Open Web App Development Workflow

This page will briefly discuss the development workflow for developing Open Web Apps. For a description of what Open Web Apps are, see the wiki page. Much of this document is derived from the Developing an HTML+JS Open Web App Quickly wiki page.

1. Setup OpenMRS

In order to develop Open Web Apps, you will need an OpenMRS server running locally. The ways to do this are described below.

Standalone

The quickest and easiest way to set up a server locally is to download the latest standalone server from the OpenMRS downloads page. You will need Java 1.7+ to run the standalone server. Once it's downloaded, extract the ZIP archive and run the `jar` file either by double clicking it or executing the following command:

Run the OpenMRS standalone server

```
java -jar openmrs-standalone.jar
```

The first time you run the server you will be asked if you want to insert dummy data into the system. This is recommended if you want some data to play with.

SDK

Another easy way to run an OpenMRS server is to use the OpenMRS SDK. To use this method you must have MySQL 5.6, Java 1.7+ and Maven 3+ installed. Newer versions of MySQL, even with compatibility options, will not work. To install and configure the OpenMRS SDK, run the following:

Install the OpenMRS SDK

```
mvn org.openmrs.maven.plugins:openmrs-sdk-maven-plugin:setup-sdk
```

To create a new OpenMRS Platform server, run the following:

Create a Platform Server

```
 mvn openmrs-sdk:setup-platform -DserverId=platform -Dversion=1.11.5
```

Finally, navigate to the server directory (probably `~/openmrs/platform`) and run:

Run a Platform Server with the SDK

```
mvn openmrs-sdk:run
```

It is also possible to run a Reference Application server using the SDK. See the SDK docs for instructions on how to do this.

Enterprise

The enterprise install is usually meant for production environments, and involves installing MySQL and Tomcat manually, then downloading the OpenMRS Platform WAR file from the OpenMRS downloads page and deploying it to Tomcat. See the full documentation here.

Docker

To set up and instance of the OpenMRS Platform using Docker, follow the instructions in the README file in this repository. To support rapid local deploys, you'll need to make the following change to your `docker-compose.yml` file:

Support Local Deploys to Docker

```
openmrs-platform-tomcat:
  build: tomcat
  ports:
    - "8080:8080"
  container_name: openmrs-platform-tomcat
  links:
    - openmrs-platform-mysql
+  volumes:
+    - /host/location/of/owadata:/usr/local/tomcat/.OpenMRS/owa
openmrs-platform-mysql:
  build: mysql
  ports:
    - "3306:3306"
  container_name: openmrs-platform-mysql
```

This will create an `owadata` directory at `/host/location/of/` on your Docker host and map that to the Open Web Apps data directory in the container. Once you've scaffolded your app (see section #3 below), edit the `LOCAL_OWA_FOLDER` variable in `gulpfile.js` to point to this directory in order to support local deploys.

2. Install the Open Web Apps module

This is described on the Open Web Apps Module page. It's the same as installing any other other module

3. Scaffold Your Open Web App

A boilerplate Open Web App along can be scaffolded using the Yeoman OpenMRS OWA generator. You will need NodeJS 6+ installed to do this. See the install instructions here.

Once you have NodeJS installed, you need to install Yeoman, as follows:

Install OWA Generator Dependencies

```
npm install -g yo
```

You can then install the generator:

Install OWA Generator

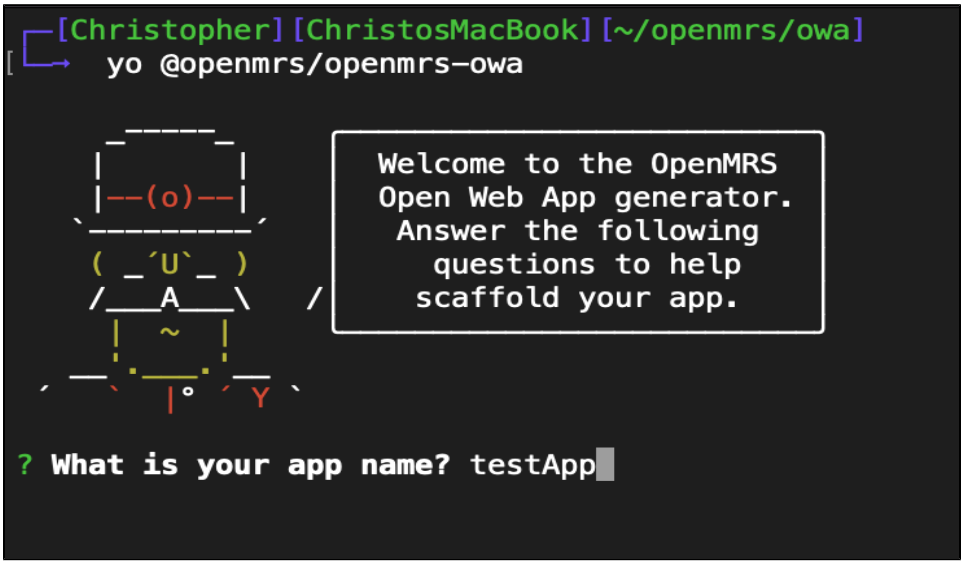
```
npm install --global @openmrs/generator-openmrs-owa
```

Next, create a directory for your Open Web App and change into it. Then run the following to scaffold your app:

Scaffold Open Web App

```
yo @openmrs/openmrs-owa
```

This will start up the Yeoman generator. The output should be something like:



Follow the options in the Yeoman generator in order to scaffold the app. Depending on your setup options it might look something like this when you are done:

```
[Christopher][ChristosMacBook][~/openmrs/owa]
  yo @openmrs/openmrs-owa

     _-----_
    |       |     ╭──────────────────────────╮
    |--(o)--|     │ Welcome to the OpenMRS   │
   `---------´    │ Open Web App generator.  │
    ( _´U`_ )     │ Answer the following     │
    /___A___\  /  │ questions to help        │
     |  ~  |      │ scaffold your app.       │
   __'.___.'__    ╰──────────────────────────╯
 ´   `  |° ´ Y `

? What is your app name? testApp
? What is your app description? App to display and share a list of tests.
? What libraries would you like to include? AngularJS
? What type of server are you running locally? SDK
? What URL will your app be served from? http://localhost:8080/openmrs/owa/testApp/app/index.html
? What is the path of your local Open Web Apps directory? /Users/Christopher/openmrs/my-server/owa
? What is your GitHub username? ChrisBOfficial
? What will the GitHub repo URL be? https://github.com/ChrisBOfficial/testApp
   create package.json
   create webpack.config.js
   create app/manifest.webapp
   create .gitignore
   create app/css/testapp.css
   create app/js/testapp.js
   create karma.conf.js
   create test/encounterSearchTest.js
   create test/observationSearchTest.js
   create test/patientSearchTest.js
   create test/providerSearchTest.js
   create app/js/home/components/breadcrumbs.component.js
   create app/js/home/components/breadcrumbsPatientCreate.component.js
   create app/js/home/components/encounter.component.js
   create app/js/home/components/header.component.js
   create app/js/home/components/notification.component.js
   create app/js/home/components/observation.component.js
   create app/js/home/components/patientCreate.component.js
   create app/js/home/components/patientSearch.component.js
   create app/js/home/components/provider.component.js
   create app/js/home/components/translate.component.js
   create app/js/home/controllers/breadcrumbs.controller.js
   create app/js/home/controllers/breadcrumbsPatientCreate.controller.js
   create app/js/home/controllers/encounter.controller.js
   create app/js/home/controllers/header.controller.js
   create app/js/home/controllers/notification.controller.js
   create app/js/home/controllers/observation.controller.js
   create app/js/home/controllers/patientCreate.controller.js
   create app/js/home/controllers/patientSearch.controller.js
   create app/js/home/controllers/provider.controller.js
   create app/js/home/controllers/translate.controller.js
   create app/js/home/home.js
   create app/js/main/main.component.js
   create app/js/main/main.controller.js
   create app/js/main/main.js
   create app/index.html
   create app/js/home/breadcrumbs.html
   create app/js/home/breadcrumbsPatientCreate.html
   create app/js/home/encounter.html
   create app/js/home/header.html
   create app/js/home/home.html
   create app/js/home/notification.html
   create app/js/home/observation.html
   create app/js/home/patientCreate.html
   create app/js/home/patientCreatePage.html
   create app/js/home/patientSearch.html
   create app/js/home/provider.html
   create app/js/home/translate.html
   create app/js/main/main.html
   create app/img/omrs-button.png
   create app/img/openmrs-with-title-small.png
   create app/img/loading.gif
   create README.md
   create LICENSE


I'm all done. Running npm install for you to install the required dependencies. If this fails, try running the command yourself.
```

Ensure that the `APP_ENTRY_POINT` inside `webpack.config.js` is set to the proper location of your OWA's `index.html`. Congratulations, you have successfully scaffolded your app!

AngularJS Dependency Changes

It's possible that you can encounter TypeErrors if your OWA is using Angular 1.x. Inside `package.json` in the OWA's root folder, remove the ^ before the version number of each dependency related to Angular. Run `npm install` afterwards to install the correct versions of those dependencies. See here for more information.

You may also have to remove the following snippet found inside `home.js` if you are still encountering issues:

Dependency code snippet

```
.config(['$qProvider', function ($qProvider) {
    $qProvider.errorOnUnhandledRejections(false);
}])
```

4. Development

All the Open Web App files are in the `app` directory, everything else is used for building and managing packages. Any files you add manually must be added in the `app` directory.

The `app/manifest.webapp` files contains the information that OpenMRS needs to host your app. See the Open Web Apps Module documentation for details. The `launch_path` property contains the path to your app entry point. This is the page that will be loaded when you click on your app in OpenMRS.

General Tips

If you're building your OWA with Angular, all code should be inside components and designed in a modular manner. This will help facilitate an upcoming migration to a more recent version of Angular.

If you're building your OWA with React, please refer to and follow the style guide. There are also React Components available for use.

For the latest information on which technologies to use, check the OpenMRS Radar which is updated quarterly.

Building The App

To build and package your app in a distributable file, use npm Webpack.

Scaffold Open Web App

```
npm run build:prod
```

This creates a zip file in the `app` directory. You can then upload this to an OpenMRS implementation using the Open Web Apps Module.

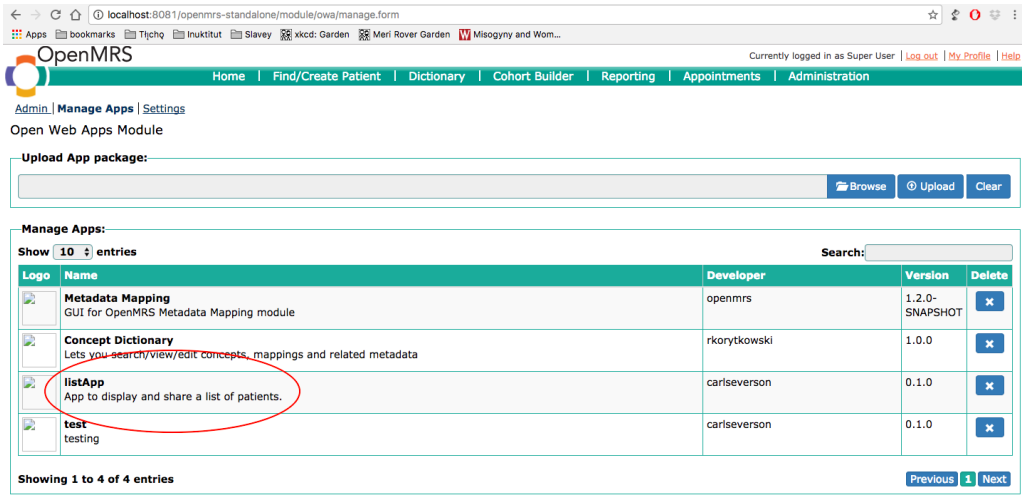
Local Deployment

To speed up the development workflow, we can deploy directly to the app data directory on the filesystem, again using using npm Webpack:

Scaffold Open Web App

```
npm run build:deploy
```

The app should now show up in your OpenMRS implementation (**Administration -> Open Web Apps Module -> Manage Apps**) and will look something like this:



If you have issues using Webpack to deploy the app, ensure that the deploy directory is correct. Open the `config.json` file in the apps directory and make sure that `LOCAL_OWA_FOLDER` points to the correct directory. For example:

```
{
 "LOCAL_OWA_FOLDER": "/Users/username/openmrs-standalone-2.5/appdata/owa",
}
```

Using Browsersync

npm can use Browsersync to watch the files as you development and dynamically reload the app as you make changes to the code. Each time you save a file in the app's directory, the app will reload and display in the browser. This is an extremely useful tool. To use browsersync run the following command in your app's directory:

Build App

```
npm run watch
```

If you have issues getting browsersync to work, ensure that the app is being injected. Open the `config.json` file in the `app` directory and make sure that `APP_ENTRY_POINT` points to the correct path. Your `config.json` file should look

```
{
 "LOCAL_OWA_FOLDER": "/Users/username/openmrs-standalone-2.5/appdata/owa",
 "APP_ENTRY_POINT": "http://localhost:8081/openmrs-standalone/owa/appList/index.html"
}
```

5. Add Home Screen Link

Since you've built a standalone app, you want to let people access it from the OpenMRS home screen. In the Standalone or the Reference Application, you can do this by going to **System Administration** -> **Manage Apps** -> **Add App Definition** and then adding a definition like:

Home Page Link Config

```
{
    "id": "listApp",
    "description": "Demo App",
    "order": 0,
    "extensions": [
        {
            "id": "demoapp.homepageLink",
            "extensionPointId": "org.openmrs.referenceapplication.homepageLink",
            "type": "link",
            "label": "Demo App",
            "url": "owa/demoapp/index.html",
            "icon": "icon-plane",
            "requiredPrivilege": "Replace with a privilege name, or else remove"
        }
    ]
}
```

You can see the available icons at http://demo.openmrs.org/openmrs/uicommons/icons.page.

For more details about **Adding Icon to Reference Application Homepage**, just visit Adding OWA Icon to the Reference ApplicationHomepage

Resources

- Open Web Apps Module
- Developing an HTML+JS Open Web App Quickly
- Screencast - OpenMRS OWA Web Apps Development Workflow (this screencast is excellent, but references an older workflow using bower and gulp, which have been dropped from the workflow)
- Screencast - OpenMRS Open Web App (OWA) Development Tutorial (this screencast is consistent with newer workflow)
- Release Open Web Application