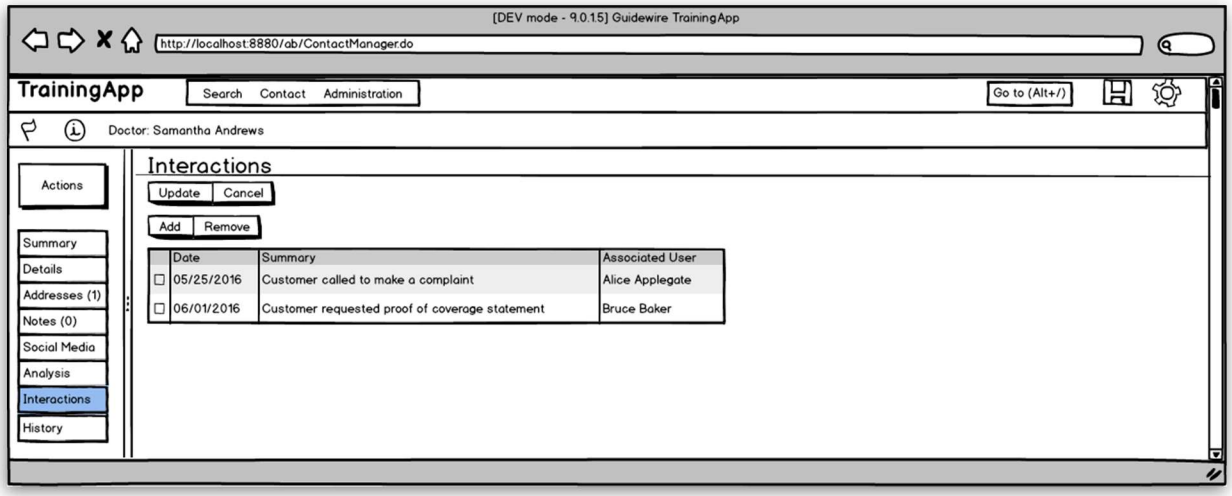
Lesson 1 Creating New Entities

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.



"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.

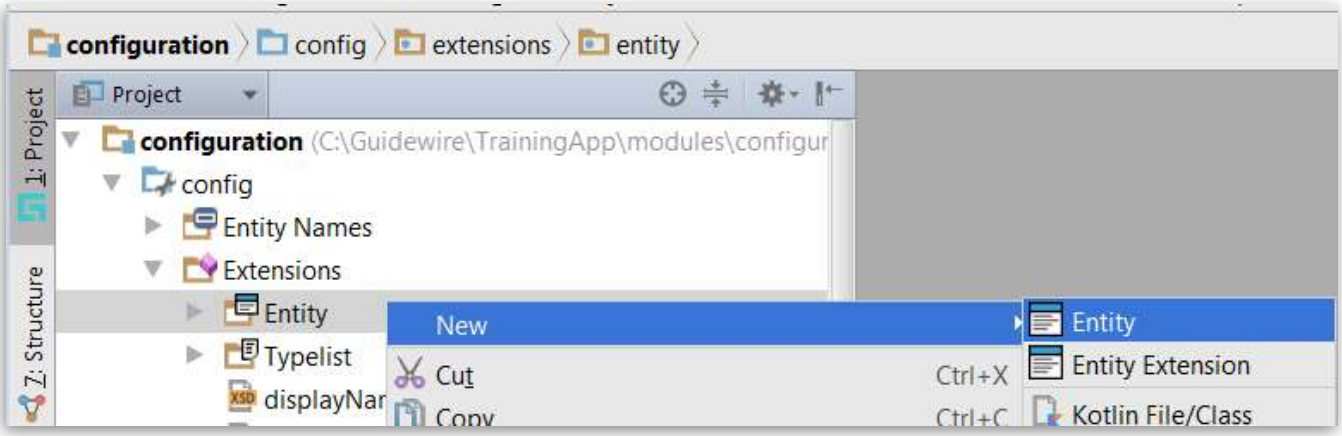
1. **Restart the server to deploy the changes**
   a) During server restart Studio first runs the code generators, then compiles the project and finally deploys the resources
2. **Regenerate the Data Dictionary**
   a) Follow the steps to regenerate the Data Dictionary
3. **Open the Data Dictionary**
   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

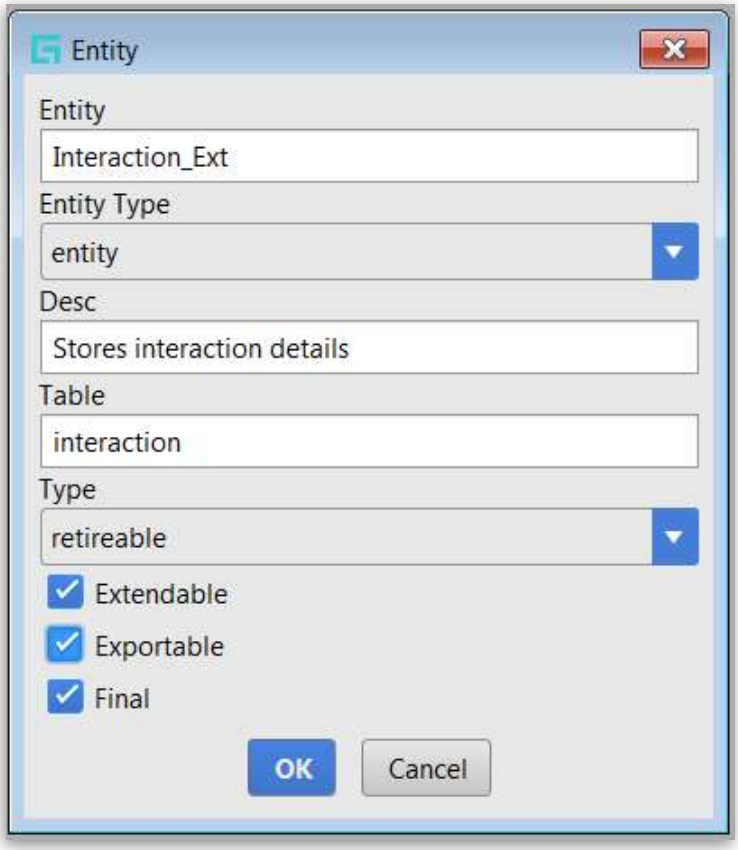
1.4 Lab Solution: Create a new entity

Solution

Exact details on how to complete the lab.

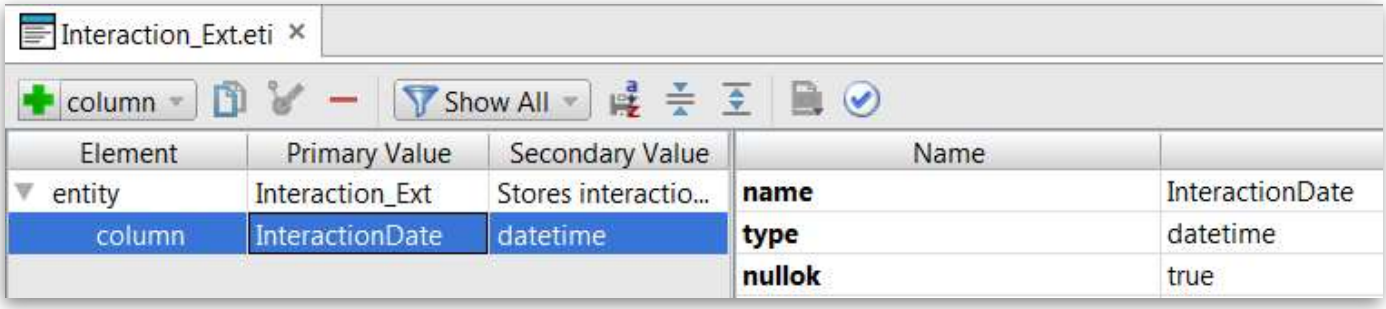
1. Create the Interaction_Ext entity



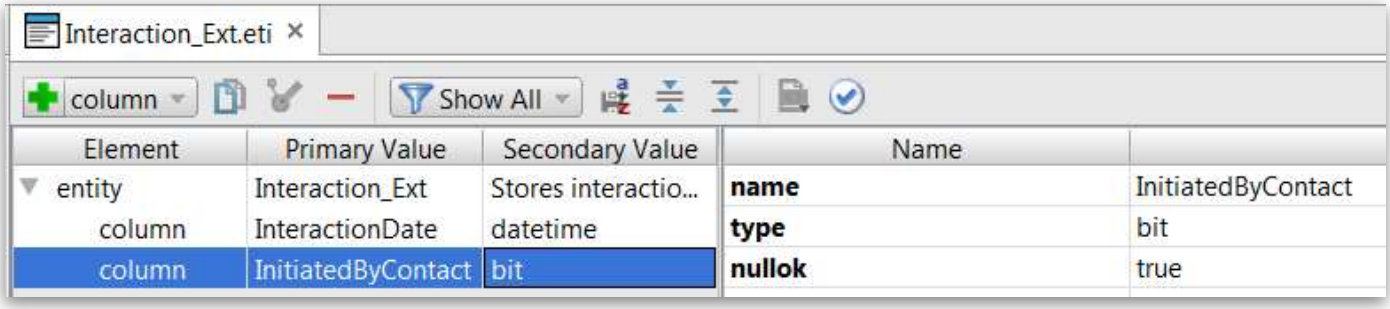


2. Add new elements to Interaction_Ext.eti

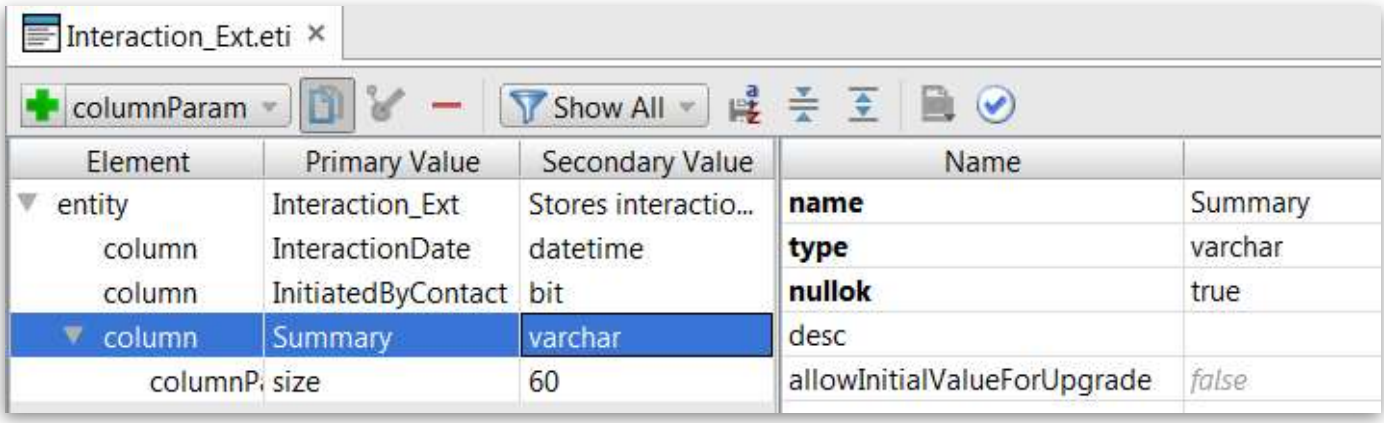
a) Add the InteractionDate field



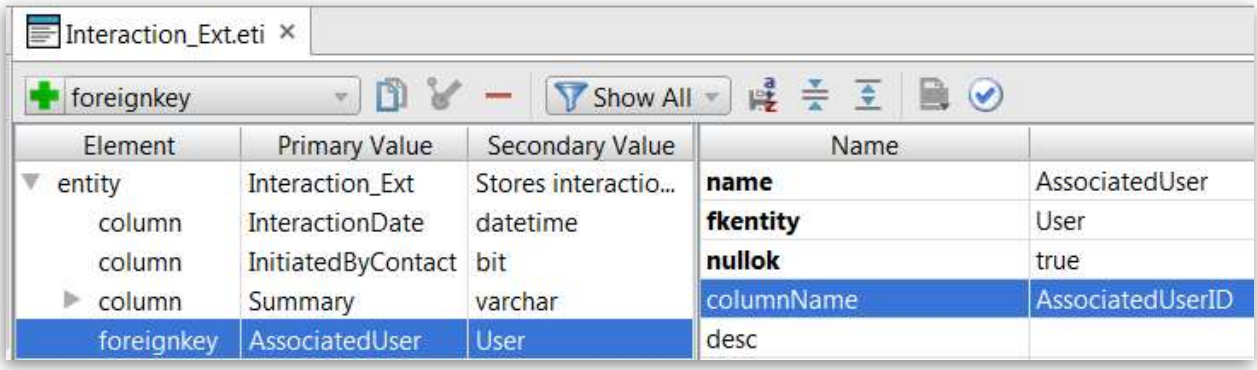
b) Add the InitiatedByContact field



c) Add the Summary field and its columnParam



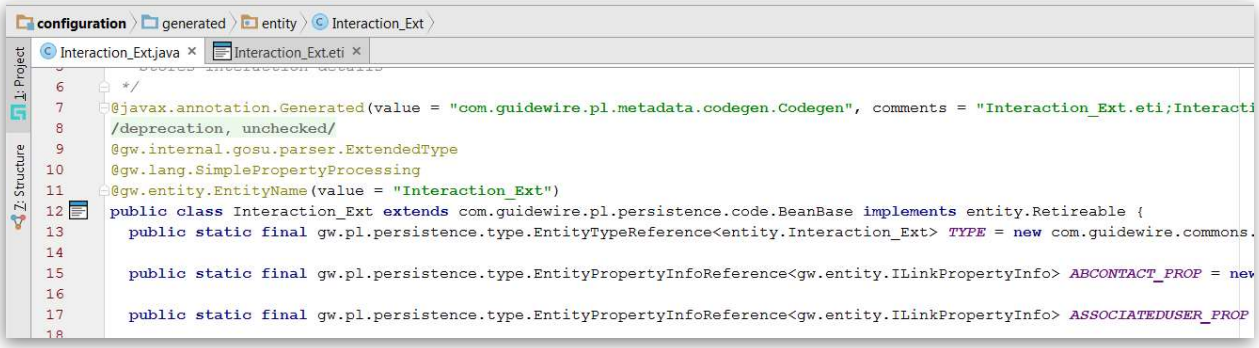
d) Add the AssociatedUser field



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.

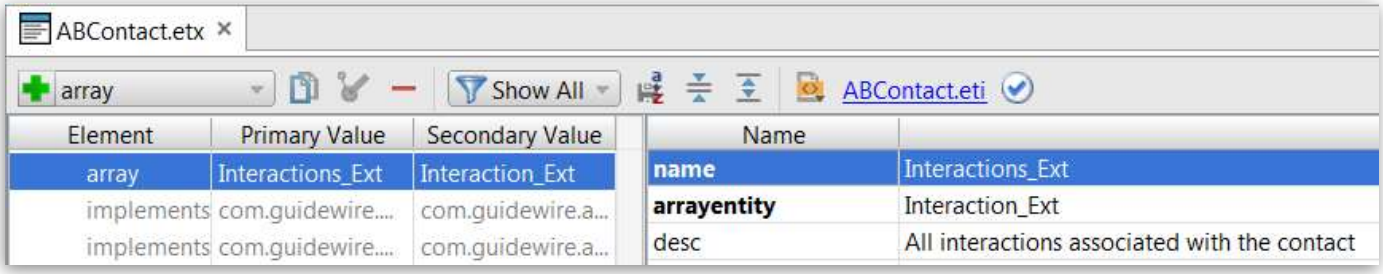


1.5 Lab Solution: Define an array relationship between two entities

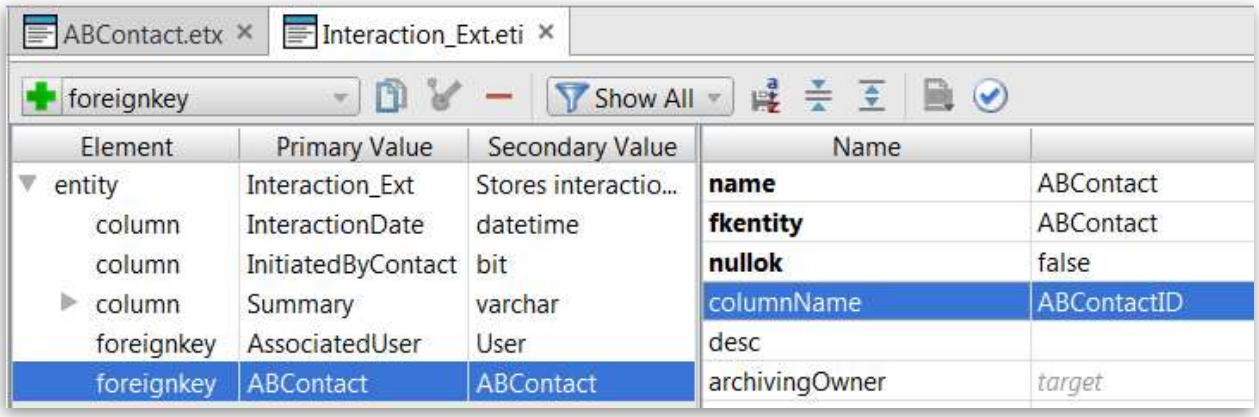
Solution

Exact details on how to complete the lab.

1. **Add the array key to the ABContact.etx**

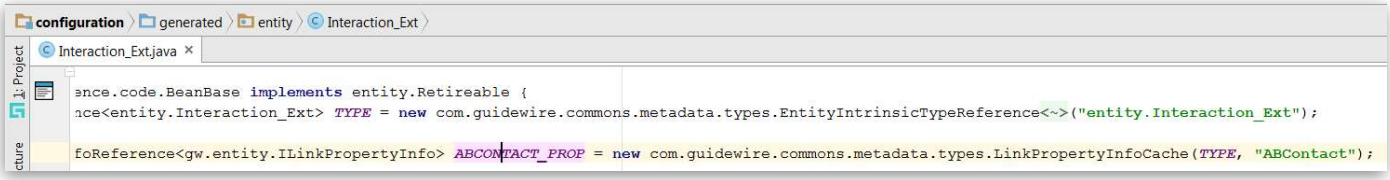


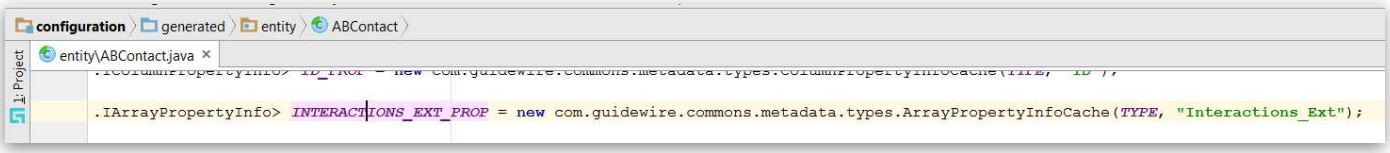
2. **You must also add a foreign key to Interaction_Ext.eti since every array requires a reverse foreign key on the other side of the relationship.**

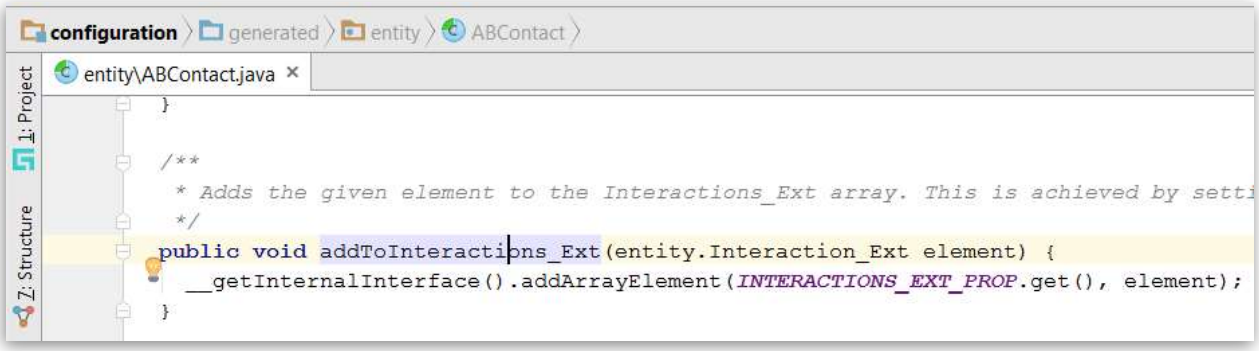


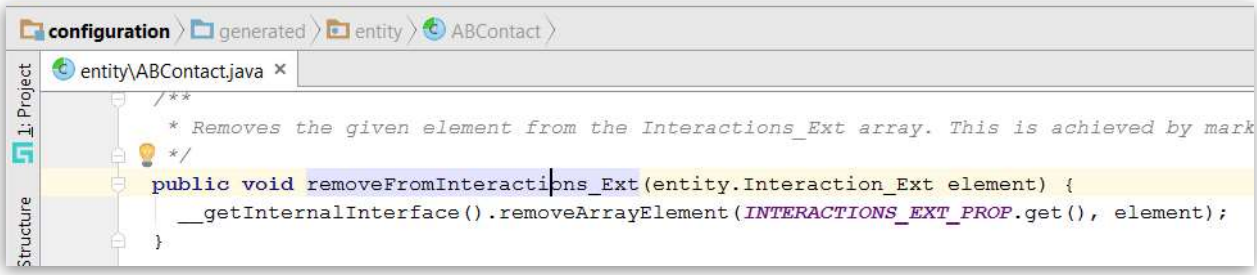
3. **Entities support incremental code generation. You can click on the validate icon or use the CTRL + S keystroke to kick off incremental code generation. You can open the generated Java class once the code generation is completed.**

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








configuration
generated
entity
ABContact
entity\ABContact.java
1: Project
Structure
/**
* 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);
}