

New Developer Onboarding with OpenLMIS

Updated: May 3, 2018

Before You Start

1. Read the product overview to get a sense of what OpenLMIS can do:
<http://openlmis.org/product/>
2. Watch YouTube videos to get a tour of OpenLMIS:
 - a. [A Better World Through OpenLMIS](#) (3 min)
 - b. [Requisitions](#) (18 min)
 - c. [Stock Management](#) (13 min)
 - d. <https://www.youtube.com/watch?v=nWkXhnauweA> (OpenLMIS oversight)
 - e.
3. Play with the interactive demo to see OpenLMIS in action:
 - a. URL: <https://demo-v3.openlmis.org/>
(this site shows the newest stable release; but you can also use <https://test.openlmis.org/> to see the latest build updated after every commit)
 - b. Login Credentials
 - i. Username: 'administrator'
 - ii. Password: 'password'
 - iii. For all other user accounts (to try different roles/permissions) see <https://github.com/OpenLMIS/openlmis-auth/tree/master/demo-data>
4. Learn about the code:
 - a. View the components documentation: <http://docs.openlmis.org/en/latest/components/>
 - b. Read about the Architecture behind OpenLMIS v3:
<http://docs.openlmis.org/en/latest/architecture/>
5. Read Contributing to OpenLMIS: <http://docs.openlmis.org/en/latest/contribute/>
6. Review the OpenLMIS Conventions: <http://docs.openlmis.org/en/latest/conventions/>.
If you're contributing code to OpenLMIS, these are the conventions to follow.

Getting Started

1. Set up Docker and start OpenLMIS on my workstation
 - a. What is required?
 - i. Use of the Command Line/Terminal is required for a local install or dev environment, so you must be familiar with that.
 - ii. Use of Git and GitHub is required to clone, pull and make pull requests.
 - iii. A processor that supports virtualization, i.e. it must be able to run Docker.

- iv. We recommend a workstation with at least 16GB of memory and 2 CPU cores.
 - v. Docker is required, and allows you to develop and run OpenLMIS on Linux, Mac or Windows.
 - vi. Internet connectivity is required for the steps below.
- b. Installing Docker:
 - i. Download [Docker Community Edition](#), which is freely available for Linux, Mac, and Windows.
 - ii. If you are on Linux, you need to [install Docker Compose](#). On Mac and Windows, Docker Compose is bundled into Docker Community Edition.
 - iii. If you are on Windows, you may need a special version of Docker. [Docker Community Edition for Windows](#) works if you have Windows 10 Professional or Enterprise 64-bit. For previous versions of Windows or Home Edition you need to download the legacy [Docker Toolbox](#). However, some developers have faced challenges getting OpenLMIS running with the Docker Toolbox version. [This document](#) may also help Windows-based developers.
 - iv. *Configuration note: In all platforms except Linux, configuring Docker preferences/settings is required. **We recommend giving Docker at least 8GB of memory and 2 CPU cores.***
 - v. *Version note: OpenLMIS recommends the latest Stable version of Docker. Docker Community Edition includes Docker Engine and Docker Compose. This meets the minimum requirements to run OpenLMIS v3.*
- c. Installing and running OpenLMIS v3 (stable or latest)
 - i. For latest SNAPSHOT: To install the latest code snapshot, git clone or download the Reference Distribution GitHub repo:
<https://github.com/OpenLMIS/openlmis-ref-distro>
 - ii. For stable: To install the most recent stable release, download an OpenLMIS Reference Distribution stable release here:
<https://github.com/OpenLMIS/openlmis-ref-distro/releases>
- d. Configuring and Running OpenLMIS v3
 - i. Once you have cloned or downloaded the Reference Distribution...
 - ii. Follow the README.md instructions under 'Quick Setup':
<https://github.com/OpenLMIS/openlmis-ref-distro#quick-setup>
(This will guide you through steps to configure your settings and spin up each microservice component to boot up OpenLMIS.)
 - iii. If you run into any problems during these steps, reach out for help in Slack [#help](#) or see our [Docker Cheat Sheet](#) for ideas.
- e. Logging in

- i. Once OpenLMIS is running, a set of login accounts will be available for you as part of the demo data loaded by default:
<https://github.com/OpenLMIS/openlmis-auth/tree/master/src/main/resources/db/demo-data>
 - ii. If you get a 502 Bad Gateway error while trying to log in, see the note in the Quick Setup linked above about waiting for all components to start up.
 - f. Success! You have a copy of OpenLMIS v3 up and running!
2. Make a dev environment for a specific component to work on
- a. Each microservice component of OpenLMIS has its own Git repository and its own build script. To develop for OpenLMIS, the general process is to choose which component to work on and follow the instructions for [Git and Pull Requests](#).
 - b. Most of the Service components use Java 8 and Java Spring to provide RESTful JSON APIs. Most use a PostgreSQL database for storage. Install Java 8 JDK in your environment (i.e. for Debian Linux: `sudo apt-get install openjdk-8-jdk`)
 - c. Install an IDE such as IntelliJ IDEA or the Java IDE of your choice.
Note: OpenLMIS only has helper tasks for IntelliJ IDEA. In addition, there are helpful tips and documentations specific to IntelliJ IDEA. Keeping up with these configurations will improve your developer experience.
 - i. IntelliJ IDEA Knowledge Base <https://openlmis.atlassian.net/wiki/x/MQCyB>
 - ii. SonarQube and SonarLint User Guide
<https://openlmis.atlassian.net/wiki/x/QwDABQ>
 - d. Choose a microservice that you wish to build such as [openlmis-report](#) and follow the [Quick Start guide in the README](#). OpenLMIS uses docker containers for development where dependencies are automatically downloaded. Your developer environment does not need additional dependencies.
 - e. The UI components use JavaScript with AngularJS 1 and SASS for CSS. The Reference UI includes multiple components built together into a single-page application that runs in Chrome or Firefox. You can use any IDE to edit the Javascript, HTML and CSS files. Generally you check out one GitHub repo for the part of the UI you are working on and use the [Dev UI](#) to build a local copy of the Reference UI with your changes. Then that Reference UI runs with the Reference Distribution running all the other components.
 - f. See the README file of the specific component you want to work on. Each component README file has instructions for building and testing changes. See all components here: <http://docs.openlmis.org/en/latest/components/>
Note: UI components are grouped as a “vertical slice” with the API service component

they match, and are also appear in the Reference UI section.

3. Connect with other developers and the OpenLMIS Community
 - a. Sign up for an [OpenLMIS Slack account](#) to start communicating with OpenLMIS developers around the world.
 - i. When joining Slack, please introduce yourself, letting everyone know the following: Your preferred name (and pronouns); Where you are based; Your years of coding experience; and Technologies you are familiar with.
 - b. Sign up for an [Atlassian OpenLMIS account](#) to get access to JIRA and Confluence for OpenLMIS tickets and wiki.
 - c. Check out the OpenLMIS Community: <http://openlmis.org/about/community/>
 - i. Learn about the Committee meetings and email lists, [product](#) and [tech](#)
 1. Join the [Dev Forum google group](#) email list
4. Find a programming task to get started with.
 - a. Look for any [“QuickWin” ticket](#) or an open [Bug ticket](#) in JIRA. These are usually small, achievable tickets that might be a good place to get started programming in OpenLMIS.
OR
 - b. Visit the [2018 OpenLMIS Ideas Page](#) for projects to pick up and contact the mentor for each via Slack.
5. Feeling stuck? Reach out to other developers in the Community:
 - a. Ask for help on Slack in #gsoc and #help [OpenLMIS Slack: \(invite link\)](#)
 - b. Ask the dev forum: <https://forum.openlmis.org/c/developers>
 - c. Ask a mentor via Slack (look at the [2018 OpenLMIS Ideas Page](#) for mentor list)