

Module Development I

Creating a basic module

Introduction

- ▶ Modules are the primary way of adding functionality to OpenMRS
- ▶ Modules can be shared with others and published on the OpenMRS website
- ▶ Non-programmers can install them without writing code

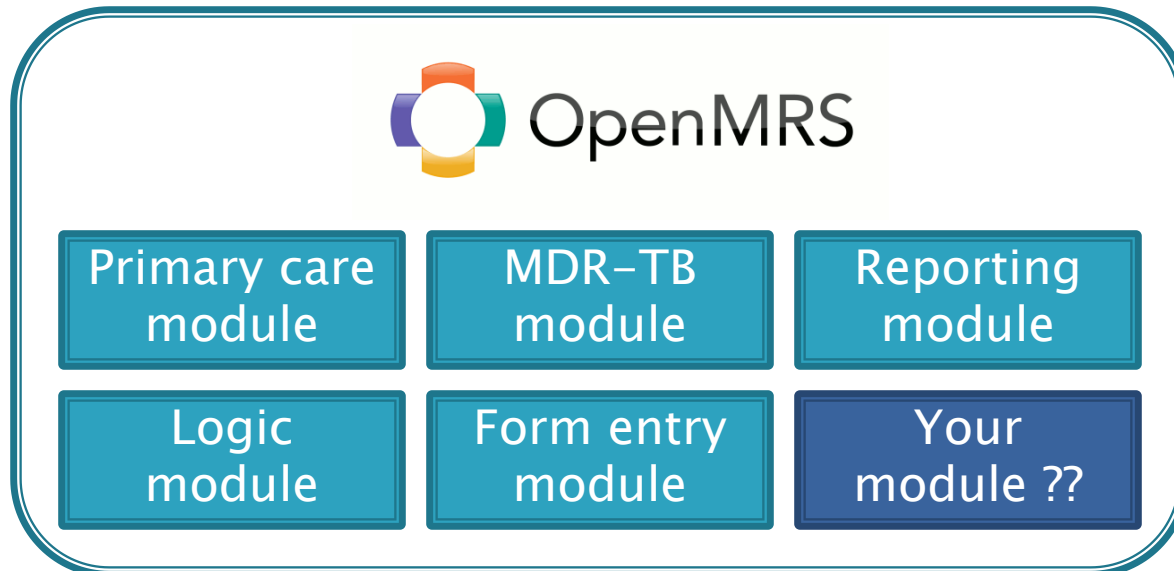


OpenMRS



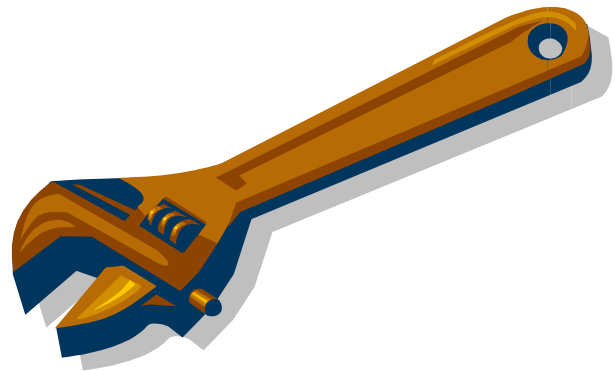
OpenMRS as a platform

- ▶ OpenMRS isn't really an application – on its own it doesn't have very much useful functionality for a clinic
- ▶ It's a flexible platform on which people can build modules to satisfy specific needs...







Module capabilities

- ▶ Add new pages
- ▶ Add items to existing pages
- ▶ Completely override existing pages
- ▶ Add a new database tables
- ▶ Add new services (which can provide new functionality to other modules)



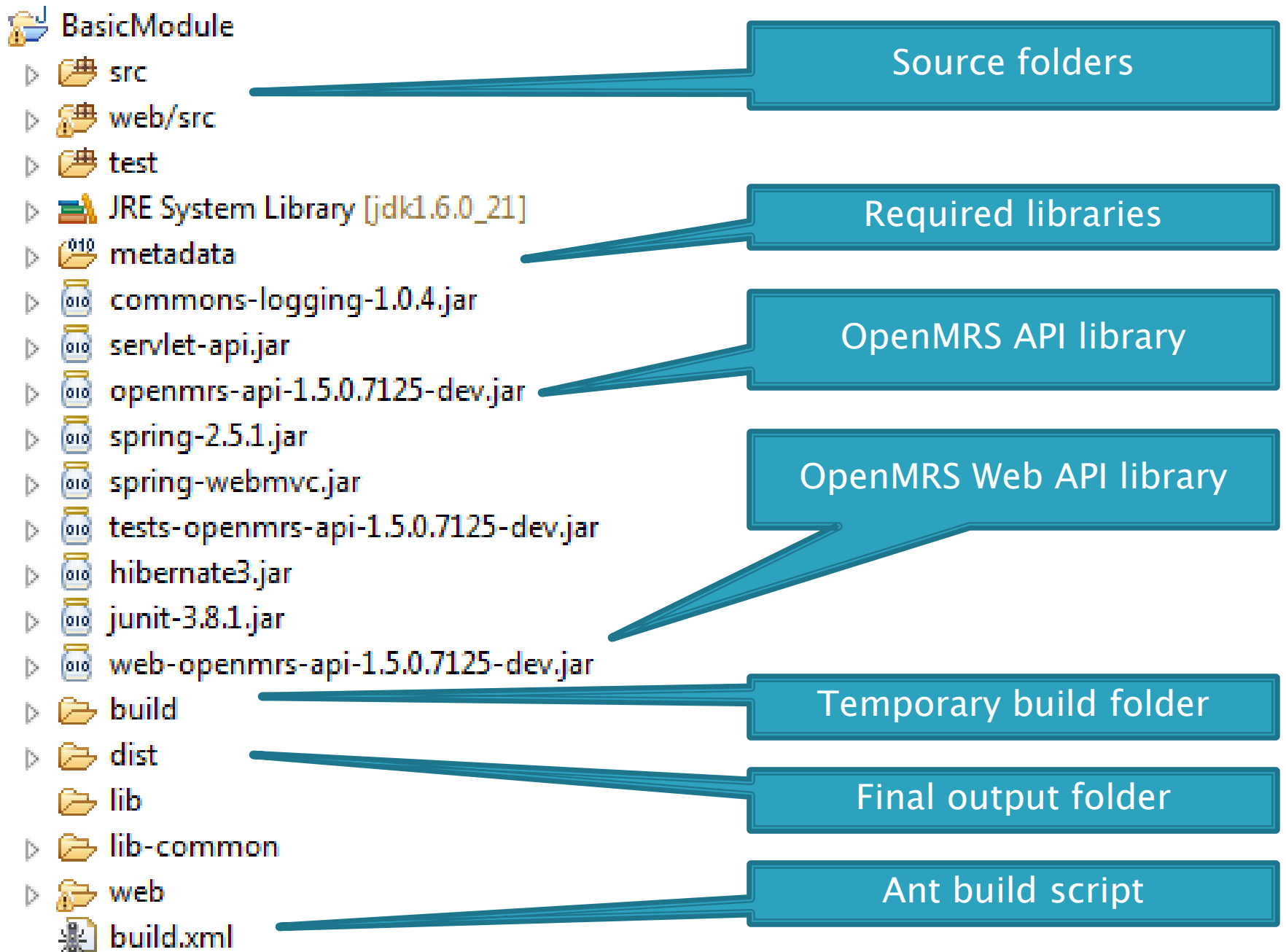
Modules

- ▶ A module IS...
 - A JAR file containing classes and other resources
 - Renamed to have the ***.omod** extension
- ▶ A module is NOT...
 - A runnable Java program
 - A separate web application

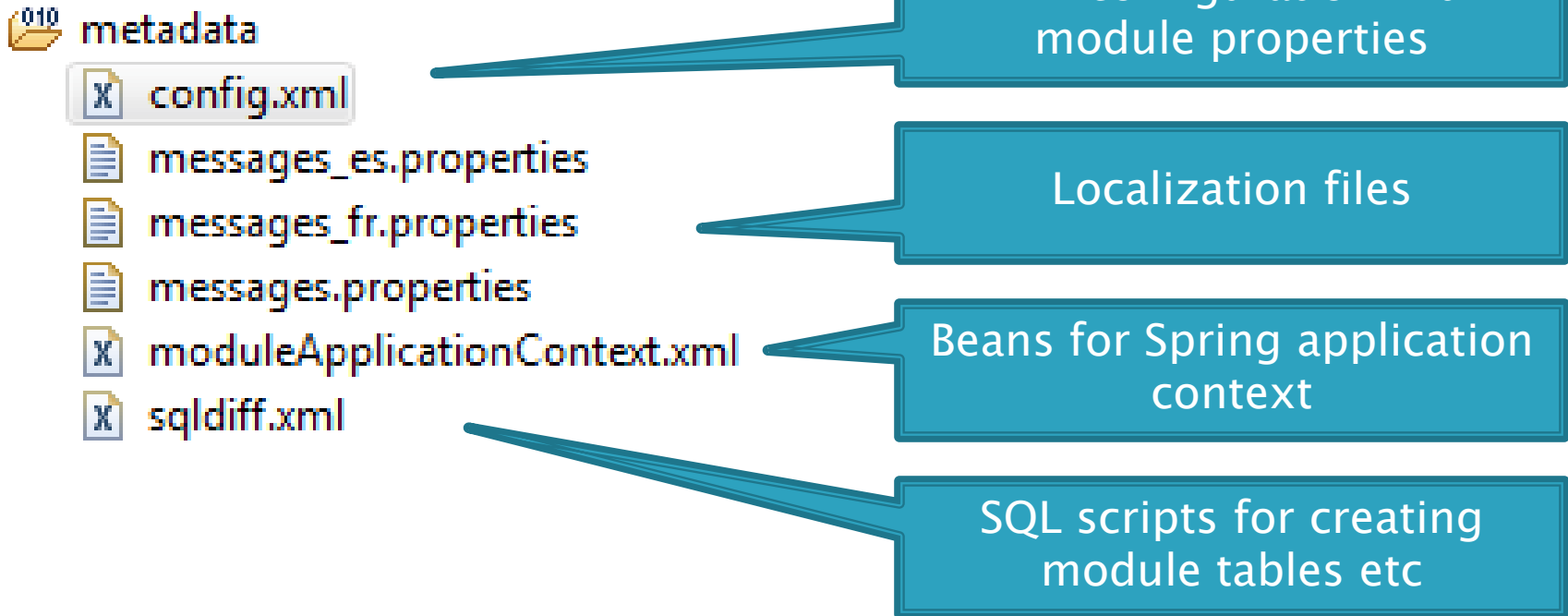
 logmanager-1.3.2.omod	29/06/2010 10:08	OMOD File	154 KB
 htmlwidgets-1.5.0.omod	12/11/2009 15:13	OMOD File	85 KB
 htmlformentry-1.6.4.1.omod	28/06/2010 14:38	OMOD File	149 KB
 formimportexport-1.7.0.omod	30/10/2009 11:03	OMOD File	443 KB

Creating a module

- ▶ You could create a new Java project from scratch... but its a lot easier to start with an existing module... like...
- ▶ **Basic Module:** basic skeleton module
<http://svn.openmrs.org/openmrs-modules/basicmodule/>
- ▶ **AwesomeExample Module:** has an example DAO and service
<http://svn.openmrs.org/openmrs-modules/devexamples/jsp-and-service/>



Metadata



config.xml

```
<module configVersion="1.2">  
  <id>basicmodule</id>  
  <name>Basic Module</name>  
  <version>1.0</version>  
  <package>org.openmrs.module.@MODULE_ID@</package>  
  ...  
</module>
```

Unique id

Base Java
package of
module classes

@MODULE_ID@ is replaced by the value of the <id> tag
and @MODULE_PACKAGE@ by the value of the <package> tag

- ▶ Module id should be unique – if your module will be published you should request an id from code@openmrs.org

config.xml

```
<module configVersion="1.2">
  ...
  <author>Ben Wolfe</author>
  <description>Useful for creating other modules</description>
  <activator>@MODULE_PACKAGE@.BasicModuleActivator</activator>
  <require_version>1.2.0.3400</require_version>
  ...
</module>
```

Class which handles
module being started
and shutdown

OpenMRS version
required by this module

- ▶ Saying that your module requires version 1.X of OpenMRS, implies that it will work and has been tested on version 1.X of OpenMRS !

Creating a new module

- ▶ We can use BasicModule as a starting point..
 1. Change the basic information fields such as:
`id`, `name`, `author`, `description`
 2. Change the name of the module activator,
and then also refactor the class name in the
src folder
 3. Rename bean ids and classes in the
application context file, and refactor those
classes in the *src* folder

Creating a new module

4. Modify the messages files to use the new module id
5. Update AdminList.java to use the new message ids and page URLs
6. Update the project name in *build.xml*
7. Run the build script...
8. Try deploying the new module in OpenMRS

References

- ▶ <http://wiki.openmrs.org/display/docs/Creating+Modules>