Advancement of OAuth2 Module and Improvements in SMART OWA

Primary mentor	Unknown User (harsha89)
Backup mentor	Unknown User (lahiruj)
Assigned to	
Interested People	

Abstract

The OAuth2 module is functional and works fine against OpenMRS Reference Application 2.x releases after the work on OAuth module enhancements and SMART apps support. Also the EHR Launch Flow for SMART Applications is fully integrated in the module.The objective of this project is to upgrade the OAuth2 module by migrating to Spring Security OAuth2 2.x and creation of all new SMART OWA which fully supports SMART Apps functionality. Another major objective is to extend the functionality of scopes and launch context by adding more scopes. The api layer of the module is well tested and omod layer needs to tested with proper unit tests so as to increase code coverage of the module. Also check and upgrade Spring, Spring Security, Jackson and Hibernate dependencies making sure that module works fine against the latest OpenMRS Reference Application release. The final aim is to get OAuth2 module ready so as to get it bundled with next release of OpenMRS.

Project Champions

Unknown User (mavrk) Unknown User (maany) Unknown User (harsha89) Unknown User (prabodhk)

Goals

- **Migrating to Spring Security OAuth2 2.x** : Currently the OAuth2 module works on Spring Security OAuth2 version 1.0.5. Upgrade the version to Spring Security OAuth2 2.x ensuring that the module works fine against latest OpenMRS release. This would require many dependencies and code changes. Please see the Platform Release Notes.
- **SMART OWA** :At present, we have a simple SMART OWA with basic UI to support the SMART apps in OpenMRS Reference application. We aim at making a new SMART OWA with better UI and functionality of registering, running and editing SMART Apps.
- **Scopes and Launch Context** : The OAuth2 module currently support patient and user specific scopes and launch context. Extend this to other resource specific scopes and launch context. Support for more complex scopes can be added. Read about scopes and launch context here [http://hl7.org/fhir/smart-app-launch/scopes-and-launch-context/index.html]
- **EHR-launch flow** : As of now, SMART apps need to be configured according to OpenMRS and the client secret needs to be hard-coded into the authorization headers in order to successfully run some SMART apps. This needs to be fixed.
- **Increase Code Coverage** : Write unit tests for the omod layer and increase code coverage. Research on writing tests for controllers needs to be done and finally omod layer must be properly tested. Follow OpenMRS Unit Tests Conventions and also add raw test data.

Expected Deliverables

- An OAuth2 module upgraded on Spring Security OAuth2 version 2.x (This is a priority!)
- A proper react based SMART Open Web Application which fully supports SMART Apps. (Begin after the first deliverable is complete)
- Support for more SMART scopes and launch context.
- Increased overall test coverage.
- An OpenMRS OWA demonstrating the implicit grant type flow (Bonus Karma points, if time permits)
- Android Client demonstrating as Password protocol flow (Bonus Karma points, if time permits)

Sounds cool. How can you get started?

- Go through the OAuth specification (RFC 6749) and understand OAuth2 and it's grant types.
- Go through the OAuth2 module and all child pages to see what work is already done.
- Go through the project report https://pkatgithub.github.io/GSoC-2018-Final-Evaluations/from GSoC 2018.
- Go through the project report https://mavrk.github.io/GSOC-2017-final/ from GSoC 2017.
- Run the module on your machine and test it's functionality.
- Take a look at how the Spring Security and Spring Security OAuth2 projects are wired up in the module.
- Take a look at authentication scheme used by SMART Apps and identify how OAuth2 module can serve as the authentication manager for such apps .
- Come up with timeline along with how each week has used to develop the module to meet with required goals.
- Create tickets in JIRA for tasks to be completed during GSoC.

Additional Tips for Proposal

While not mandatory at all, it would be great help if you include the following in your proposals:

- UI for the SMART OWA with various use cases.

Requirements

- Good Java and React skills.

- Familiarity with J2EE web programming (e.g., JSPs)
- Familiarity / willing to learn OAuth2.
- Soft skills to interact with the HAPI and FHIR community and OpenMRS community in order to gather requirements and technical feedback.
- Learn SMART Apps.

Resources

- Understanding OAuth2 : https://tools.ietf.org/html/rfc6749
- http://blog.facilelogin.com/2012/08/wso2-oauth-20-playground-with-wso2.html
- UI Framework Guide : UI Framework Step By Step Tutorial[https://wiki.openmrs.org/display/docs/UI+Framework+Step+By+Step+Tutorial]
- SMART on FHIR[http://docs.smarthealthit.org/]
    - Authorization Guide[http://docs.smarthealthit.org/authorization/]
    - Scopes and Launch Context [http://hl7.org/fhir/smart-app-launch/scopes-and-launch-context/index.html]
- OAuth module enhancements and SMART apps support
- OpenMRS - OAuth2 Module
- https://pkatgithub.github.io/GSoC-2018-Final-Evaluations/
- mavrk.github.io/GSOC-2017-final/