Testing

To be the robust and reliable information system that health care systems demand, we need to be overly vigilant about testing the OpenMRS code. This section contains all testing information including unit tests, continuous integration testing, release testing, and user testing.

Developers: Please Contribute Tests

We are always looking for people that want to add to our testing arsenal.

If you are new to OpenMRS please read Getting Started as a Developer first to get your accounts and development environment set up before writing any code.

You can look through the existing issues that have a status of READY FOR WORK and filter for title containing "test" or a label like "test", "testing" as shown here

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
TRUNK-5068	Add tests to ProgramWorkflowService	🟧	2017-02-20	2019-03-13		Unassigned	Ivo Ulrich	↑	READY FOR WORK	Unresolved
TRUNK-1656	Create a Testing Feedback Module	➕	2010-08-04	2014-09-14		Unassigned	Jeremy Keiper	⌃⌃	READY FOR WORK	Unresolved
TRUNK-4023	1.9.x tests fail on 32-bit JDK	🟧	2013-07-18	2013-07-31		Unassigned	Terry Child	⌄⌄	READY FOR WORK	Unresolved

3 issues

If you cannot find an issue related to testing, don't worry, there is a lot to do 😋

In that case either go to coveralls.io/github/openmrs/openmrs-core and look for classes that have either no test at all (0% coverage) or classes that have low coverage (coverage percentages are a topic of debate, how much is enough, but I would say below 80% isnt good).

```
mvn clean package
mvn jacoco:report
```

The coverage reports are now in

- openmrs-core/api/target/site/jacoco/index.html
- openmrs-core/web/target/site/jacoco/index.html

How To Write Unit Tests

- How to write unit tests
- How to write unit tests for modules
- How to write unit tests for the web layer pages
- Unit Testing Conventions

Continuous Integration

We have a Bamboo server set up to run our unit tests after every commit to key branches. On our GitHub repository of OpenMRS core we also use Travis CI which runs your changes to the code and tests on every pull request anyone makes. We also generate test coverage and use coveralls.io to check pull requests to make sure the changes anyone adds are well tested. These checks also run on module repositories within the OpenMRS GitHub organization but it might be that not every module makes use of all the checks.

Visit the Junit section of http://ci.openmrs.org to see the latest results of the unit tests in the 1.4.x/1.5.x/..x and trunk branches.

Everyone: Testing Releases

Application level release testing is done with selenium and jbehave. The "release-test" maven module folder contains those tests.

The Testing Releases page describes the recipe(s) used to perform high application level testing of the system prior to releases. OpenMRS relies heavily on volunteerism to help test releases with real world data and situations. Join the implementers mailing list and watch for requests for help in testing alpha, beta, and release candidates.

Usability Testing

The Design section of the wiki contains information about our current efforts.

Load Testing

To check how a server or the OpenMRS application responds to load, you can check out this page.

Quality Assurance Team

A dedicated QA Team have been formed for OpenMRS. To learn more about their work or join them see this page.

Other Resources

- http://www.vogella.de/articles/JUnit/article.html