

# For Open Web App Developers

## Quick Overview

1. Quick Overview
2. What is an OWA?
  - a. Practicality
  - b. Architecture
  - c. Dependencies
    - i. Required
    - ii. Typical Languages
  - d. Tools
3. How to write an OWA
  - a. Installing & Creating an OWA
  - b. Editing an Existing OWA
    - i. Structure
4. Examples
  - a. Example 1
  - b. Example 2
  - c. Example 3

## What is an OWA?

### Practicality

This section features an overview of when OWAs can be used in OpenMRS development.

Open Web Apps (OWAs) are similar to standard websites, but are packaged apps that are contained in a zip file. They can be accessed online but can also be installed and downloaded to a device to work offline. OWAs can also access APIs, mainly the OpenMRS REST API, that allow the device to use features like a calendar or camera offline, compared to some websites that do not allow this access.

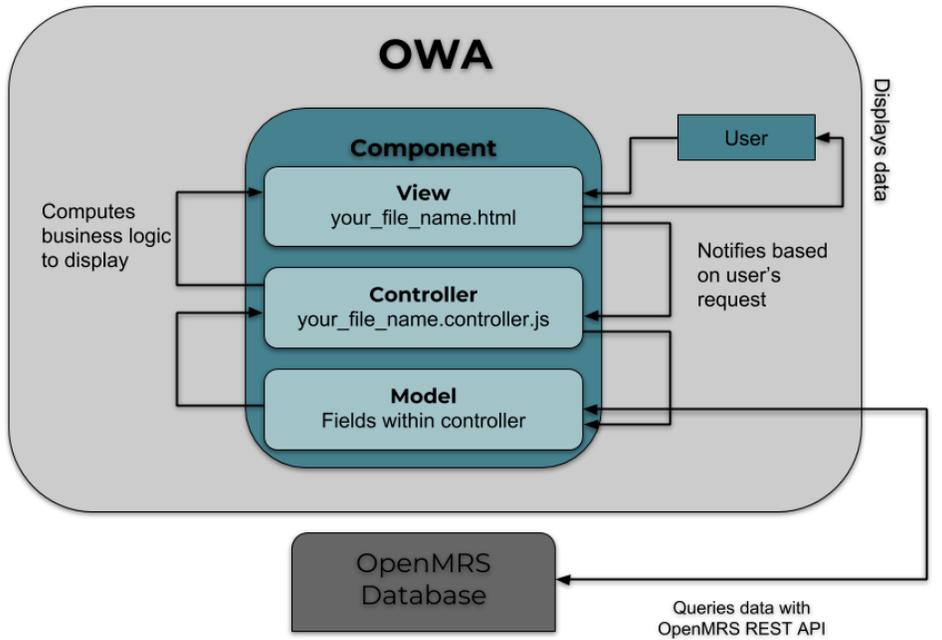
### Architecture

OpenMRS like many open web apps follows an MVC architecture. MVC stands for Model, View and Controller. The Model is where data objects are stored. The model does not interact with the view at all. As seen in figure 1.1, the model only interacts with the controller. The controller, is where all the business logic takes place. The view is what is displayed to the user. The view is where the client interacts with the app. The view interacts with both the controller and the model. The view sends requests to the controller which then sends information back to the view.

Currently, the new OWAs for OpenMRS should be written in React. OWAs that are written in Angular should be organized into components to help facilitate the migration to the most recent version of Angular.

To learn more about Angular, React, and jQuery, go to <https://radar.openmrs.org>.

# MVC Architecture



## Dependencies

This section features dependencies needed to install and work on an OWA along with languages commonly used.

### Required

- Java Development Kit (JDK)
  - <https://docs.oracle.com/en/java/javase/11/install/overview-jdk-installation.html#GUID-8677A77F-231A-40F7-98B9-1FD0B48C346A>
- Node.js
  - <https://nodejs.org/en/download/package-manager/>
- NPM
  - Bundled with Node!
- Maven
  - <https://maven.apache.org/download.cgi>
- OpenMRS SDK
  - `mvn org.openmrs.maven.plugins:openmrs-sdk-maven-plugin:setup-sdk`
- Yeoman
  - `npm install -g yo`

### Typical Languages

- React JS (for new OWAs)
- Javascript
- Angular JS (for old OWAs)
- HTML
- CSS
- MySQL

### Tools

These tools are recommended for the development environment.

- Linter
  - Example
    - eslint
- IDE /Suitable Text Editor
  - Examples
    - Atom
    - Sublime
    - VS Code

# How to write an OWA

## Installing & Creating an OWA

Link to see how to install and create an OWA: <https://wiki.openmrs.org/display/docs/Open+Web+App+Development+Workflow#OpenWebAppDevelopmentWorkflow-2.InstalltheOpenWebAppsmodule>

## Editing an Existing OWA

- Make sure to have NodeJS 4+ installed
- Make code changes as necessary
- Build steps
  - Install Gulp and Bower using the command ``npm install -g gulp bower`` for **\*\*first time users only\*\***
  - Install dependencies using the command ``npm install && bower install`` for **\*\*first time users only\*\***
  - Build using the command ``gulp``
  - After running these commands a zip file will be created in the dist directory, which should be uploaded to OpenMRS Open Web Apps module

## Structure

Once the OWA has been installed, various files will be included and there will be some dependencies used for the OWA. Each OWA follows a typical structure inside the app folder. There is a CSS folder that holds all of the information for the OWA that pertains to CSS and styling. Another folder called JS holds information regarding Javascript and the functions utilized. This folder will also have areas to separate the components related to the OWA. The img folder contains any images that the OWA is connected to, and the manifest.webapp has the information and configuration details about the OWA. Finally, the index.html document is used as a container to house and display the various HTML components of the page.

## Examples

### Example 1

Observation Admin OWA

<https://github.com/openmrs/openmrs-owa-obsadmin>

The purpose of Obsadmin OpenMRS Open Web App is for a Data Manager who might want to clean up a patient's data. This will allow a Data Manager to go in a fix incorrectly recorded data. All of the code for the OWA can be found in the GitHub repository for the project, and the repository includes instructions to download, install, and test the OWA.

### Example 2

Concept Dictionary OWA

<https://github.com/rkorytkowski/openmrs-owa-conceptdictionary>

The purpose of this OpenMRS OWA is providing a single place to add, edit, manage, and view concepts within OpenMRS. It is built using Angular, and can be deployed using NPM. All of the code for the OWA can be found in the GitHub repository for the project, and the repository includes instructions to download, install, and test the OWA.

### Example 3

System Administration OWA

<https://github.com/openmrs/openmrs-owa-sysadmin>

The System Administration OWA aims to migrate the legacy functionalities to the modern open web apps. It increases the user experience and feasibility of the legacy functionalities, extends the usage of the legacy functionalities with REST APIs, extends the administrative features using existing functionalities, and improves the problems which are identified in the legacy UI model.