
Lab		Indicates if test was run in a Perfecto CQ Lab.	No
Automation Framework		indicates the automation framework supported by the device used by automation script.	No

Test history information

The history graph in the **History** column shows a series of nodes. Each node represents a single run of the test.

The History mechanism defines test similarity by test name and execution capabilities. If there is a difference in either the name or the capabilities (for example, the `osVersion` capability is included in one test run but not in another), the tests are considered 'not similar'. As a result, they are not connected in history.

The test run whose details are described in this report is displayed as a double-ring in the history (). This makes it easier to identify when this test run was executed relative to other test runs. When reading the history graph, keep in mind that:

- The latest test run is always represented by the right-most node.
- No more than five nodes appear in the history graph. If the specific run occurred prior to the five latest runs, the graph shows a break represented by three dots ().
- The color of the node represents the test result status for that particular run, where green means 'passed', yellow means 'blocked', and red means 'failed' ().
- Move the pointer over a node to display a tooltip with details for that run.
- Arrows around an icon () identify test runs with scheduled retries that have been collapsed into a single test report.

The Scheduled Retries is turned off by default. To turn it on in your cloud instance, contact [Perfecto Support](#).

For a test to be considered a retry, it must share the same parameters and CI job name and number or be part of the same execution. Perfecto does not list a test that is considered a retry in the table and does not take it into account when calculating statistics. Only the last test in a retry series makes it into the statistics. For more information, see [STR](#).

Failure/blocked reason

Smart Reporting is designed to provide the test data from your test runs, but also to allow you - the tester or the test manager - to better understand the results of your tests. If your test ends with a success status, then you know that all is well. However, when a test fails, Smart Reporting may analyze the test data and provide a *failure reason* that indicates what caused the test to fail.

The Smart Reporting system provides functionality that supports generating this failure reason classification, either:

- **Manually by the test script, based on the stage in the script execution when the test is determined to have failed.**
- **Automatically. Smart Reporting analyzes the entire test information and generates the reason based on a heuristic classification.**

In either case, the failure reason is color-coded (where green means 'passed', yellow means 'blocked', and red means 'failed') and displayed in the table of the test reports, in the **Failure/Blocked Reason** column, to allow a quick overview of the different failed tests. If a test fails without reporting a failure reason, the **Status** column will show a red icon, but the **Failure/Blocked Reason** column will be blank.

The Smart Reporting system automatically identifies instances when the test failed before the test was able to start. When these failures occur, the report is marked with a **blocked** failure reason (that will appear in the table as the Failure Reason, color of reason in yellow) and the **Status** column will show a **Blocked** status (🚫). These blocked failure reasons are completely controlled (failure reason text and color) by the Smart Reporting system. Some examples of blocked failure reasons include:

- Device in use
- Device not connected
- Device not found

Tags column

You can move the pointer over the tag icon in the **Tags** column for a specific test to display a tooltip that lists all tags associated with the test.



This tooltip is an **active tooltip**. Clicking a tag filters the table to show only tests associated with that tag.

Your browser does not support the HTML5 video element

Sidebar

Use the sidebar on the right to:

- Filter the list of displayed test runs, to limit the list to those that comply with the selected parameters or time frame.
- Create groups of test runs to display in the groups overview.

For more information on the sidebar, read [here](#).

Focus on what's important to you

The following sections explain how you can group and filter information in the **Report Library** to narrow your focus on the details you need to see.

Search reports by name

When the **Report Library** includes a wide variety of tests, you can use the **Search** field at the top to focus on the set of tests you are interested in.

To search reports:

1. Start typing the name (or any sub-string of the name). Smart Reporting presents a list of suggested available test names.
2. Do one of the following:
 - Select the test name from the list of suggestions.
 - Complete typing the requested name and press enter.

The list refreshes to display only tests with names that match the search term. If no tests match the search term, a 'No reports found' message appears.

Filter by status

You can use the **Statistical Overview** area to focus on reports with a selected final status:

- Click a counter (**Failed**, **Passed**, **Blocked**, **Unknown**) to filter the list to show only those reports with the selected status.
- Click **Tests** to list all test reports.

For example, if 26 tests have a status of **Blocked**, clicking the **Blocked** counter filters the list to display only the 26 reports that are currently blocked.

Configure columns

The table may show information that is not relevant at the moment. You can configure it to hide or show columns as needed.

To configure columns:

1. Click the pliers icon  above the table to display a menu of columns you can configure to appear in the Report Library table.

Custom fields defined in a test appear as column names at the end of the list.

2. Select the box next to columns you want to include in the table or clear check boxes of columns you don't want to display.

The **Report Name** and **Status** columns are *not* configurable. They are always selected.

Filter or group by custom fields

Use the custom fields defined at the different context classes or as part of the JVM command line parameters (with CI) to refine your results beyond built-in filter categories and tags. Custom fields are <name, value> pairs, so they allow for an additional level of segregation. You can filter and group by custom fields. In addition, you can add a custom field as a column to the report table (see [Configure columns](#)).

To filter by custom fields:

1. In the sidebar, under **Filter**, click **ADD FILTER GROUP** and select a value from the list. You may need to scroll down to find the field you are looking for.
The custom field is added as a filter category.
2. From the list that opens below the custom field, select a value and whether you want to include or exclude this value.
The list of reports refreshes to show the updated filter settings.

To group by custom fields:

- In the sidebar, under **Group By**, select the required custom field from the list.

Filter by tags

Best Practice

For greater flexibility and more refined filter and group results, we recommend the use of [custom fields](#).

Use the tags defined by the different context classes to better focus the analysis of the test results by filtering the tests to the subset of interest.

Note: Perfecto tracks up to 3000 tags. If the number of tags in your system exceeds 3000, Perfecto may not find a tag if you try to filter by it.

See [Tag-driven reports \(RTDD workflow\)](#) on how to optimize your test tags for efficient test analysis.

To filter by tags, do one of the following:

- In the **Tag** column, click the tag icon. In the tooltip that displays, click the tag you want to filter by.
- In the sidebar, under **Filter**, expand the **Tags** section and select one or more relevant tags.

The following figure shows the table filtered by the tags **iPhone 7** and **XCTest**. Note the indications of the filtering in the right pane.

The screenshot displays the Perfecto Test Analysis interface. At the top, there's a navigation bar with 'Perfecto' logo, 'Test Analysis' dropdown, and various dashboard options like 'CI Dashboard', 'Heatmap', 'Report Library', and 'Live Stream'. Below this, a breadcrumb trail shows 'DinoView1' > 'Latest reports' > 'Java x iPhone 7 x Clear all'. A search bar is present on the right. The main content area features a 'Total Success Rate' section with a circular progress indicator showing 779 TESTS, 7 FAILED, and 772 PASSED. To the right is a 'History' chart showing performance trends from Nov 09 to Nov 14. The central part of the dashboard is a table of test results, filtered by 'iPhone 7' and 'XCTest'. The table has columns for Report Name, Status, History (Old + New), Failure / Blocked Reason, Platform, Device model, Browser, OS, Time (Start, Duration), Tags, and Lab. The right sidebar contains 'Time Frame' (Latest reports), 'Group By' (None), and 'Filter' (Clear all) sections. Under 'Filter', 'Tags' are expanded, showing 'iPhone 7' and 'Java' as selected filters.

Report Name	Status	History (Old + New)	Failure / Blocked Reason	Platform	Device model	Browser	OS	Time	Duration	Tags	Lab
validateStreamSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:57:38	1m 35s 395ms	6	PERFECTO	
validateResolutionSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:57:32	6s 215ms	6	PERFECTO	
ApiTestSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:57:29	2s 675ms	6	PERFECTO	
validateStreamSanity	✗	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:52:52	1m 44s 154ms	6	PERFECTO	
validateResolutionSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:52:45	6s 289ms	6	PERFECTO	
ApiTestSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:52:43	2s 721ms	6	PERFECTO	
validateResolutionSanity	✓	●●●●●●●●		Mac	latest (78)	macOS ...	Nov 14, 2019 18:50:55	3s 817ms	6	PERFECTO	
validateStreamSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:48:12	2m 1s 348ms	6	PERFECTO	
validateResolutionSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:48:05	7s 565ms	6	PERFECTO	
ApiTestSanity	✓	●●●●●●●●		Windows VM	latest (78)	10	Nov 14, 2019 18:47:58	6s 290ms	6	PERFECTO	
validateResolutionSanity	✓	●●●●●●●●		Mac	latest (78)	macOS ...	Nov 14, 2019 18:47:52	3s 777ms	6	PERFECTO	
Activate Card on iOS	✗	●●●●●●●●		Apple iPhone-7	12.1.2		Nov 14, 2019 18:45:31	41s 949ms	4	PERFECTO	
Login to App on iOS	✓	●●●●●●●●		Apple iPhone-7	12.1.2		Nov 14, 2019 18:45:03	28s 678ms	4	PERFECTO	