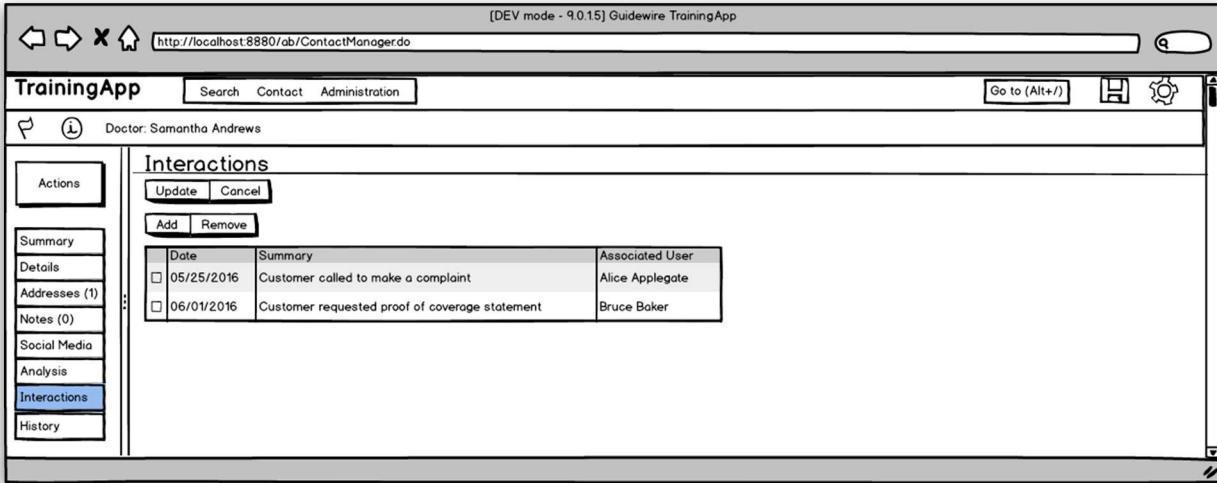


# Lesson 1 Creating New Entities

The business analysts want to capture specific information about each contact interaction such as a telephone call, email message, postal mail letter, or in-person office visit. Eventually, they want the information displayed in an Interactions section of TrainingApp.



The screenshot shows a web browser window for 'Guidewire TrainingApp' (http://localhost:8880/ab/ContactManager.do). The page title is '[DEV mode - 9.0.15] Guidewire TrainingApp'. The main content area is titled 'Interactions' and contains a table with two rows of data. The table has columns for Date, Summary, and Associated User. The first row shows a date of 05/25/2016, a summary of 'Customer called to make a complaint', and an associated user of 'Alice Applegate'. The second row shows a date of 06/01/2016, a summary of 'Customer requested proof of coverage statement', and an associated user of 'Bruce Baker'. On the left side, there is a sidebar with navigation links: Actions, Summary, Details, Addresses (1), Notes (0), Social Media, Analysis, Interactions (which is selected and highlighted in blue), and History.

Date	Summary	Associated User
05/25/2016	Customer called to make a complaint	Alice Applegate
06/01/2016	Customer requested proof of coverage statement	Bruce Baker

*"We want a CSR (Customer Service Representative) such as Alice Applegate to be able to create and edit details about contact interactions."* – Insurance company business analysts

In this lab, your job is to make the necessary **data model changes** to meet customer requirements. As a configuration developer, you will use the Entity Editor in Guidewire Studio to modify the TrainingApp data model. You will implement the **user interface changes** in a later lesson.

## 1.1 Prerequisites

## 1.2 Lab: Create a new entity

As a configuration developer, you want to be able to create new entities. In this lab, you will use the Entity Editor to create an entity to capture contact interaction details.

1. Create a new entity named `Interaction_Ext` and add the following fields:

Field Name	Datatype	Null ok?
InteractionDate	date and time	true
InitiatedByContact	A boolean value	true
Summary	A string of up to 60 characters	true
AssociatedUser	A foreign key to User	true

### 1.2.1 Verification

2. Use Studio's code generation feature to generate the Java class from the entity and verify the generated Java class
  - a) If possible, use incremental code generation
  - b) Verify that there were no errors during code generation
  - c) Open the generated Java class



#### Important!

Read carefully.

3. For each new entity element, remember to set the `nullok` attribute to true if not specifically defined to be false.
4. When required, add an element description and specify column parameters



## Best Practices

Use \_Ext in the entity name

Notice that the name of the entity has an \_Ext suffix. For new entities, Guidewire recommends that the name of the entity should end with \_Ext (or start with Ext\_). This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

## 1.3 Lab: Define an array relationship between two entities

As a configuration developer, you want to be able to define different types of relationships between entities. In this lab, you will define an array relationship between ABContact and Interaction\_Ext:

"An ABContact entity can have zero or many Interactions associated with it." – Insurance company business analysts

### 1. Add an array key field to ABContact

- d) Extend the ABContact entity to define a new array. The array key field should point to Interaction\_Ext. The array name should be descriptive and should follow the recommended naming conventions.

### 2. Use Studio's code generation feature to generate the Java class from the entity and verify the generated Java class

- a) If possible, use incremental code generation
- b) Verify that there were no errors during code generation
- c) Open the generated Java classes and verify that you can see the new fields
- d) Remember, for every data model array the framework automatically generates `addTo<ArrayName>(element)` and `removeFrom<ArrayName>(element)` functions. Verify that you can find those methods in the generated Java class.



## Best Practices

Use \_Ext in entity field name when extending base application entities

For fields that are added to a base application entity, Guidewire recommends that the field name should

end with \_Ext (or start with Ext\_). This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

### 1.3.1 Verification



#### Activity

Verify the work you have done

As a configuration developer, you want to be able to properly deploy new and changed data model resources.

##### 1. Restart the server to deploy the changes

- a) During server restart Studio first runs the code generators, then compiles the project and finally deploys the resources

##### 2. Regenerate the Data Dictionary

- a) Follow the steps to regenerate the Data Dictionary

##### 3. Open the Data Dictionary

- a) In Windows Explorer, navigate to the data dictionary.
- b) Open the data dictionary using your preferred browser.

##### 4. View the Interaction\_Ext entity

- a) Verify each new field and associated datatype.

##### 5. View the ABContact entity

- a) Verify you can see the new array



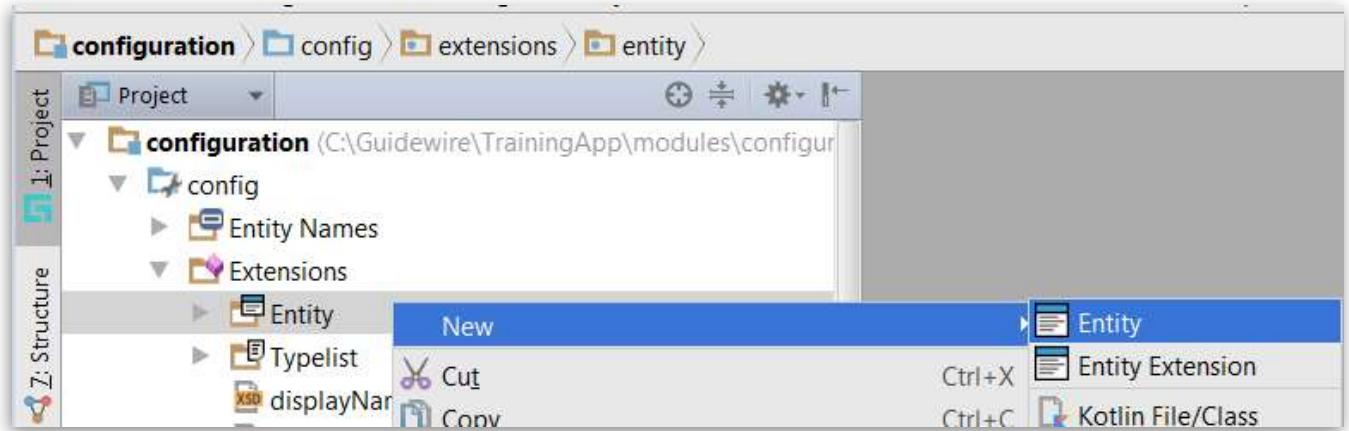
## 1.4 Lab Solution: Create a new entity

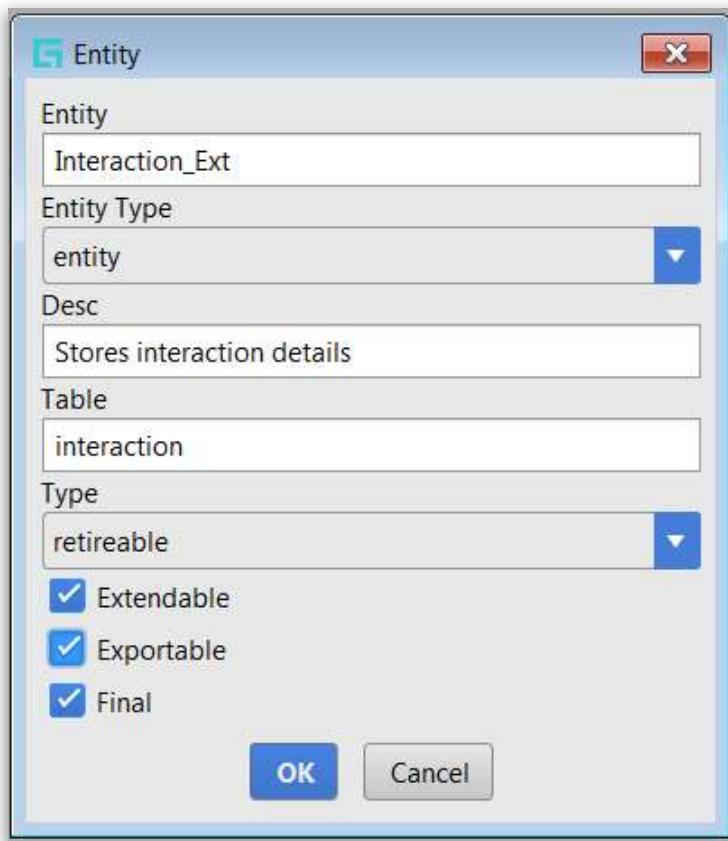


## Solution

Exact details on how to complete the lab.

### 1. Create the Interaction\_Ext entity





## 2. Add new elements to Interaction\_Ext.eti

- Add the InteractionDate field

The Interaction\_Ext.eti element editor shows the following table:

Element	Primary Value	Secondary Value	Name	
entity	Interaction_Ext	Stores interactio...	<b>name</b>	InteractionDate
column	InteractionDate	datetime	<b>type</b>	datetime
			<b>nullok</b>	true

b) Add the InitiatedByContact field

The screenshot shows the Entity Designer interface for the 'Interaction\_Ext.eti' entity. A new column named 'InitiatedByContact' has been added, highlighted in blue. The column is defined as a bit type.

Element	Primary Value	Secondary Value	Name
entity	Interaction_Ext	Stores interactio...	<b>name</b>
column	InteractionDate	datetime	<b>type</b>
column	InitiatedByContact	bit	<b>nullok</b>

c) Add the Summary field and its columnParam

The screenshot shows the Entity Designer interface for the 'Interaction\_Ext.eti' entity. A new column named 'Summary' has been added, highlighted in blue. It is defined as a varchar type with a size of 60. A corresponding columnParam row is also present.

Element	Primary Value	Secondary Value	Name
entity	Interaction_Ext	Stores interactio...	<b>name</b>
column	InteractionDate	datetime	<b>type</b>
column	InitiatedByContact	bit	<b>nullok</b>
column	Summary	varchar	<b>desc</b>
	columnParam size	60	allowInitialValueForUpgrade

d) Add the AssociatedUser field

The screenshot shows the Entity Designer interface for the 'Interaction\_Ext.eti' entity. A new foreign key named 'AssociatedUser' has been added, highlighted in blue. It is associated with the 'User' entity and maps to the 'AssociatedUserID' column.

Element	Primary Value	Secondary Value	Name
entity	Interaction_Ext	Stores interactio...	<b>name</b>
column	InteractionDate	datetime	<b>fkentity</b>
column	InitiatedByContact	bit	<b>nullok</b>
column	Summary	varchar	<b>columnName</b>
foreignkey	AssociatedUser	User	<b>desc</b>

- 3. Entities support incremental code generation. You can click on the validate icon or use the CTRL + S keystroke to kick off incremental code generation. You can open the generated Java class once the code generation is completed.**

**Note:** The code generation was successful if the Codegen tool window has nothing to show.

Screenshot of some of the Java code generated by the Codegen tool.

```

configuration > generated > entity > Interaction_Ext
Interaction_Ext.java x Interaction_Ext.eti x

6  /*
7   * @ajax.annotation.Generated(value = "com.guidewire.pl.metadata_codegen.Codegen", comments = "Interaction_Ext.eti;Interaction_Ext")
8   * /deprecation, unchecked/
9   * @gw.internal.gosu.parser.ExtendedType
10  * @gw.lang.SimplePropertyProcessing
11  * @gw.entity.EntityName(value = "Interaction_Ext")
12  public class Interaction_Ext extends com.guidewire.pl.persistence.code.BeanBase implements entity.Retireable {
13      public static final gw.pl.persistence.type.EntityTypeReference<entity.Interaction_Ext> TYPE = new com.guidewire.commons.
14
15      public static final gw.pl.persistence.type.EntityPropertyInfoReference<gw.entity.ILink PropertyInfo> ABCONTACT_PROP = new
16
17      public static final gw.pl.persistence.type.EntityPropertyInfoReference<gw.entity.ILink PropertyInfo> ASSOCIATEDUSER_PROP
18

```

## 1.5 Lab Solution: Define an array relationship between two entities



Exact details on how to complete the lab.

1. Add the array key to the ABContact.etx

Element	Primary Value	Secondary Value	Name
array	Interactions_Ext	Interaction_Ext	<b>name</b> Interactions_Ext
implements	com.guidewire....	com.guidewire.a...	<b>arrayentity</b> Interaction_Ext
implements	com.guidewire....	com.guidewire.a...	desc All interactions associated with the contact

2. You must also add a foreign key to Interaction\_Ext.eti since every array requires a reverse foreign key on the other side of the relationship.

The screenshot shows the ABContact.etx configuration interface. The 'Interaction\_Ext' entity is selected. A table lists the foreignkey properties:

Element	Primary Value	Secondary Value	Name	
entity	Interaction_Ext	Stores interactio...	<b>name</b>	ABContact
column	InteractionDate	datetime	<b>fkentity</b>	ABContact
column	InitiatedByContact	bit	<b>nullok</b>	false
▶ column	Summary	varchar	<b>columnName</b>	ABContactID
foreignkey	AssociatedUser	User	<b>desc</b>	
foreignkey	ABContact	ABContact	<b>archivingOwner</b>	<i>target</i>

3. Entities support incremental code generation. You can click on the validate icon or use the CTRL + S keystroke to kick off incremental code generation. You can open the generated Java class once the code generation is completed.

Note: The code generation was successful if the Codegen tool window has nothing to show.

```

configuration > generated > entity > Interaction_Ext >
Project: Interaction_Ext.java
public class Interaction_Ext extends com.guidewire.code.BeanBase implements entity.Retireable {
  public static final String TYPE = "entity.Interaction_Ext";
  public static final ILinkPropertyInfo ABCONTACT_PROP = new com.guidewire.commons.metadata.types.LinkPropertyInfoCache(TYPE, "ABContact");
}
  
```

```

configuration > generated > entity > ABContact >
Project: ABContact.java
public class ABContact extends com.guidewire.code.BeanBase implements entity.Retireable {
  public static final String TYPE = "entity.ABContact";
  public static final IColumnPropertyInfo ID_PROP = new com.guidewire.commons.metadata.types.ColumnPropertyInfoCache(TYPE, "ID");
  public static final IArrayPropertyInfo INTERACTIONS_EXT_PROP = new com.guidewire.commons.metadata.types.ArrayPropertyInfoCache(TYPE, "Interactions_Ext");
}
  
```

```

configuration > generated > entity > ABContact >
Project: ABContact.java
public class ABContact extends com.guidewire.code.BeanBase implements entity.Retireable {
  ...
  /**
   * Adds the given element to the Interactions_Ext array. This is achieved by setting
   * the element to the array.
   */
  public void addToInteractions_Ext(entity.Interaction_Ext element) {
    __getInternalInterface().addArrayElement(INTERACTIONS_EXT_PROP.get(), element);
  }
}
  
```

configuration > generated > entity > ABContact >

entity\ABContact.java

```
/*
 * Removes the given element from the Interactions_Ext array. This is achieved by mark
 */
public void removeFromInteractions_Ext(entity.Interaction_Ext element) {
    _getInternalInterface().removeArrayElement(INTERACTIONS_EXT_PROP.get(), element);
}
```