

OpenHIE Sandbox



the Sandbox project was formed to create an instance of OpenHIE based on the work done in Rwanda on the [RHEA Project](#). While this served us well in the early days we now are looking to become more of a support community based on the needs of real-world implementers. So this page will become archived.

OpenHIE Sandbox Project

What is the OpenHIE sandbox?

The OpenHIE sandbox is a set of environments where all stages of development and testing can be seen. Originally the sandbox was just a staging environment for internal testing and has expanded to 2 other environments as needed. Over time we would like to see our release process more automated.

Goals

- Recreate OpenHIE environment in Rwanda.
- Create a Web Demo for public use.
- Implement Continuous Integration for OpenHIE component interoperability
 - Nightly Builds
 - Automated deployments
 - Automated tests
 - Statistics on code quality and coverage

Environments



Unknown macro: 'gliffy'

Development (dev)

The dev environment of the sandbox is where automated builds are deployed. The builds come from our CI server and are automatically deployed nightly.

Testing (test)

The OpenHIE sandbox is the place to find all of the newest developments in the community. This will serve as a place to experiment with new builds of software, new configurations and data. This is not stable and could change at any time.

Demo (prod - hosted)

The OpenHIE demo site is a place to try out what the community has deemed as stable. It is preloaded with data to demonstrate the basic features of the software recommended by each community. This will be reset each night to clear any changes made to the machines.

Downloadable (prod - self hosted)

The downloadable release is a copy of the demo site made available for use on your local machine. You will need virtualbox to run the images and meet the minimum hardware requirements to run all of the virtual machines at once for full functionality. With this you may make configuration changes suited to your environment and experiment with the software on your local machine.

Maturity model of code



Unknown macro: 'gliffy'

Steps

1. Code is committed to a community repo.
2. CI checks code repository for latest changes.
3. Code is compiled / packaged and made into an artifact. Unit tests are also run if available.

- Artifact is deployed to our dev environment. Manual testing may occur, for functionality and interoperability. Test interoperability against staging (which should mirror prod).
- Release artifact is deployed. Bamboo build agent runs integration tests. Users will do User Acceptance Testing to determine if all expected functionality works as expected.
- If all tests pass, make a Release of OpenHIE, which specifies OpenHIE (e.g. v0.2.1 = OpenHIM 2.3, OpenEMPI 2.9.6, ...etc). Deploy using automated scripts to demo environment.

Status

Testing Environment

OHIE Component	GUI URL
Point of Care	http://poc.test.ohie.org:8080/openmrs
Client Registry (OpenEMPI)	http://crlin.test.ohie.org:8080/openempi-admin
Client Registry (Mohawk CR)	TBA
Interoperability Layer	http://iol.test.ohie.org:8089/auth/login
Health Worker Registry	http://hwr.test.ohie.org/openhie-hwr/index.php/login
Interlinked Registry	http://hwr.test.ohie.org:8984/CSD
Terminology Service	http://ts.test.ohie.org:8080/dtsserverws (dtsuser/dts)
Facility Registry	http://fr.test.ohie.org/
Shared Health Record	http://shr.test.ohie.org:8080/openmrs

Demo

[OpenHIE Demo](#)