Wiki

Customize DB Column Sizes

« Back to Configuring

Tags: work in progress customization

Table of Contents [-]

Introduction

There are frequently cases which arise where the default table column sizes are not adequate or you wish to alter the appearance of a form element (such as the display width rendered with the liferay-ui taglib). There is in fact a mechanism which allows this to be managed without having to perform the operations by hand, but only when building from source or customizing an EXT build environment.

Building from Source

This mechanism can only be used when building the portal from source or when using an EXT build environment. See Getting set up for more information on setting up the Liferay sources or see Development in the ext environment for more information on setting up the Liferay EXT build environment.

Without ext environment

There are two ways to alter model hints when developing portlets without ext environment (for example, with Portal Pack). You should create portal-ext.properties file with only one line:

```
```

model.hints.configs=pmh.xml }}} All model hints should be written in **pmh.xml**. Both of these files then must be moved to WEB-INF/classes directory or jarred and put into service/lib directory that's created with build-service task on first run.

Model Hints

The portal-model-hints.xml controls primarily how fields appear when a bean object is being displayed using the "liferay-ui" bean display taglib. However, there are certain configurations used to adjust DB column sizes during the build phase. These are referred to as *model hints*.

In other words, they will alter how the "create" sql scripts are generated, so that during development, you are not forced to maintain post db modifications by hand.

There are two scenarios which you'll encounter: Existing Liferay Service Layer and Modified EXT Service Layer. We'll discuss these in depth later on, but for now, let's go over some valid model hints.

Model Hint Values

Name	Value Type	Description	
auto-escape	boolean	sets whether text values should be escaped	

check-tab	boolean	unknown -- used in *com.liferay.portlet.wiki.model.WikiPage*	
day-nullable	boolean	allows the day to be null in a date field	
default-value	String	sets the default value for the field	
display-height	integer	sets the display height of the form field rendered using the liferay-ui taglib	
display-width	integer	sets the display width of the form field rendered using the liferay-ui taglib	
max-length	integer	sets the column maximum size for SQL file generation	
month-nullable	boolean	allows the month to be null in a date field	
upper-case	boolean	convert all characters to UPPER CASE	
year-nullable	boolean	allows the year to be null in a date field	
year-range-delta	integer	specifies the number of years to display from today's date in a date field rendered with the liferay-ui taglib	

Applying Collections of Model Hints

Model hint collections are used to apply the same series of hints to multiple fields. Examples of these can be shown below as pulled from the *portal-model-hints.xml* file.

```
<model-hints>
    <hint-collection name="CLOB">
        <hint name="max-length">2000000</hint>
    </hint-collection>
    <hint-collection name="URL">
        <hint name="max-length">4000</hint>
    </hint-collection>
    <hint-collection name="TEXTAREA">
        <hint name="display-height">105</hint>
        <hint name="display-width">500</hint>
        <hint name="max-length">4000</hint>
    </hint-collection>
    <hint-collection name="SEARCHABLE-DATE">
        <hint name="month-nullable">true</hint>
        <hint name="day-nullable">true</hint>
        <hint name="year-nullable">true</hint>
        <hint name="show-time">false</hint>
    </hint-collection>
...
</model-hints>
```

Now, when applying a collection of hints, simply use **<hint-collection name="name" />** to reduce the amount of redundant typing.

Use a TEXTAREA for a String

Perhaps you need a multiple line field rendered as a TEXTAREA. This can be accomplished by specifying a *<hint-collection/>* as shown below.

```
<model name="com.some.place.Entity">
   <field name="remarks" type="String">
       <hint-collection name="TEXTAREA" />
   </field>
   ...

</model
```

Use a CLOB for a String

Similar to using a TEXTAREA for a string, CLOBs can be defined by specifying a *<hint-collection/>* as shown below.

```
<model name="com.some.place.Entity">
   <field name="entitySettings" type="String">
       <hint-collection name="CLOB" />
   </field>
   ...

</model
```

Specifying a URL

If you're looking to store a URL as a field, there is even a *<hint-collection/>* to specify. This is demonstrated below.

```
<model name="com.some.place.Entity">
   <field name="homePageURL" type="String">
      <hint-collection name="URL" />
   </field>

   ...
```

```
 </model>
```

Existing Liferay Service Layer

In this case you want to use the existing Liferay service layer without modification, but you need custom column sizes or wish to alter the appearance generated by the liferay-ui taglib. This is requires the use of the Liferay source code and, depending on your environment, *%portal_source%* or *${portal_source}* is used to designate the full path to the Liferay sources.

Liferay Version	Operating System	Path to File	
<4.4	Windows	%portal_source%\portal-ejb\classes\META-INF\portal-model-hints.xml	
<4.4	NIX	${portal_source}/portal-ejb/classes/META-INF/portal-model-hints.xml	
>4.4	Windows	%portal_source%\portal-impl\classes\META-INF\portal-model-hints.xml	
>4.4	NIX	${portal_source}/portal-impl/classes/META-INF/portal-model-hints.xml	

An example snippet of the *portal-model-hints.xml* file is shown below for reference.

```
 <model
name="com.liferay.portlet.documentlibrary.model.DLFolder">
  <field name="folderId" type="String" />
  <field name="groupId" type="long" />
  <field name="companyId" type="String" />
  <field name="userId" type="String" />
  <field name="userName" type="String" />
  <field name="createDate" type="Date" />
  <field name="modifiedDate" type="Date" />
  <field name="parentFolderId" type="String" />
  <field name="name" type="String">
   <hint name="max-length">100</hint>
  </field>
  <field name="description" type="String">
   <hint-collection name="TEXTAREA" />
  </field>
  <field name="lastPostDate" type="Date" />
</model>
```

Notice that certain *String* fields have a child *hint* element?

```
<field name="name" type="String">
  <hint name="max-length">100</hint>
</field>
```

The *max-length* attribute of *hint* is used to adjust the column size (and type) used during SQL script creation. The table below describes the logic used.

Hint	Column Type	
max-length < 4000	VARCHAR(max-length)	
max-length = 4000	STRING	
max-length > 4000	TEXT	

After adjusting the *max-length* of the desired fields, the service generation must be re-executed before changes will appear in the create-.sql files.

Modified EXT Service Layer

Frequently developers use the portal as a base for complex integrated development. In this case development takes place in the *ext* environment. See [Development in the ext environment].

Once you've defined your new services in the service.xml files, you must run service generation at least once before you can adjust the model hints.

Once you've run the service generation, locate the file

Liferay Version	Operating System	Path to File	
<4.4	Windows	%ext_source%\ext-ejb\classes\META-INF\ext-model-hints.xml	
<4.4	NIX	${ext_source}/ext-ejb/classes/META-INF/ext-model-hints.xml	
>4.4	Windows	%ext_source%\ext-impl\classes\META-INF\ext-model-hints.xml	
>4.4	NIX	${ext_source}/ext-impl/classes/META-INF/ext-model-hints.xml	

The file will contain one entry for each *entity* you've defined in your combined service.xml files. Modification is the same as above.

Note: Each time a new *entity* is defined, you must re-run service generation before finding the listing in *ext-model-hints.xml*. Also, existing hint customizations on unmodified *entities* will be maintained across service generation executions.

Liferay Model Hints Default Settings

Default values for TEXT (String) fields

Name	Value	
display-height	15	
display-width	350	
max-length	75	

Default values for TEXTAREA (String) fields

Name	Value	
display-height	100	
display-width	500	
max-length	4000	

Overwriting Liferay Default Settings

You can also use ext-model-hints.xml to overwrite the portals default hints. For example I discovered that the displayed input fields of a users phone number and it's extension are much to short.

A quick adding and editing of the *ext-model-hints.xml* solved the problem in mere seconds, as demonstrated below.

```
<model name="com.liferay.portal.model.Phone">
  <field name="number" type="String">
```

```
         <hint name="display-width">200</hint>
      </field>
      <field name="extension" type="String">
      <hint name="display-width">150</hint>
      </field>
   </model>
```

Set Default Hints for an Entity

It is also possible to set some default hints for all fields in an entity. This is done using the *<default-hints/>* tag and is demonstrated below to set the default *display-width* to 200.

```
<model name="com.some.place.Entity">
   <default-hints>
       <hint name="display-width">200</hint>
   </default-hints>

   ...

</model
```

This automatically makes all fields for the com.some.place.Entity have a *display-width* of 200.

Date Appearance Customization

If you're using the liferay-ui taglib, you've probably noticed that dates also show a time selector. What if you just wanted to show a delta of years, such as starting from 70 years from today? To do this is easy, just make a change similar to the following.

```
<model name="com.some.place.Entity">
   <field name="birthday" type="Date">
```

```
      <hint name="year-range-delta">70</hint>
      <hint name="show-time">false</hint>
   </field>

   ...

</model>
```

Disable String Auto Escaping

Perhaps you are storing something that should not be auto escaped into the field. This can also be disabled by using a hint. A typical example is storing XML data or HTML fragments. The field can be setup as shown below.

```
<model name="com.some.place.Entity">
   <field name="xmldata" type="String">
      <hint-collection name="CLOB" />
      <hint name="auto-escape">false</hint>
   </field>

   ...

</model>
```

Forcing Strings to Uppercase

Likewise, you can force a String to be uppercase by setting the *upper-case* hint as shown below.

```
<model name="com.some.place.Entity">
   <field name="code" type="String">
      <hint name="upper-case">true</hint>
   </field>
```

```
    ...

</model>
```

Future Enhancements

It is foreseeable that further enhancements might be desirable so that not only String fields can be customized this way. Therefore; for reference, the code involved in using the hints and generating the SQL is shown in the table below.

Liferay Version	Class	Method	
<4.4	com.liferay.portal.tools.ServiceBuilder.java	_createSQLTables()	
>4.4	com.liferay.portal.tools.servicebuilder.ServiceBuilder.java	_createSQLTables()	

0 Attachments | 26050 Views

Average (6 Votes)

★★★★☆

Comments

Threaded Replies	Author	Date
I want to comment on this part : "You can also...	Christianto Sahat	March 19, 2009 1:05 AM
liferay should be renamed into: YACF (yet...	Fox Mulder	November 10, 2011 5:58 AM

Christianto Sahat

I want to comment on this part : "You can also use ext-model-hints.xml to overwrite the portals default hints". This solution is not working when I try it. Have you tried this before ? You have to define the entity again in service.xml. But the problem is, it would generate entity classes and interfaces in ext environment, and re-create hibernate mapping in ext-hbm.xml and spring configuration in ext-spring.xml. So after running the service builder, I have to delete all the interfaces and classes, ext-hbm.xml, ext-spring.xml, just to have model-hints.xml and portal-tables.sql generated. Anyone know how to solve this ?

Sign in to vote.

Posted on 3/19/09 1:05 AM.

Fox Mulder

liferay should be renamed into: YACF (yet another configuration-ext file) ... plus they are all over the place. honestly, why can i not do this kind of stuff inside service xml where i actually configure database columns. this file is not even in the same directory

Sign in to vote.

Posted on 11/10/11 5:58 AM.