The Apache Way 101: Day 10

APACHE LOCAL COMMUNITY
Indore Chapter

Praveen Sharma

Sr. Software Developer at HotWax Systems
4+ years of experience

Apache OFBiz Contributor

APACHE LOCAL COMMUNITY
Indore Chapter

APACHE LOCAL COMMUNITY
Indore Chapter

APACHE LOCAL COMMUNITY

- Local groups of Apache (Open Source) enthusiasts
- Similar to Java User Group, Google Developer Group, Facebook Developer Circles, Mozilla Reps
- Bring together local enthusiasts to meet and exchange knowledge, thoughts and ideas

APACHE LOCAL COMMUNITY
Indore Chapter

Agenda

- OFBiz Development for Beginners
  a. Creating custom component
  b. Creating New Entity Model
  c. Controllers in OFBiz
  d. OFBiz Services
  e. Screen Widget

Apache OFBiz®

Welcome to Apache OFBiz®! A powerful top level Apache software project. OFBiz is an Enterprise Resource Planning (ERP) System written in Java and houses a large set of libraries, entities, services and features to run all aspects of your business.

For more details about OFBiz please visit the OFBiz Documentation page:

OFBiz documentation

OFBiz License

Introduction To Component

OFBiz is organized into groups of directories and files where some directories have a special meaning and are called "*Components*". Some of the set of OFBiz components :-

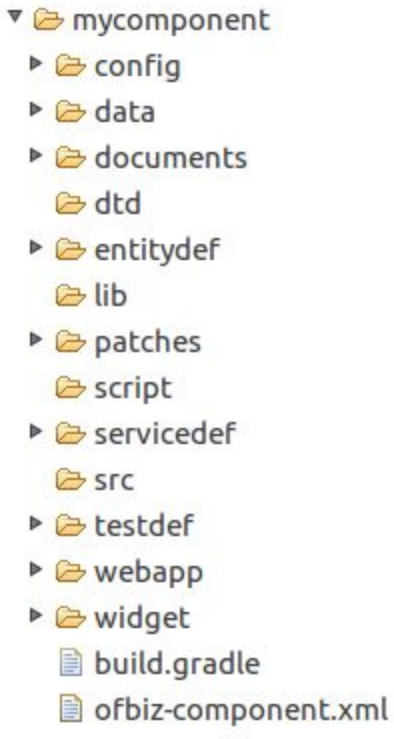
Framework Components: The framework directory contains all the Components which are necessary to run OFBiz.

Application Components: The application directory contains Components that represent many of the business related Applications packaged with OFBiz.

Plugins Components: All Custom Components.

Component Structure

To find an OFBiz Component is to find the top-level directory where the Component begins and to locate the configuration file `ofbiz-component.xml` used to configure that Component.



Creating Component

To create a new component, the following project parameters are used

- **PluginId**: Mandatory.
- **PluginResourceName**: Optional, default is the Capitalized value of pluginId.
- **WebappName**: Optional, default is the value of pluginId.
- **BasePermission**: Optional, default is the UPPERCASE value of pluginId.

Creating Component

It’s very easy to setup a new component in OFBiz in plugin directory. By using this command you can start create a new component.

For Windows: > gradlew createPlugin -PpluginId=mycomponent -PpluginResourceName=MyComponent -PwebappName=mypluginweb -PbasePermission=MYCOMPONENT

For Linux/Mac: > ./gradlew createPlugin -PpluginId=mycomponent -PpluginResourceName=MyComponent -PwebappName=mypluginweb -PbasePermission=MYCOMPONENT

This will create new plugin in **/plugins/mycomponent**

Deleting Component

You can delete the component using this command:-

For Windows: > gradlew removePlugin -PpluginId=mycomponent

For Linux/Mac: > ./gradlew removePlugin -PpluginId=mycomponent

Creating New Entity Model

Entity Engine supports n numbers of entities spread across multiple sources. To create new entities is simply to add entity definition to entity definition file mostly is entitymodel.xml.

To define entitymodel.xml as a entity resource it’s needed to added as resource of type model in ofbiz-component.xml like:

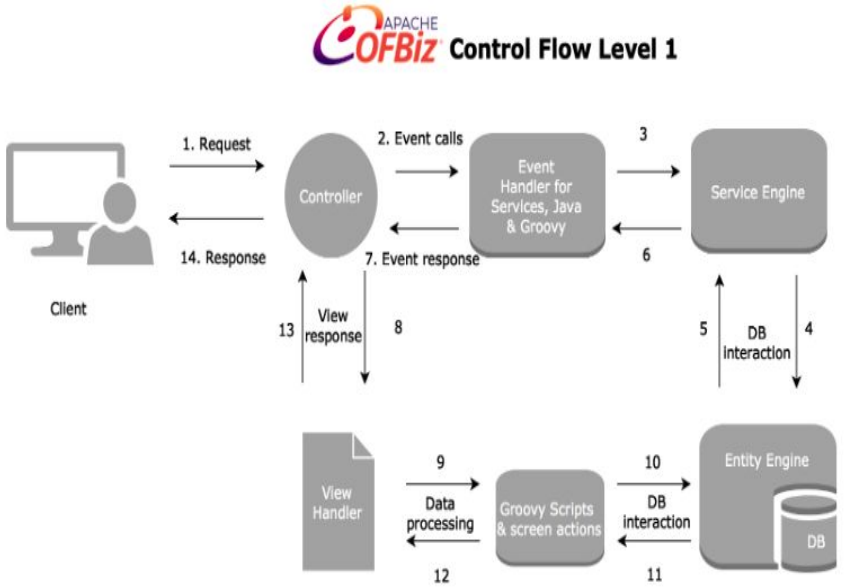
```
<entity-resource type="model" reader-name="main" loader="main" location="entitydef/entitymodel.xml"/>
```

Example:

```
<entity entity-name="demoType" package-name="org.apache.ofbiz.mycomponentdemo" title="demo Type Entity">
  <field name="demoTypeId" type="id"><description>primary sequenced ID</description></field>
  <field name="description" type="description"></field>
  <prim-key field="demoTypeId"/>
</entity>
<entity entity-name="demo" package-name="org.apache.ofbiz.mycomponentdemo" title="demo Entity">
  <field name="demoId" type="id"><description>primary sequenced ID</description></field>
  <field name="demoTypeId" type="id"></field>
  <field name="firstName" type="name"></field>
  <field name="lastName" type="name"></field>
  <field name="comments" type="comment"></field>
  <prim-key field="demoId"/>
  <relation type="one" fk-name="ODEM_OD_TYPE_ID" rel-entity-name="demoType">
     <key-map field-name="demoTypeId"/>
  </relation>
</entity>
```

Controllers

A Browser or other requestor initiates an HTTP or HTTPS request message and the Controller Servlet intercepts that request. The Controller Servlet determines processing control based on the request-map entry in the controller.xml file.



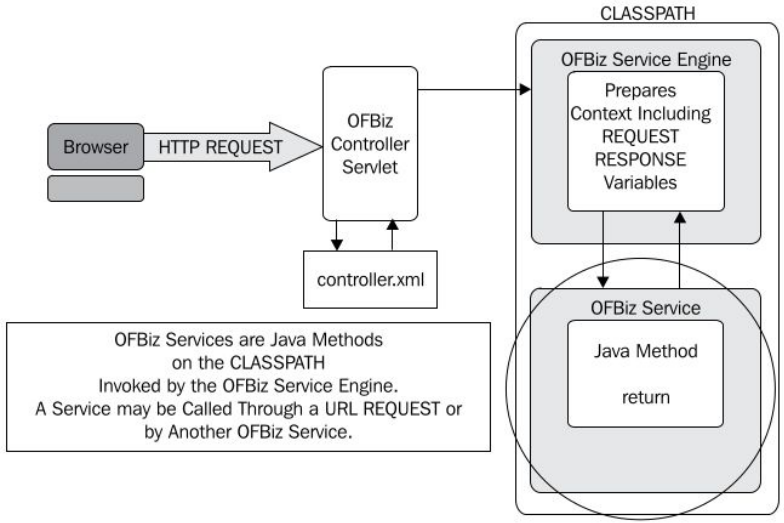
Example:

```
<request-map uri="myService">
        <security https="false" auth="false"/>
        <!-- invoke points to the Service name as defined in the Service Definition →
        <event type="service" invoke="myService"/>
        OR
        <event type="java" path="org.ofbiz.MyEvents" invoke="myEvent" />

        <response name="success" type="view" value="main"/>
        <response name="error" type="view" value="error"/>
</request-map>
```

OFBiz Services

OFBiz "Services" are reusable code snippets implemented as Java, xml or groovy methods. What makes using OFBiz Services so compelling? Unlike OFBiz Events or servlets, the Service context is controlled by OFBiz, thus relieving the programmer from having to do such mundane tasks as managing transactions, handling retries, and validating input and output values. Once written, a Service may be invoked as many times and by as many Service consumers as required.

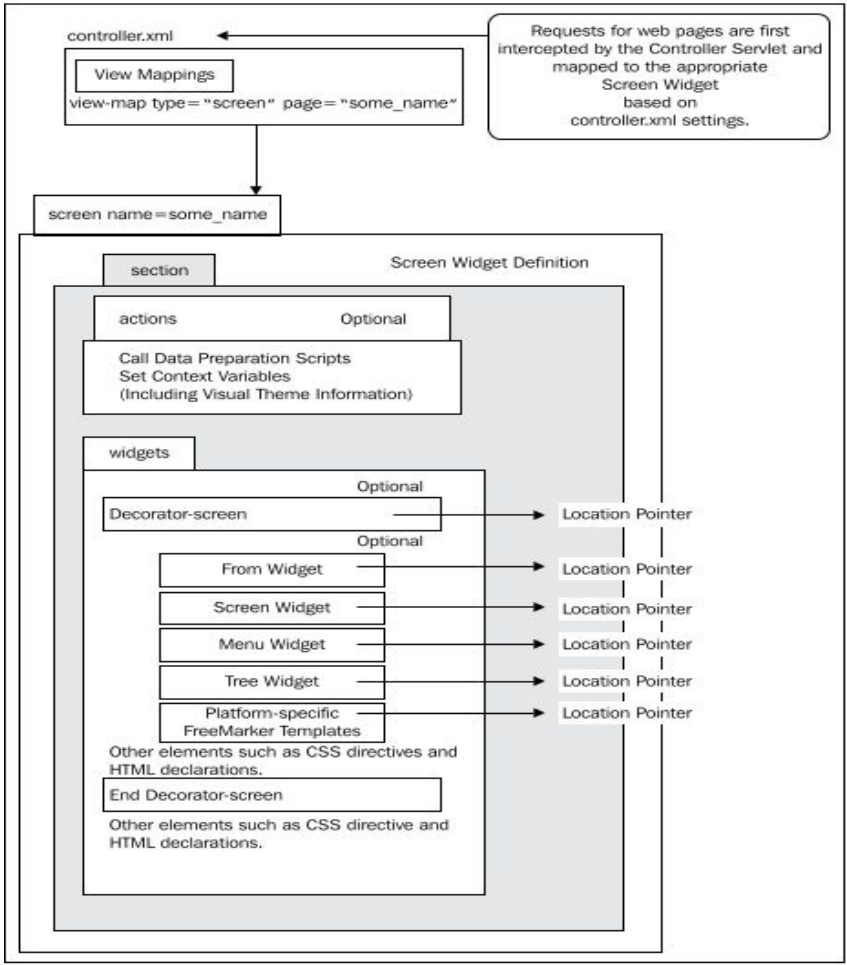


Example:

```
<service name="performFind" auth="false" engine="java" invoke="performFind"
location="org.apache.ofbiz.common.FindServices">
  <description>Some Description</description>
  <attribute name="entityName" type="String" mode="IN" optional="false"/>
  <attribute name="inputFields" type="java.util.Map" mode="IN" optional="false"/>
  <attribute name="fieldList" type="java.util.List" mode="IN" optional="true"/>
  <attribute name="orderBy" type="String" mode="IN" optional="true"/>
  <attribute name="noConditionFind" type="String" mode="IN" optional="true"/>
  <attribute name="distinct" type="String" mode="IN" optional="true"/>
  <attribute name="viewIndex" type="Integer" mode="IN" optional="true"/>
  <attribute name="viewSize" type="Integer" mode="IN" optional="true"/>
  <attribute name="listIt" type="org.apache.ofbiz.entity.util.EntityListIterator" mode="OUT" optional="true"/>
  <attribute name="listSize" type="Integer" mode="OUT" optional="true"/>
  <attribute name="queryString" type="String" mode="OUT" optional="true"/>
  <attribute name="queryStringMap" type="java.util.Map" mode="OUT" optional="true"/>
</service>
```

Screen Widget

Any widget element may contain any number of nested elements within it, including file location pointers to other Screen, Form, Menu, and Tree widget definitions as shown:



Today's Task

- Complete the tutorial

https://cwiki.apache.org/confluence/display/OFBIZ/OFBiz+Tutorial+-+A+Beginners+Development+Guide+for+17.12

- Try find a Jira issue.

Important References

Minimum System Requirement for OFBiz Setup

Best Practices

How to migrate OFBiz from Derby to MySQL database

Groovy DSL for OFBiz business logic

OFBiz Tutorial: A Beginners Development Guide for 17.12
Apache OFBiz Youtube Tutorials and Videos

Service Engine Guide

Entity Engine Guide

General Entity Overview

Control Servlet Guide

JSP Tag Library Guide

Glossary of OFBiz Terms and Concepts

Data Model Diagrams

Universal Data Model

Apache OFBiz CookBook

Thank You

Like and follow us

- https://www.facebook.com/alcindorechapter/
- https://www.linkedin.com/company/alcindorechapter
- https://twitter.com/ALC_Indore



APACHE LOCAL COMMUNITY
Indore Chapter