

App Framework Developer Documentation

The App Framework defines a set of lightweight guidelines that help us build different parts of the OpenMRS system in a decoupled way, avoiding a monolithic design that become unwieldy as it grows. The App Framework is agnostic to how you actually implement functionality, and very intentionally allows you to write "server-side apps" using the [UI Framework](#), or "client-side apps" using HTML5 + JavaScript + REST.

Tutorials and Further Documentation

[App Framework Step by Step Tutorial](#)

Terminology

App

App Template

Extension Point

Extension

Context Model

How to define an App

You can define one or more apps in a json file in your module. The framework will scan the classpath and load all files matching `"/apps/*app.json"`, so you might place an app definition file at `/omod/src/main/resources/apps/findPatient_app.json`.

The contents of this file should be a list of [AppDescriptors](#). For example:

```
[
  {
    "id": "coreapps.patientDashboard",          // unique id for this app, unique across all apps, typically
    starts with your module id
    "description": "Basic page summarizing patient's visits and demographics",          // plain text or a
    message code (not displayed to end-users)
    "extensionPoints": [                      // defines where other apps can plug functionality into this app
      {
        "id": "coreapps.patientDashboard.visitActions",          // unique id for this extension point,
        must be unique across all loaded apps
        "description": "coreapps.patientDashboard.extension.visits.description"          // plain text or
        message code (not displayed to end-users)
      },
      {
        "id": "coreapps.patientDashboard.overallActions",
        "description" : "coreapps.patientDashboard.extension.actions.description"
      }
    ],
    "contextModel": [                          // extensions may count on the app providing these values (dev documentation,
    not enforced by the framework)
      "patientId",
      "activeVisitId"
    ]
  }
]
```

How to define an App Template

A generic module can define a template that other modules inherit to create app instances. You can define one or more app templates in a json file in your module. The framework will scan the classpath and load all files matching `"/apps/*AppTemplates.json"`, so you might place a definition file at `/omod/src/main/resources/apps/registerPatientAppTemplate.json`.

The contents of this file should be a list of [AppTemplates](#). For example:

```
[
  {
    "id": "coreapps.template.findPatient",          // unique across all app templates, typically starts with
your module id
    "description": "Simple patient search by name and/or id",          // plain text or message code
    "contextModel": [          // extensions may count on apps with this template providing these values (dev
documentation, not enforced by the framework)
      "patientId"
    ]
    "configOptions": [          // apps that inherit from this template may override these or inherit the
default values
      {
        "name": "onSelectUrl",
        "description": "What url to go to after choosing a patient. Supports {{patientId}}",
        "defaultValue": "/coreapps/patientDashboard.page?patientId={{patientId}}"
      }
    ]
  }
]
```

Another module would instantiate this template in one of its *app.json files like this:

```
[
  {
    "id": "referenceapplication.quickSummary",          // unique app id, (independent of the template)
    "instanceOf": "coreapps.template.findPatient",          // unique id of the template to inherit
    "label": "Quick Patient Summary",          // text or message code
    "description": "Search for a patient by name and/or ID and see a patient summary",          // text or
message code
    "extensions": [          // extensions that attach to extension points of _other_ apps
      {
        "id": "referenceapplication.quickSummary.homepageLink",          // id for this extension, must
be unique across all apps
        "extensionPointId": "org.openmrs.referenceapplication.homepageLink",
        "type": "link",
        "label": "referenceapplication.app.quickSummary.label",
        "url": "coreapps/findPatient.page?app=referenceapplication.quickSummary",
        "icon": "icon-search",
        "order": 1,
        "requiredPrivilege": "App: referenceapplication.quickSummary"
      }
    ],
    "config": {          // in this example we override this setting
      "onSelectUrl": "/somemodule/patientSummary.page?patientId={{patientId}}"
    }
  }
]
```

How to display an App on the home page

An app does not automatically display on the home page (e.g. the Patient Dashboard app couldn't, because it requires you to first select a patient using another app).

To display an app on the homepage, the app needs to attach to the special "org.openmrs.referenceapplication.homepageLink" extension point with an extension of type "link" as shown here:

```
[
  {
    "id": "coreapps.activeVisits",
    "description": "coreapps.activeVisits.app.description",
    "extensions": [
      {
        "id": "coreapps.activeVisitsHomepageLink",
        "extensionPointId": "org.openmrs.referenceapplication.homepageLink",
        "type": "link",
        "label": "coreapps.activeVisits.app.label",          // text or message code (this is shown to
the user)
        "url": "coreapps/activeVisits.page",                // clicking on the app icon goes to this link
        "icon": "icon-calendar",                            // see the uicommons style guide icons
        "requiredPrivilege": "App: coreapps.activeVisits"  // require this privilege to show the
link
      }
    ]
  }
]
```