

# Requiring another module in your module



See the code for this example in action in the idgenws module: <http://svn.openmrs.org/openmrs-modules/idgenws/trunk>

## Modify pom.xml files

If your API layer requires a reference to the module, add a reference to the api/pom.xml file:

```
<dependency>
  <groupId>org.openmrs.module</groupId>
  <artifactId>idgen</artifactId>
  <version>2.0.1</version>
  <type>jar</type>
  <scope>provided</scope>
</dependency>
```

If your web layer requires a reference to the module, add a reference in the omod/pom.xml file:

```
<dependency>
  <groupId>org.openmrs.module</groupId>
  <artifactId>idgen</artifactId>
  <version>2.0.1</version>
  <type>jar</type>
  <scope>provided</scope>
</dependency>
```



It is important to make sure you set the parameter `<scope>provided</scope>` so that the jar of the referenced module is **not** included in this module's omod.

## Install the omod into your local repository



If the module are trying to include exists in the parent, you can skip this step

When running "mvn package" you will probably see an error like:

```
Failed to resolve artifact.
```

This happens because maven is unable to find the "idgen" jar in any public repository. We can get around this by either publishing to the openmrs maven repository (mavenrepo.openmrs.org) or by putting it in a local repo.

## Installing omod to a local repo

The maven error output gives you a hint to out to install to a local repo. This script will download the omod/jar and install it for you using that line:

### Shell script for downloading a PRE-maven module and including it

```
#!/bin/sh
moduleid="idgen"
version="2.0.1"
wget https://modules.openmrs.org/modules/download/$moduleid/$moduleid-$version.omod
mv $moduleid-$version.omod $moduleid-$version.jar
mvn install:install-file -DgroupId=org.openmrs.module -DartifactId=$moduleid -Dversion=$version -Dpackaging=jar
-Dfile=$moduleid-$version.jar
```

If the module you want to include is mavenized, get **BOTH** the api and omod jars from the target folders and "mvn install" them. In that case, the omod renaming goes like this:

```
mv $moduleid-$version.omod $moduleid-omod-$version.jar
```

If you are using Windows, use svn to check out the entire module on which you are dependent. You can find the omod/jar files in your Eclipse workspace.

You can follow the same steps by using MS syntax

<environ var>=<value> BECOMES set <environ var>=<value>

\$<environ var> BECOMES %<environ var>%

You can either copy the 2 files to a temporary location and rename the omod there, or you can copy the omod file to its new name; you want to make sure the original file remains in your Eclipse workspace.

Your life will be easier if you add %M2\_HOME%/bin to your path (through My Computer>Advanced>Environment Variables); the maven doc erroneously says %M2\_HOME%

## Installing omod to openmrs maven repo

Open a new ITSM ticket in jira to request that your module be added.

## Modify config.xml file

Add an element like this:

```
<require_modules>
  <require_module version="2.0.1">org.openmrs.module.idgen</require_module>
</require_modules>
```