Solution: Field-Level Validation

In this lab, you will make several configuration changes in order to improve the field-level validation in TrainingApp.

Requirements

This lab requires that you use TrainingApp 8.0, Guidewire Studio 8.0, and a supported web browser. To view, edit, and delete various contacts, log in to TrainingApp as Alice Applegate. The default URL for TrainingApp is <http://localhost:8880/ab/ContactManager.do>. The login/password for Alice Applegate is aapplegate/gw.

1. Field-level validation in the UI

In ABContactDetailsCompanyDV, for contacts of the type ABAutoRepairShop, there is the AutoRepairLicense field. You will modify several widget properties to aid user input into the field, ensuring only valid license values.

Part A

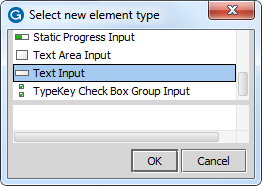
In this part of the exercise, you will configure the AutoRepairLicense field to show the user a water mark.

Investigation

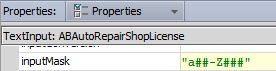
1. View the current widget behavior
2. In Guidewire Studio, debug the server.
3. In TrainingApp, search for Burlingame Saab.
4. In the side bar menu, select Details.
5. Click Edit.
6. Enter an alphanumeric value in the AutoRepairLicense field.
7. Click Update.
8. Open the PCF in Guidewire Studio
9. To open the PCF, use ALT+SHIFT+E.

Configuration

1. Modify the License widget to show a watermark
2. In the ABContactDetailsCompanyDV.pcf file, change the ABAutoRepairLicense widget to be an input widget of the type that supports field-level validation.



1. Modify the PCF file so that the AutoRepairLicense field shows the following water mark:
   * a##-Z###



"a##-Z###"

Verification

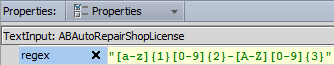
1. Verify the modified widget behavior
2. In TrainingApp, deploy your changes to the PCF file.
3. Edit the Details for Burlingame Saab.
4. For the AutoRepairLicense field,
5. Enter a12-Z345 and verify that there is no field format warning.
6. Click Update.
7. Click Edit. Enter ABC12345 and verify that there is a mouse over field format warning:
   * Field must be a valid value in the format of "a##-Z###"
8. Click Update.

Part B

In this part of the exercise, you will configure the AutoRepairLicense field to match the pattern of the field watermark.

Configuration

1. Modify the License widget to validate the input so that it matches the watermark
2. In the ABContactDetailsCompanyDV.pcf file, continue to modify the ABAutoRepairLicense widget.
3. Modify the PCF file so that the AutoRepairLicense field matches the pattern of the watermark:
   * The first character must be a lower case alphabet character.
   * The second and third characters must be numbers.
   * The fourth character is a dash.
   * The fifth character must be an upper case alphabet character.
   * The sixth, seventh and eighth characters must be numbers.



"[a-z]{1}[0-9]{2}-[A-Z][0-9]{3}"

Verification

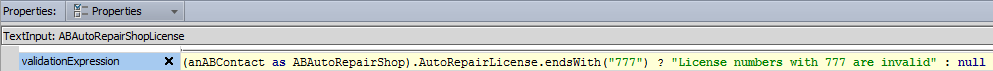
1. Verify the modified widget behavior
2. In TrainingApp, deploy your changes to the PCF file.
3. Edit the Details for Burlingame Saab.
4. Verify that license field value is ABC12345. Click Update.
5. Verify the info bar error message:
   * "License: The value in this field is invalid"
6. Enter u89-R847 as a valid license field value. Click Update.
7. Verify that the you were able to commit the changes.

Part C

In this part of the exercise, you will configure the AutoRepairLicense field to validate the value before it is committed. If the license number contains the string "777", then the user should be notified that license number is invalid.

Configuration

1. Modify the License widget to
2. In the ABContactDetailsCompanyDV.pcf file, continue to modify the ABAutoRepairLicense widget.
3. If the license number contains the string "777", then the user should be notified that license number is invalid.



(anABContact as ABAutoRepairShop).AutoRepairLicense.endsWith("777") ? "License values that contain 777 are invalid" : null

Verification

1. Verify the modified widget behavior
2. In TrainingApp, deploy your changes to the PCF file.
3. Edit the Details for Burlingame Saab.
4. Change the license field value to r8-U777.
5. Click Update.
6. Verify the info bar error message:
   * "License: License numbers with 777 are invalid"
7. Enter a12-Z345 as a valid license field value. Click Update.
8. Verify that the you were able to commit the changes.
9. Field-level validation in the Data Model

In this exercise, you will implement field-level validation in the data model.

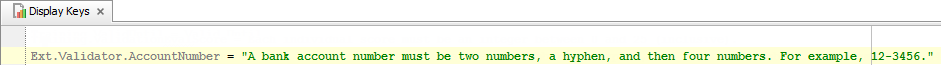
Requirements

Acme Insurance requires that all bank account numbers follow the pattern XX-XXXX, where X is a digit from 0 through 9. Acme Insurance wants to enforce this pattern in the data model. Without needing to modify existing widget properties in the application, Acme Insurance requires that there is a watermark identifying the pattern. Acme Insurance wants to ensure that the validation warning for an invalid bank account value is localizable.

Tasks

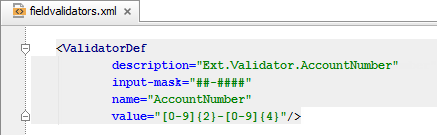
1. Create a validator display key error message
2. Create a display key that reads:

"A bank account number must be two numbers, a hyphen, and then four numbers. For example, 12-3456."



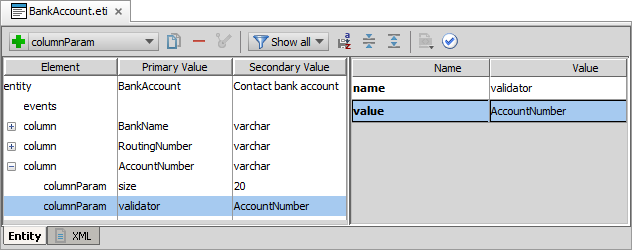
Ext.Validator.AccountNumber = "A bank account number must be two numbers, a hyphen, and then four numbers. For example, 12-3456."

1. Create a field validator
2. Specify the attributes that meet the exercise requirements.



<ValidatorDef   
 description="Ext.Validator.AccountNumber"  
 input-mask="##-####"   
 name="AccountNumber"   
 value="[0-9]{2}-[0-9]{4}"/>

1. Associate the field validator with an entity element
2. Add an entity field validator for the AccountNumber field in the BankAccount entity.



1. Deploy your changes
2. In Guidewire Studio, restart (stop then debug) the server.

Verification

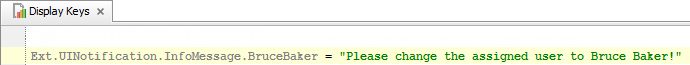
1. Verify the entity field validator behavior
2. In TrainingApp, add a new bank account for the Burlingame Saab contacnt.
3. Verify the ##-#### watermark for the Account Number cell.
4. For the account number, try to enter A1-B2345.
5. Verify that the cell turns red and clears out the invalid field value.
6. Enter 12-3456 as the account number. Click Update.
7. Confirm that a new bank account was added for Burlingame Saab.
8. User Interfacte Notification

In this exercise, you will display a UI notification when the users clicks the Select Least Busy User button on the Summary page.

Requirements

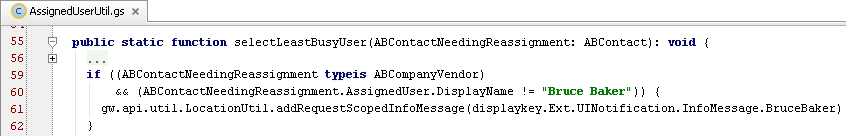
Whenever a user clicks the Select Least Busy User button, Acme Insurance wants the user to see a message if certain conditions are met. When the contact is of the type ABCompany and the AssignedUser field is not Bruce Baker, Acme Insurance wants a UI notification to appear that reads: "Please change the assigned user to Bruce Baker!"

1. Create a display key
2. Create a display key for the UI notification that reads "Please change the assigned user to Bruce Baker!".



Ext.UINotification.InfoMessage.BruceBaker = "Please change the assigned user to Bruce Baker!"

1. Modify the selectLeastBusyUser method
2. In Guidewire Studio, if the server is not already running, debug the server.
3. Open the AssignedUserUtil.gs class
4. Add an if statement at the top of the selectLeastBusyUser method to displays an information message that reads "Please change the assigned user to Bruce Baker!" only when the ABContactNeedingReassignment object is of the type ABCompanyVendor and the current assigned user is not already Bruce Baker.

   
if ((ABContactNeedingReassignment typeis ABCompanyVendor) && (ABContactNeedingReassignment.AssignedUser.DisplayName != "Bruce Baker")) {  
gw.api.util.LocationUtil.addRequestScopedInfoMessage(displaykey.Ext.UINotification.InfoMessage.BruceBaker)  
}

1. Deploy your changes
2. Reload changed classes.
3. Reload PCFs.

Verification

1. Verify the modified widget behavior
2. Edit the Burlingame Saab contact.
3. Change the Assigned User to Bruce Baker. Click Update.
4. Click Edit and then click Suggest Least Busy.
5. Verify that there is no UI notification.
6. Click Edit and then click Suggest Least Busy.
7. Verify the UI notification is an information message and shows the display key test.

|  |  |
| --- | --- |
|  | Stop and ask your instructor to review your completed lab. |