

Let's Get Started with MFTF

Introduced by Tom Erskine at Merge Test Fest, the 'Magento Functional Testing Framework'(MFTF) is an open source solution for creating functional tests for Magento applications. This cross-platform solution aims at replacing traditional functional testing framework and reducing mistakes due to complex functional flows.

MFTF is a series of readymade tests to check the functionality of Magento 2 systems. In some scenarios, auto tests are run to cover extended functionalities. These auto tests are easy to code and implement, even by manual testers in a [Magento development company](#).

The Features of MFTF that Make It Desirable

As MFTF is based on technologies like Selenium, Allure, and Codeception, it caters a wide variety of testing scenarios and cases. The following features of MFTF make it a popular trend over other primitive technologies.

Guarantees security and stability

It's built completely by Magento and for Magento and provides great customer support services

Infrastructure scalability lets MFTF run different tests in different environment

Provides flexibility with modularity support

Easy to customize existing tests

A reporting/logging tool makes it more informative

Highly compatible

Setting up Environment for MFTF

1. Upgrade your PHP Version: First of all, you need to upgrade your PHP version according to the Magento instance that you prefer to use.
2. Install the Composer
3. Install Java to run Selenium Server

4. Use Selenium Server on your System: The Selenium jar file will help you to run MFTF test cases.

5. Get a Web Browser Driver: A web browser driver is required to run the test cases. You need to install the correct one according to your choice of browser.

Setting up MFTF Framework

The following steps will guide you through MFTF setup process.

Step 1- Install Magento 2.3

The first and foremost step is to install Magento 2.3. For this you can either follow this link

<https://devdocs.magento.com/guides/v2.3/install-gde/install-quick-ref.html>

or download from a trusted source and extract to your local host directory. Next, go to the Magento directory on your web browser and complete the setup.

Step 2 – Install the Dependencies

Use the following command to install MFTF

```
composer install -d dev/tests/acceptance/
```

Step 3 –Build Project

Run the following command in the Magento Project Root

```
vendor/bin/mftfbuild:project
```

Step 4 – Edit Environment Settings

Edit the .env file in [Magento_Root]/dev/tests/acceptance/ directory, with

```
vim dev/tests/acceptance/.env
```

Furthermore, set the following parameters required for tests –

MAGENTO_BASE_URL: should contain a domain name of the Magento instance that will be tested.

MAGENTO_BACKEND_NAME: must contain the relative path for the Admin area.

MAGENTO_ADMIN_USERNAME: should contain the username required for authorization in the Admin area.

MAGENTO_ADMIN_PASSWORD: must contain the user password required for authorization in the Admin area.

Step 5 – Enable the Magento CLI commands

Run the following command in [Magento_Root]/dev/tests/acceptance directory

```
cp dev/tests/acceptance/.htaccess.sample dev/tests/acceptance/.htaccess
```

Step 6 – Create and Run the Tests

In order to run the tests, you need to have a running Selenium server and 'MFTF' commands

```
Rcd /Java -Dwebdriver.chrome.driver=chromedriver -jar selenium-server-standalone-3.14.0.jar
```

Generate the tests

Run the following command to run complete test cases

```
vendor/bin/codecept run functional
```

You may also run a specified group (vendor/bin/mftfrun:group product)by using vendor/bin/mftf command.

Next, run all the Magento tests

In the dev/test/acceptance, run the following command to run all test cases

```
vendor/bin/mftf run functional
```

Step 7 – Generate Reports

MFTF generates test reports in CLI. For visual representation of reports, you can also use Allure Framework.

Install Allure

Install 'npm'using the following command

```
sudo apt install npm
```

```
npm install -g allure-commandline --save-dev
```

Next, run Allure to check the final report

To check the test result report in Allure, use the following command—

```
allure serve dev/tests/_output/allure-results/
```

Conclusion

Finally, MFTF facilitates the testing of user interactions with web applications. It helps testers with easy automation of regression testing. The above-mentioned steps will help you start MFTF from scratch and generate test reports with ease.