Implement the reporting framework

To take advantage of the reporting framework in your automation scripts, perform the following steps:

1. Download the Reporting SDK.
2. Create an instance of the `ReportiumClient` object.
   The `ReportiumClient` monitors the application and transfers the report information and artifacts to the Reporting storage server.
3. Create a `PerfectoExecutionContext` instance.
   The execution context defines metadata for the Execution report. The metadata includes "tags" that can be used to select the execution report from the list of reports.
   A single Execution report may include multiple test reports.
4. Start the "test" by notifying the Reporting Client that the test unit for the report is starting.
   This indicates the point of a single test. Use the `testStart()` method of the `ReportingClient` class.
5. Define the *logical steps* of the test.
   Each logical step may include a number of actual command steps in the automation script but delineate a logical sequence of command steps.
   The report will show the collection of the logical steps that can be drilled down to check the component command steps.
6. Stop the "test" by notifying the Reporting client that the test unit has completed, providing a status for the test completion.
7. Gain access to the final report.
   The Reporting client can supply the URL to the report for retrieval.
   The actual report is not readily available because it has to be compiled from the different component artifacts.
8. In addition, the Reporting client supplies an API that supports exporting the execution report or individual test reports in different formats.

The following video shows an overview of how to integrate the Reporting SDK into your test script:

The following pages provide details on how to implement reporting for your programming language and framework:

- C#
- Java
- JavaScript
- Python
- Ruby