QUnit

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

QUnit is a unit testing framework for JavaScript programming language. QUnit has been important in the field of test-driven development, and is used by jQuery, jQuery UI, and jQuery Mobile projects. QUnit is capable of testing any generic JavaScript codebase.

This tutorial explains the fundamental concepts of QUnit and how to use QUnit in day-to-day life of any unit testing project while working with JavaScript.

Audience

This tutorial has been prepared for the beginners to help them understand the basic functionality of QUnit tool. After completing this tutorial, you will find yourself at a moderate level of expertise in using QUnit testing framework from where you can take yourself to the next level.

Prerequisites

We assume you are going to use QUnit to handle all levels of JavaScript projects development. Hence, it will be good if you have knowledge of website development using any Scripting language, especially JavaScript Scripting and website testing process.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. QUnit – Overview

Testing is the process of checking the functionality of the application whether it is working as per the requirements and to ensure that at the developer level, unit testing comes into picture. Unit testing is the testing of a single entity (class or method). Unit testing is very essential for every software organization to offer quality products to their clients.

Unit testing can be done in two ways as mentioned in the following table.

Manual Testing	Automated Testing
Executing the test cases manually without any tool support is known as manual testing.	Taking tool support and executing the test cases using automation tool is known as automation testing.
Time consuming and tedious. Since the test cases are executed by human resources, it is very slow and tedious.	Fast Automation. Runs test cases significantly faster than human resources.
Huge investment in human resources. As test cases need to be executed manually, more number of testers are required.	Less investment in human resources. Test cases are executed using automation tool hence, less number of testers are required.
Less reliable, as tests may not be performed with precision each time due to human errors.	More reliable. Automation tests perform precisely the same operation each time they are run.
Non-programmable. No programming can be done to write sophisticated tests, which fetch hidden information.	Programmable. Testers can program sophisticated tests to bring out hidden information.

What is QUnit ?

QUnit is a unit testing framework for JavaScript programming language. It is important in the test driven development, and is used by jQuery, jQuery UI, and jQuery Mobile projects. QUnit is capable of testing any generic JavaScript codebase.

QUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code, which can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little..." which increases the programmer’s productivity and the stability of program code reducing the programmer’s stress and the time spent on debugging.

Features of QUnit

QUnit is an open source framework used for writing and running tests. Following are its most prominent features:

- QUnit provides Assertions for testing expected results.
- QUnit provides Test fixtures for running tests.
- QUnit tests allow to write code faster, which increases the quality.
- QUnit is elegantly simple. It is less complex and takes less time.
- QUnit tests can be run automatically and they check their own results and provide immediate feedback. There's no need to manually comb through a report of test results.
- QUnit tests can be organized into test suites containing test cases and even other test suites.
- QUnit shows test progress in a bar that is green if the test is going fine, and it turns red when a test fails.

What is a Unit Test Case?

A Unit Test Case is a part of code which ensures that another part of the code (method) works as expected. To achieve the desired results quickly, test framework is required. QUnit is a perfect unit test framework for JavaScript programming language.

A formal written unit test case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition.

There must be at least two unit test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases as positive and negative.

2. QUnit – Environment Setup

There are two ways to use QUnit.

- **Local Installation** – You can download QUnit library on your local machine and include it in your HTML code.

- **CDN Based Version** – You can include QUnit library into your HTML code directly from Content Delivery Network (CDN).

Local Installation

- Go to the https://code.jquery.com/qunit/ to download the latest version available.

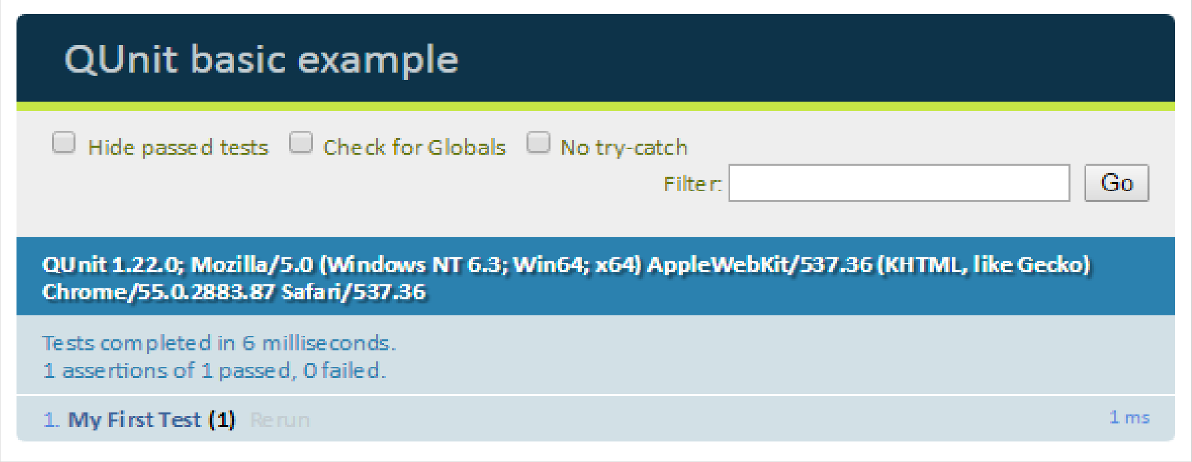
- Place the downloaded **qunit-git.js** and **qunit-git.css** file in a directory of your website, e.g. /jquery.

Example

You can include **qunit-git.js** and **qunit-git.css** files in your HTML file as follows –

```
<html>
   <head>
      <meta charset="utf-8">
      <title>QUnit basic example</title>
      <link rel="stylesheet" href="/jquery/qunit-git.css">
      <script src="/jquery/qunit-git.js"></script>
   </head>
   <body>
      <div id="qunit"></div>
      <div id="qunit-fixture"></div>
      <script>
         QUnit.test( "My First Test", function( assert ) {
            var value = "1";
            assert.equal( value, "1", "Value should be 1" );
         });
      </script>
   </body>
</html>
```

This will produce the following result -



QUnit basic example

QUnit 1.22.0; Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36

Tests completed in 6 milliseconds.
1 assertions of 1 passed, 0 failed.

1. My First Test (1) Rerun 1 ms

CDN Based Version

You can include QUnit library into your HTML code directly from Content Delivery Network (CDN).

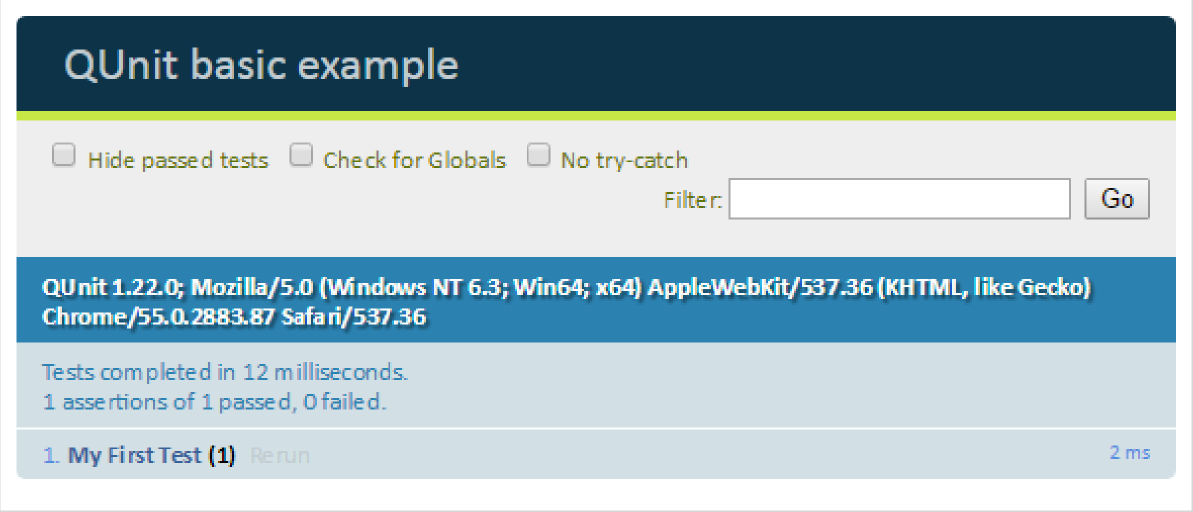
We are using jQuery CDN version of the library throughout this tutorial.

Example

Let us rewrite the above example using QUnit library from jQuery CDN.

```
<html>
   <head>
      <meta charset="utf-8">
      <title>QUnit basic example</title>
      <link rel="stylesheet" href="https://code.jquery.com/qunit/qunit-1.22.0.css">
      <script src="https://code.jquery.com/qunit/qunit-1.22.0.js"></script>
   </head>
   <body>
      <div id="qunit"></div>
      <div id="qunit-fixture"></div>
      <script>
         QUnit.test( "My First Test", function( assert ) {
            var value = "1";
            assert.equal( value, "1", "Value should be 1" );
         });
      </script>
   </body>
</html>
```

This will produce following result –



QUnit basic example

QUnit 1.22.0; Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36

Tests completed in 12 milliseconds.
1 assertions of 1 passed, 0 failed.

1. My First Test (1) Rerun 2 ms