

OpenMRS SDK

Primary mentor	Rafal Korytkowski
Backup mentor	Burke Mamlin
Assigned to	Unknown User (h3llborn)
Blog	h3llborn.com

Abstract

We want to offer module developers a more friendly development environment like the one provided by e.g. [Atlassian](#).

Objectives

1. Make it possible to run OpenMRS with an in-memory/embedded DB (configuration free and fast development setup thanks to skipping liquibase scripts, requires small changes in core)
2. Deploy a module to any version of OpenMRS straight from the maven module archetype (requires modifications in maven module archetype)
3. Improve the [OMOD Reloader Module](#) to read configuration continuously without restarting OpenMRS, create a configuration page to change settings straight from webapp
4. Package OpenMRS SDK. Consider one of the 2 approaches:
 - a. Package maven and simple bash scripts in a zip file which when unpacked on different Windows/Linux/OS X will let you run: (this is a MUST have)
 - i. omrs-create-module -- invokes the archetype & walks through creating a module
 - ii. omrs-run -- runs the module in the current directory (build module & deploy it into an OpenMRS instance)
 - b. Package JVM, maven and jgit. See for example <http://docs.oracle.com/javafx/2/deployment/self-contained-packaging.htm> (this is just nice to try)
 - i. Create a module (backed by the maven module archetype)
 - ii. Build and (re)deploy a module to any version of OpenMRS (backed by the maven module archetype)
5. Update documentation on the wiki and create a screencast

Extra Credit

1. Add more features to the OpenMRS SDK application e.g.
 - a. Clone any repository from github
 - b. Get latest changes
 - c. Create topic branch for a ticket and push it to fork

Timeline (approx)

May 25 – June 16 (Community Bonding Period):

- Get to know my fellow peers and co-workers
- Throughout this period, I will as always be available on IRC to further discuss the proposal with members of the OpenMRS community so that I will be able to adjust or add things that I may have not thought of before.
- I will be actively engaged in discussions with my Mentor to further understand and agree on what exactly needs to be modified so that OpenMRS will be able to function with an embedded database, while also examining where improvements in the reload module need to be made.
- My Mentor and I will also be looking at the options presented previously, on how best to approach the deployment of the SDK on various operating systems. This will include the installation itself as well as the use of scripts.

June 17 – July 17:

- Consult with developers and mentor on the wants for the embedded Database
 - Options to Consider
 1. [H2 Database](#)
 2. [HyperSQL](#)
- Work on integrating the chosen database
- Make sure OpenMRS is able to function properly with the implementation

July 18 – Aug 11:

- Determine the best way to package the SDK
- Consult with Mentor and peers
- Work on packaging of the SDK
 - Commands (minimum):
 - omrs-version
 - omrs-create-project
 - omrs-run
 - omrs-debug
 - omrs-create-module
 - omrs-build
 - omrs-clean
- Test to see if everything works

Aug 12 – Aug 31:

- Modify maven archetype settings for easy deployment of modules (involve [Unknown User \(breeze\)](#) in discussion)
 - Consider similar approach done for [release tests](#)
- Create scripts for both Linux and Windows to allow for module creation and deployment to OpenMRS instance
 - Possibly consult Atlassian SDK scripts

Sept 1 – Sept 16:

- Work on improving the reload-module for seamless integration
- Discuss the wants and needs, set up a list of things to accomplish
- Test and debug the improved version of the module

Sept 17 – Sept 23:

- Work on screen cast and updating documentation
- Check code for quality and make sure all works well
- Write tests, if there need to be any

Resources

1. [Atlassian SDK](#)
2. [Video overview of Atlassian SDK](#)
3. [JavaFX self-contained packaging](#)
4. [OpenMRS-SDK-proposal.pdf](#)