

Module Portlets

Create a module portlet controller

Module portlet controllers, along with form controllers, are typically put in the `omod/src/main/java/web.controller` folder and have names ending with `PortletController`. Portlet controllers should extend `PortletController`. If there are actions that need to be taken when the portlet is referenced, override the `populateModel` method. Any property added to the model in the `populateModel` method is available UNDER the "model" property. So if you add "myvar" to the model object, in your portlet jsp you access it with `#{model.myvar}`. Here is an example:

```
@Controller
@RequestMapping("*/this.portlet")
public class ThisPortletController extends PortletController {

    private static final Log log = LoggerFactory.getLog(ThisPortletController.class);
    @SuppressWarnings("unchecked")
    @Override
    protected void populateModel(HttpServletRequest request, Map<String, Object> model) {
// your code here
    }
}
```

Make sure the following line exists in your `ModuleApplicationContext.xml` file, substituting your module name for `ModuleID`. It typically is put just above the `</beans>` tag at the end.

```
<context:component-scan base-package="org.openmrs.module.ModuleID" />
```

Create the module portlet jsp

Module portlets are defined in the `omod/src/main/webapp/portlets` folder. In our example, the portlet would be named `this.jsp`. The code inside of a portlet shouldn't have to differ from a standard OpenMRS portlet, meaning that it looks exactly like a jsp (or part of a jsp).

Use the portlet in another jsp page

Portlets are included in a jsp by using the following tag:

```
<openmrs:portlet url="this.portlet" id="thisportlet" moduleId="ModuleID" />
```

Note that you must specify `moduleId="..."`, which differs from using a portlet in core code.