Extensibility

The extensibility is the ability to extend new functionality in which the system’s internal structure or data flow are minimally or not affected. As the provided documents say, Broadleaf has considered extensible design.

“Extensibility is at the core of our design, and almost every aspect of Broadleaf can be overridden, added to or otherwise modified to enhance or change the default behavior. This includes all of our services, data access objects and entities.” – from Broadleaf website

There are two types of extensions – extending entities and extending services by inheritance. The following is an implementation example for a custom entity.

@Entity

@Table(name = "MY\_ORDER")

public class MyOrderImpl extends OrderImpl {

private static final long serialVersionUID = 1L;

@Column(name = "MY\_ORDER\_ID")

private String myOrderId;

... getters / setters / equals / hashcode ...

}

Here, we've created a new custom entity that extends OrderImpl from Broadleaf Commerce. We've added our custom order id field (myOrderId), and using JPA annotation, have tied this entity and it's fields to a table and columns in the database.

Now that we've created our entity extension, we need to notify Broadleaf Commerce of its existence and how to instantiate it. To do this, we need to add a bean configuration to application context:

<bean id="org.broadleafcommerce.core.order.domain.Order" class="com.mycompany.order.domain.MyOrderImpl" scope="prototype"/>

Finally, we need to add the new entity class to the persistence unit configuration in your persistence.xml. This could end up looking similar to the following:

<persistence-unit name="blPU" transaction-type="RESOURCE\_LOCAL">

...

<class>com.mycompany.order.domain.MyOrderImpl</class>

...

</persistence-unit>

This is the least amount of configuration required for Broadleaf Commerce to utilize entity. Once configured, Broadleaf Commerce will create an instance of MyOrderImpl whenever a new Order instance is required. The services also can be extended as well.

Broadleaf can be enhanced with add-on modules. Add on modules represent functionality that can be incrementally added to the Broadleaf Commerce framework. Add-on modules may be free, open source or commercial. Examples of free, open source modules include the Inventory and SEO modules. Some other modules are payment module, tax module, shipping module and so on.

In Broadleaf, the administration application is based on our new Open Admin platform, which provides a clear path for customization using standard object oriented techniques. We also can enjoy the same level of extensibility in the admin platform that we already enjoy in the core framework (extending entities and services).

The presentation layer is enhanced from Broadleaf 2.0. First, Broadleaf provides classes that perform all of the necessary logic by @controller annotation. It is up to the implementor to extend the Broadleaf\*Controller class, annotate their own class with @Controller and provide methods that will match certain URLs. It is benefit to customizing URL, sharing controller logic and adding custom logics. We are really easy to customize logics to controller. The following is the adding custom logics example from Broadleaf.

**@RequestMapping(value = "/some/path")**

**public String doSomething(HttpServletRequest request, HttpServletResponse response, Model model) {**

**... do some cool stuff ...**

***// Call the super controller (or not if you want to completely override the functionality)***

**String returnPath = super.doSomething(request, response, model);**

**... do more stuff ...**

***// Return the template specified by Broadleaf (or not if you want to return your own template)***

**return returnPath;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Saved\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PCI Considerations

We have taken measures in our construction and design to help you achieve PCI compliance should you decide to store and use sensitive customer financial account information. Payment account information is referenced separately, allowing you to segregate confidential data onto a separate, secure database platform. API methods have been added to allow inclusion of any PCI compliant encryption scheme. Also, verbose logging is included to track payment interaction history.

Third Party Add On Modules

Third party add-on modules involve an integration with Broadleaf Commerce and another system. Typical uses of these include integrations with payment providers like PayPal, Braintree, and CyberSource.

<http://www.broadleafcommerce.com/docs/core/current/modules>