

Introduction to OFBiz

Apache Open For Business (Apache OFBiz) is a Platform to build Business Applications



Why Apache OFBiz?

- Full Stack JEE framework
- Universal Data Model
- Universal Business Process API
- Service Oriented Architecture

Applications delivered on OFBiz

- E-Commerce.
- Enterprise applications e.g CRM, Warehouse Management, Order Management, etc.
- Sub organization / Department level applications e.g Fleet Management, Maintenance workshop ticketing, Travel expense reporting etc

Data Model coverage

- Party
- Product Catalog
- Product Promotion & Product Pricing
- Order management
- Inventory and Warehouse management
- Manufacturing
- Accounting and many more

30,000 Feet View

- Applications- Here the core business component resides.
- Framework- Integrates different framework like groovy etc.
- Special purpose folder- Here Secondary Business component are there.
- Hot-Deploy- This is meant for custom development.
- Themes- As the name suggests in includes UI Themes.
- Runtime Folder- It maintains the data, logs, output etc.

create-component

Ant target: create-component

- Creates component with given name in hot-deploy folder e.g
travelexpense

In <webapp> tag, we have different attributes and their purpose is as follows:

name :- defines the name of our web application.

title :- This will be the title of the application which will be shown in the top navigation.

server :- This will let OFBiz know what server to use.

base-permission :- This line requires that the user should have the OFBTOOLS permission to be able to use the application. Since the 'admin' user has this permission we do not have to create any new users.

location :- This will be the location that is the default base directory for the server.

mount-point :- This is the URL used to access this resource. in this case it would be localhost:8080/travelexpense.

app-bar-display :- This will let OFBiz know if we want our component to show up in the main application tabs that are part of the common ofbiz decorator.

Application structure

OFBiz component directory structure.

travel-expense

-config

-data

-documents

-dtd

-entitydef

-scripts

-src

-servicedef

-webapp

-widget

-component.xml

Application organization

- config This Component includes the Uilabels.xml which include support for multiple languages.
- data - seed or test data files.
- documents – Help documents.
- entitydef - Entity definition files.
- script - All application logic in minilang or groovy.
- src - All java source files.
- servicedef - Declare OFBiz services, their definition.
- Webapp - Here we have web application.
- Widget - It includes the Screens.

The ofbiz-component.xml file is responsible for letting OFBiz know where resources are at as well as allowing you to add to the classpath

ofbiz-component.xml

```
<service-resource type="model" loader="main" location="servicedef/
services.xml"/>
  <webapp name="travelexpense"
    title="TravelExpense"
    server="default-server"
    base-permission="OFBTOOLS"
    location="webapp/travelexpense"
    mount-point="/travelexpense"/>
</ofbiz-component>
```

The web app

The webapp component is created in the travelexpense component.

The webapp component includes..

webapp-

-css

-error

-font

-includes

-js

-WEB-INF

-actions

-controller.xml

-web.xml

WEB-INF

The WEB-INF directory in webapp (hot-deploy/travelexpense/webapp/travelexpense/WEB-INF).

An OFBiz web application requires two configuration files, a controller.xml and a web.xml.

The controller.xml tells OFBiz what to do with various requests from visitors: what actions to take and what pages to render. web.xml tells OFBiz what resources (database and business logic access) are available for this web application and how to handle web-related issues, such as welcome pages, redirects, and error pages.

Web.xml

```
<display-name>Open For Business - TravelExpense Component</display-name>  
<description>TravelExpense Component of the Open For Business Project</  
description>
```

```
<context-param>
```

```
  <param-name>webSiteId</param-name>
```

```
  <param-value>travelexpenseSite</param-value>
```

```
  <description>A unique ID used to look up the WebSite entity</description>
```

```
</context-param>
```

```
<context-param>
```

```
  <param-name>localDispatcherName</param-name>
```

```
  <param-value>travelexpense</param-value>
```

```
  <description>A unique name used to identify/recognize the local dispatcher for  
                the Service Engine
```

```
  </description>
```

```
</context-param>
```

Controller.xml

```
<site-conf xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
  <include location="component://common/webcommon/WEB-INF/common-controller.xml"/>
  <description>TravelExpense Component Site Configuration File</description>

  <request-map uri="main">
    <response name="success" type="view" value="FindInvoice"/>
  </request-map>
  <view-map name="FindInvoice" type="screen" page="component://travelexpense/
widget/CommonScreens.xml#FindInvoice"/>
</site-conf>
```

To create invoice-create.ftl

```
1<#include "Prebody.ftl"/>❏
2<h4><strong>Create New Invoice</strong></h4>❏
3<form role="form" action="CreateNewInvoice">❏
4  <input type="hidden" class="form-control" name="invoiceTypeId" value="TRAVEL_EXPENSE"/>❏
5  <input type="hidden" class="form-control" name="partyId" value="EMPLOYEE"/>❏
6  <input type="hidden" class="form-control" name="statusId" value="INVOICE_IN_PROCESS"/>❏
7  <input type="hidden" class="form-control" name="partyIdFrom" value="EMPLOYEE"/>❏
8  <div class="form-group row">❏
9    <div class="col-lg-3">❏
10     <label for="journey-name">Journey Name</label>❏
11     <input type="text" class="form-control" id="journey-name" name="invoiceMessage" required/>❏
12   </div>❏
13 </div>❏
14 <div class="form-group row">❏
15   <div class="col-lg-3">❏
16     <label for="journeyDate">Date Of Journey</label>❏
17     <input type="date" id="journeyDate" name="invoiceDate" class="form-control" required/>❏
18   </div>❏
19 </div>❏
```

The screen looks like...

[Find Invoice](#) [Create Invoice](#)

Create New Invoice

Journey Name

Date Of Journey

Description

To find the invoice- findinvoice.ftl

```
<h4><strong>Find Invoices</strong></h4>❏
<<form role="form" action="ListInvoices">❏
<<div class="form-group row">❏
<<<div class="col-sm-4 col-md-3 col-lg-3">❏
<<<<label for="invoice-id">Invoice ID</label>❏
<<<<input type="text" class="form-control" id="invoice-id" name="invoiceId"/>❏
<<<</div>❏
<<<</div>❏
<<<div class="form-group row">❏
<<<<div class="col-sm-4 col-md-3 col-lg-3">❏
<<<<label for="journey-name">Journey Name</label>❏
<<<<input type="text" class="form-control" id="journey-name" name="invoiceMessage"/>❏
<<<</div>❏
<<<</div>❏
<<<div class="form-group row">❏
<<<<div class="col-sm-4 col-md-3 col-lg-3">❏
<<<<label for="journey-date">Journey Date</label>❏
<<<</div>❏
<<<</div>❏
<<<div class="form-group row">❏
<<<<div class="col-sm-6 col-md-4 col-lg-4">❏
<<<<label>Status</label><br>❏
<<<<label class="checkbox-inline">❏
<<<<input type="checkbox" value="INVOICE_IN_PROCESS" name="status1"> In Process❏
<<<</label>❏
```

The screen looks like..

Find Invoice

Create Invoice

Find Invoices

Invoice ID

Journey Name

Journey Date

dd-mm-yyyy

Status

In Process Approved Paid Cancelled

Find

To add items to invoice

```
<div class="container">
  <div class="row">
    <div class="col-md-8 col-lg-8">
      <h4><b>Invoice ID:</b> <${parameters.invoiceId}</h4>
    </div>
    <div class="col-md-3 col-lg-3">
      <button type="button" class="btn btn-primary pull-right" onclick="addItem();" id="addbtn">Add Item</button>
      <form>
        <input type="hidden" name="invoiceId" value="${parameters.invoiceId}"/>
        <button type="submit" class="btn btn-danger pull-right" formaction="getInvoice" style="display:none;" id="cnclbtn">Cancel</button>
      </form>
      <script type="text/javascript">
        function addItem() {
          document.getElementById("add-item").style.visibility="visible";
          document.getElementById("addbtn").style.display="none";
        }
        function cancel() {
          document.getElementById("add-item").style.visibility="hidden";
          document.getElementById("addbtn").style.display="block";
        }
      </script>
    </div>
    <div class="col-md-1 col-lg-1">
      <form>
        <input type="hidden" name="invoiceId" value="${parameters.invoiceId}"/>
        <button type="submit" class="btn btn-primary pull-right" formaction="editInvoice">&nbsp;&nbsp; Edit &nbsp;&nbsp;&</button>
      </form>
    </div>
  </div>
</div>
```

The screen looks like..

Find Invoice Create Invoice

Invoice ID: 10060

Add Item

Edit

Item Details

| Type | Time | Amount | Comment | |
|----------------|-------------------------|----------|---------|--------|
| Travel Expense | 2014-03-16 11:10:05.616 | \$100.00 | flight | Delete |