# Enterprise Routing System Development Guide

## Database Interaction

### Technologies Used

- Hibernate
- MySQL
- Spring
- Struts2

### Connections Detailed

For the Enterprise Routing System data management system a hookup is created between the actual application itself and the MySQL database using Hibernate. This setup is created inside of the '*src/main/resources*' package.

The database itself is instantiated and connected to inside of the '*database.properties*' file located inside the '*resources.database*' package.

The tables are pulled and mapped to actions via the '*resources.hibernate*' package where each of the tables are mapped to an entity as well as specifying the relationships they have to other tables and values so that those can be used to our advantage to ensure data integrity and make manipulating data easier and safer.

To further simplify the process of accessing tables database objects were creating using Spring. 'Bean' files were created to easily access the data via our Java code. These are located inside of the '*resources.spring*' package.

## Excerpts

***Contact hookup example: 'Contact.hbm.xml'***

```
<hibernate-mapping>
   <class name="edu.thangiah.entity.Contact" discriminator-value="Contact" table="contacts"
lazy="false">
          <id name="id" type="java.lang.Long" unsaved-value="null">
           <column name="ID" />
           <generator class="identity" />
       </id>
       <property name="firstName" type="java.lang.String" not-null="true">
          <column name="first_name" />
       </property>
   </class>
</hibernate-mapping>
```

---

***Spring Action Mapping example: 'ContactBean.xml'***

```
<bean id="contactManagementAction"
          class="edu.thangiah.action.contact.ManagementController" scope="prototype">
          <property name="contactDao" ref="contactDao" />
</bean>
```

## Struts Implementation

### Technologies Used

- Struts2
- Spring

### Struts2 Hookup Detailed

To map between the Action files, Bean Files, Database Objects, and JSP pages the Enterprise Routing System takes full advantage of everything Spring and Struts has to offer to make the connections easy and functional.

To create this hook up the first step is to ensure all proper bean files are in the '*resources.spring*' package inside of the '*src/main/resources*' source package. Inside of these bean files all references to database objects must be added as properties to the actual actions themselves. These will be detailed in an example later on.

The next step is to declare these actions inside of the '*struts.xml*' file which is in the main directory of the '*src/main/resources*' source package. This file contains all mappings between actions and their corresponding actions and JSP pages to be displayed, as well as interceptors and global response pages such as login, error, and invalid permissions.

Then the final step is actually accessing it from your JSP file. To do this a simple reference to the action is all you need such as '*updateContact.action*'.

## Excerpts

***Bean Action Mapping example: 'ContactBean.xml'***

```
<bean id="updateContactAction"
        class="edu.thangiah.action.contact.UpdateAction" scope="prototype">
        <property name="contactDao" ref="contactDao" />
</bean>
```

***Struts Action Mapping example: 'Struts.xml'***

```
<action name="updateContact" class="updateContactAction">
        <result name="success" type="redirect">contactManagement.action</result>
        <result name="input">WEB-INF/content/contact/manage.jsp</result>
        <interceptor-ref name="authStack"></interceptor-ref>
        <interceptor-ref name="requireLogin"></interceptor-ref>
</action>
```

***JSP Action Access example: 'manage.jsp' from Contact***

```
<s:if test="getMode()=='edit'">
        <s:url id="updateContact" value="updateContact.action">
                <s:param name="id" value="id" />
                        </s:url>
                <div id="editContainer">
                <s:form name="editForm" action="%{updateContact}" id="contactForm">
                        <jsp:include page="/WEB-INF/content/contact/fieldForm.jsp" />
                        </s:form>
                </div>
```