

Frontend Implementer Documentation

Frontend Configuration

There are two types of pages served by OpenMRS: server-side (JSP/GSP) and client-side (Single-SPA). This documentation will treat the latter, which is the recommended way of developing and configuring user-facing content going forward. The module which handles Single-SPA is [openmrs-module-spa](#). URLs under `/openmrs/spa`` are treated as frontend routes—they serve the frontend application, which is passed the URLs for routing.

There are essentially two pieces to this frontend: the Import Map and the Microfrontends.

The Import Map

An [import map](#) is a browser specification for controlling the url to download in-browser JavaScript modules from. OpenMRS expects that file to be served at `/openmrs/import-map.json``.

Microfrontends

Microfrontends are the mostly-independent JavaScript applications that form the pieces of the new OpenMRS frontend. Check out the [List of frontend modules \(ESM / Microfrontend\)](#). The Import Map tells OpenMRS what Microfrontends to serve, and where those Microfrontends are. There are two good ways to serve these Microfrontends: through a CDN or locally through Tomcat.

Microfrontends on a CDN

This is a good option if you have a centralized deployment and users are assumed to have internet access. Your CDN can be something like Amazon S3. With this kind of deployment, you can update the front-end resources without having to touch the server at all.

Microfrontends locally from OpenMRS server

This is the way to deploy Microfrontends if your servers are deployed on-premises and are intended to be usable without internet access.

Anything put in `/frontend`` under the Application Data directory will be served at `/openmrs/frontend``. You can produce the bundle of ESMs along with an import map, which should go in that directory, is using [Packmap](#).

The subdirectory of Application Data from which Microfrontends are served can be configured using the global property `spa.frontend.directory``. The URL path base can be configured with the global property `spa.frontend.resourceBaseUrl``.