**Declarative Rules: Condition Rules**

**Declarative Rule are defined under Decision Category**

**Advantage: No need to call rule anywhere**

**Disadvantage: We cannot restrict the rule from Calling**

**Declarative Rules gets executed automatically without calling. Whenever properties are used/Updated & any data table value changes**

**Types of Declarative Rules:**

**Declare Expressions: We can perform calculations on properties**

**Declare Constraints: Restrict a Property to have specific value( Dependency Validations)**

**Declare Onchange: Based on Change in Property Value it will automatically calls Onchange Activity**

**Declare Trigger:  Based on any Table Value Changes it will automatically calls Trigger Activity**

**Declare Index: Used to fetch Page List property Values from the Index Table**

=================

**Working with Declare Expression:**

**This rule can be used to assign a value to target property based on same calculation on the source property.**

**Ex:**

**A, B, C**

**Addition: C= A+B**

**C - Target Property ( Output)**

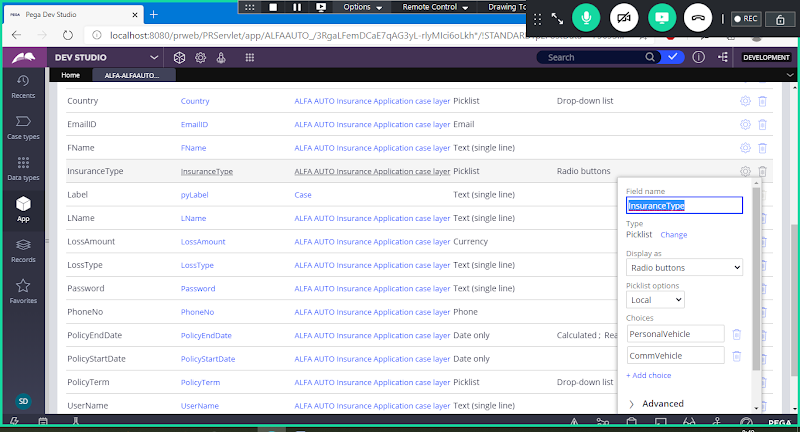
**A, B - Source Properties ( Input Values)**

**Whenever input property values gets updated declare expression rule automatically gets executed.**

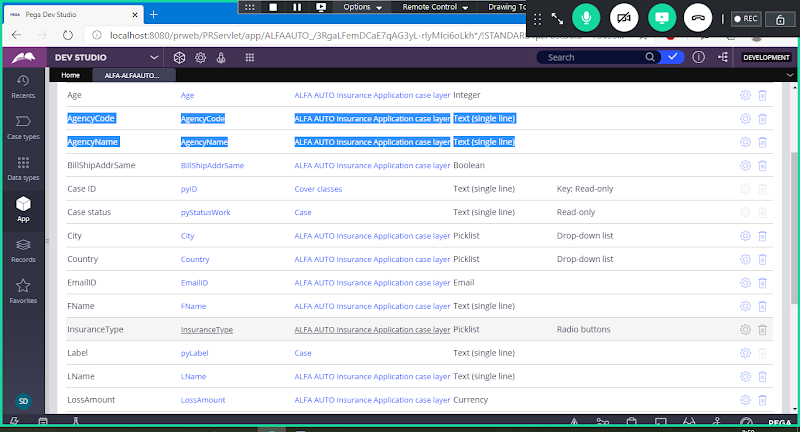
**Expressions are Defined on Target Properties. The Value for Target property will be Read Only.**

**Req: Create Property &  Sections**

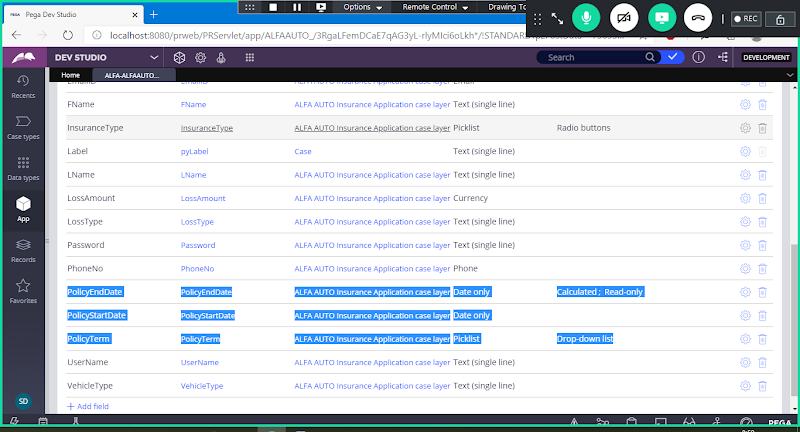
**Create Property ( Insurance Type with Radiobutton( Personal Vehicle & CommVehicle)**



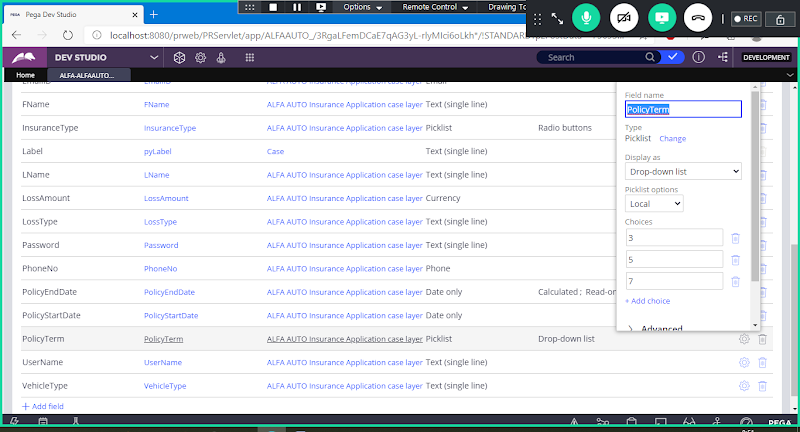
**Properties  -   Ageny Code , Agency Name)**



**Properties  --( PolicyStartDate Calender, PolicyEndDate, PolicyTerm)**



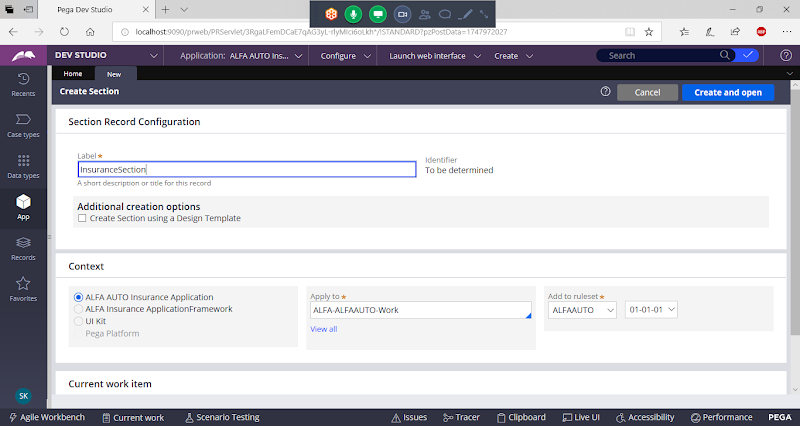
**For Policy Term Use Dropdown & Add Static Records**

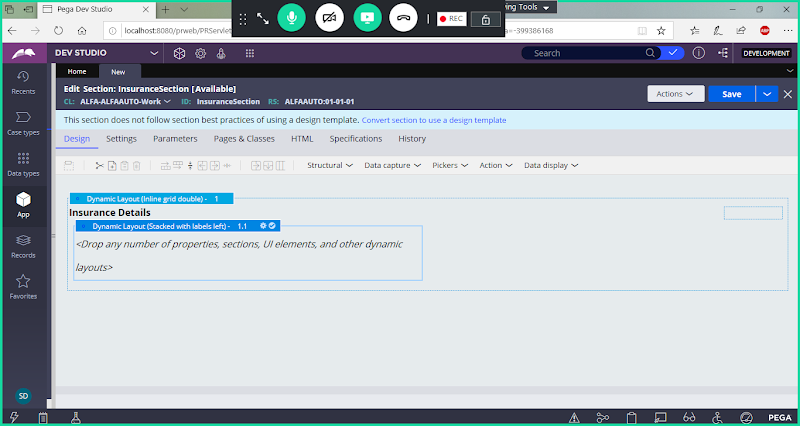


**==============**

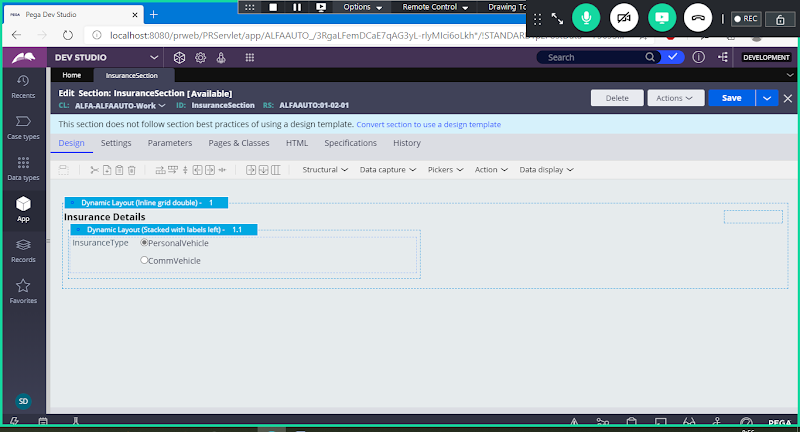
**Create sections:**

**Insurance Section:**



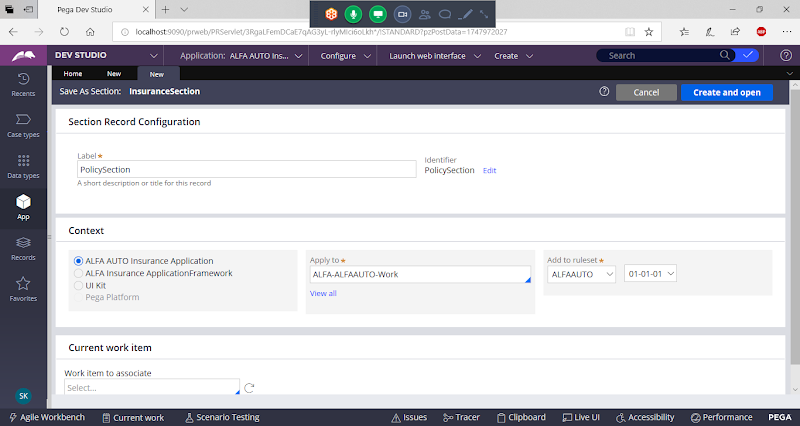


**Drag & drop Single Value Property**

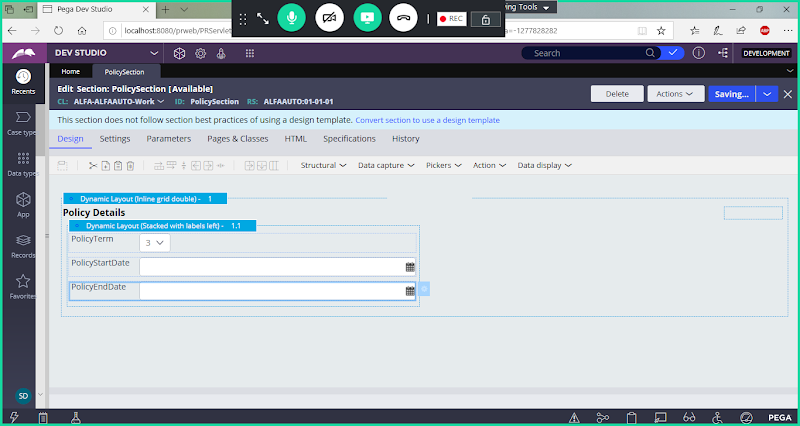


**Save**

**Policy Section:**

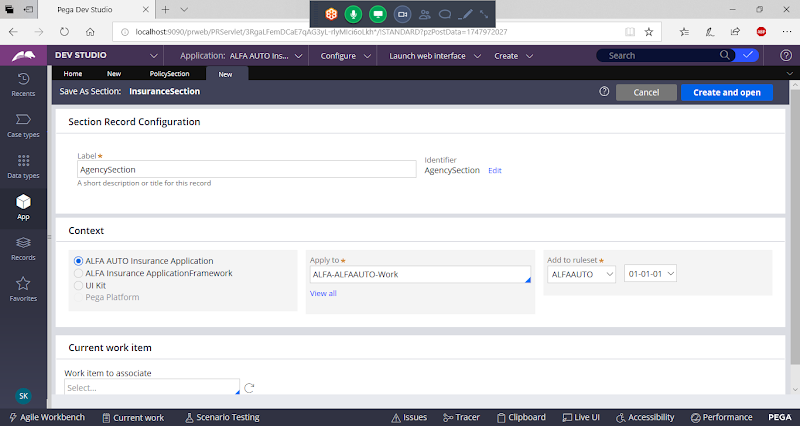


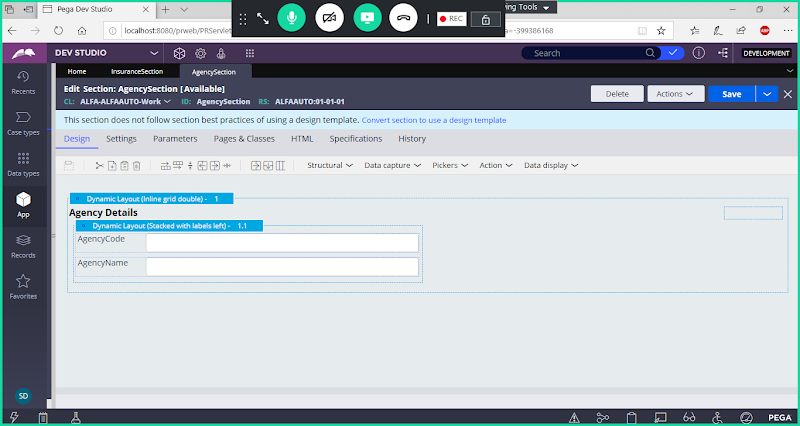
**Drag & Drop Single Value Properties**



**Save section**

**Agency Section:**



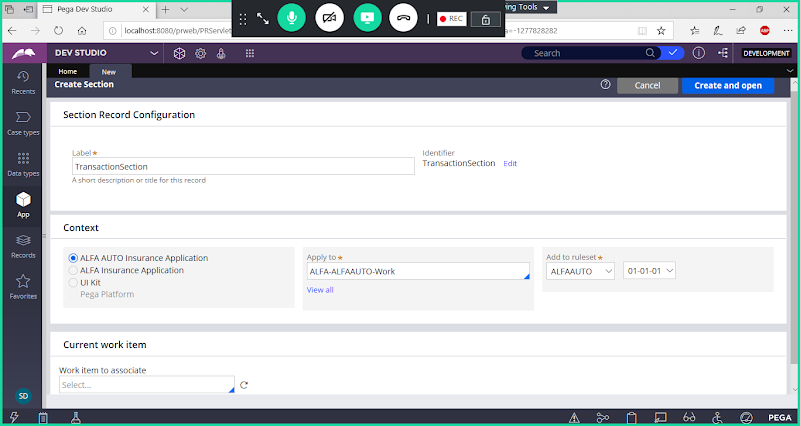


**Save Section**

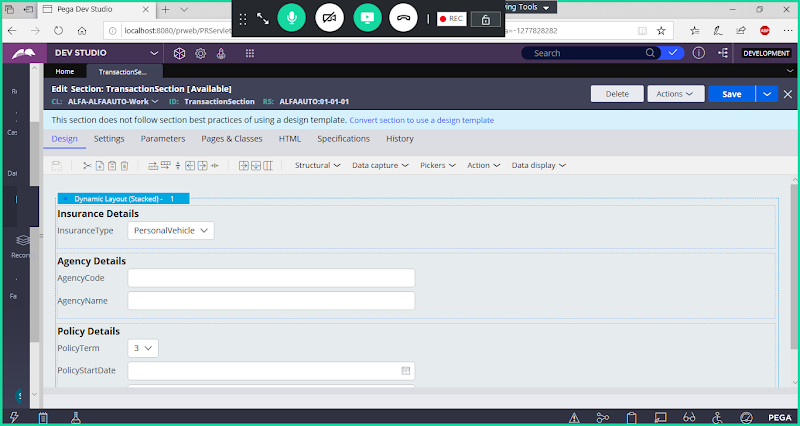
**=====**

**Add Main Section ( Transaction Section)**

**Add Transaction Section & Drag & Drop Above sections:**

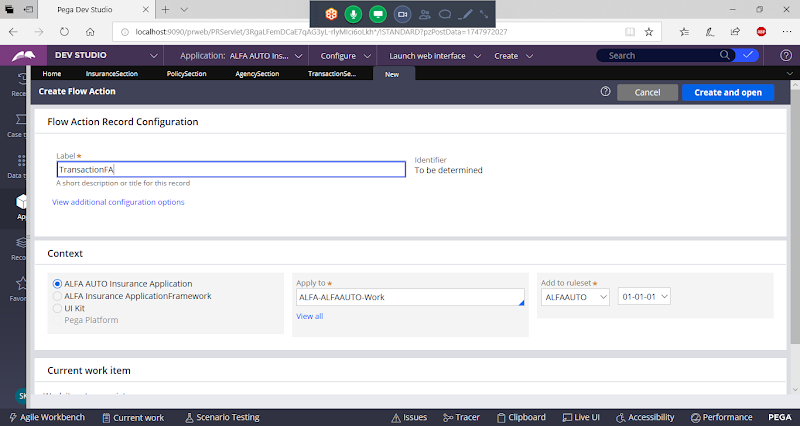


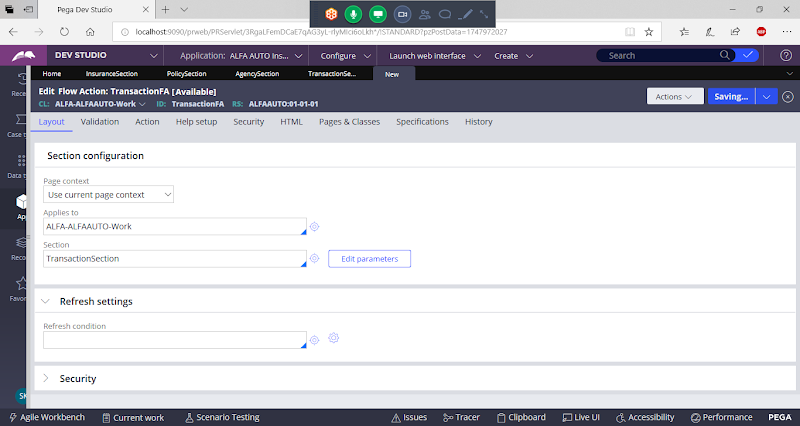
**Drag & Drop Sections**



**Save section**

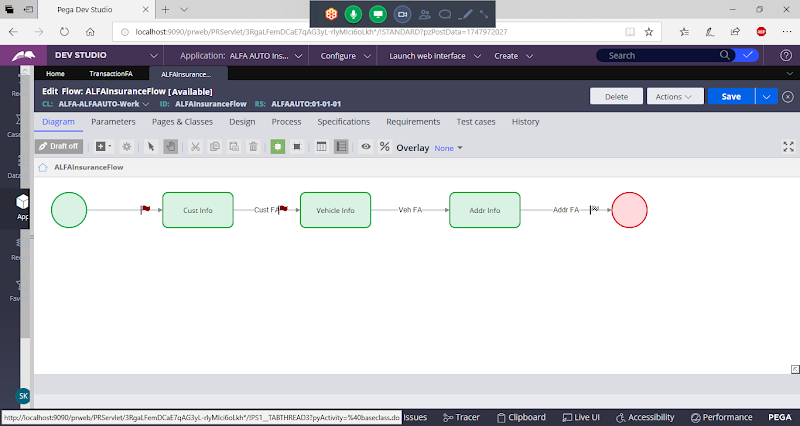
**Add Flowaction ( Transaction\_FA)**

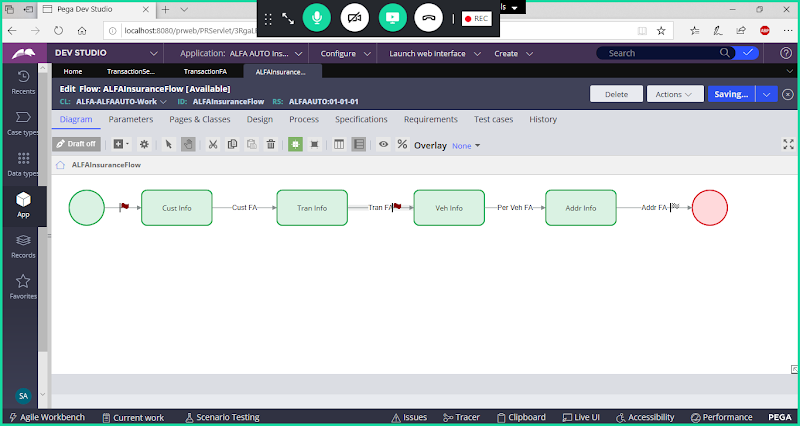




**Save**

**Now Map Flowaction in Insurance Flow. Add New Assignment after Customer Details & Call this Flowaction.**





**Add Transaction Flowaction & Save Flow.**

**Declare Expressions:**

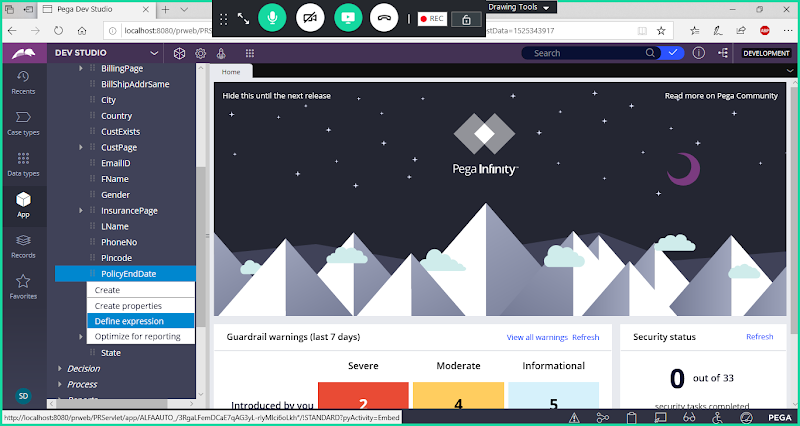
**Req: Add Declare Expression Rule for PolicyEndDate, such that when Policy Term & PolicyStartDate are given it should Calulate PolicyEndDate & display Value in Read only Format**

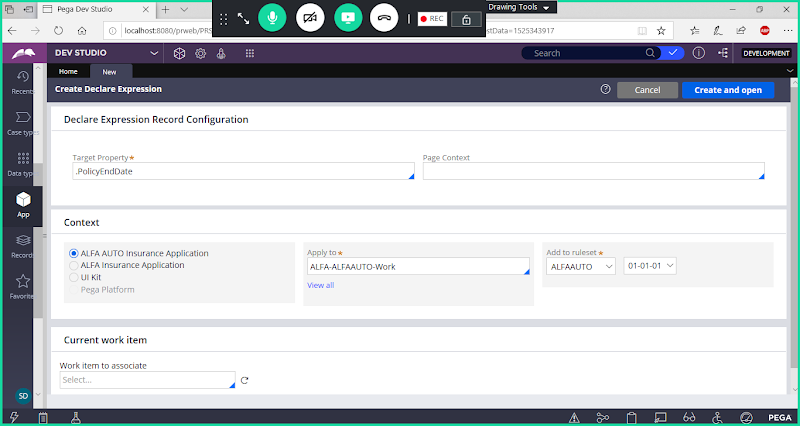
**Source Properties: Policy Term, PolicyStart Date**

**Target Property: PolicyEnddate**

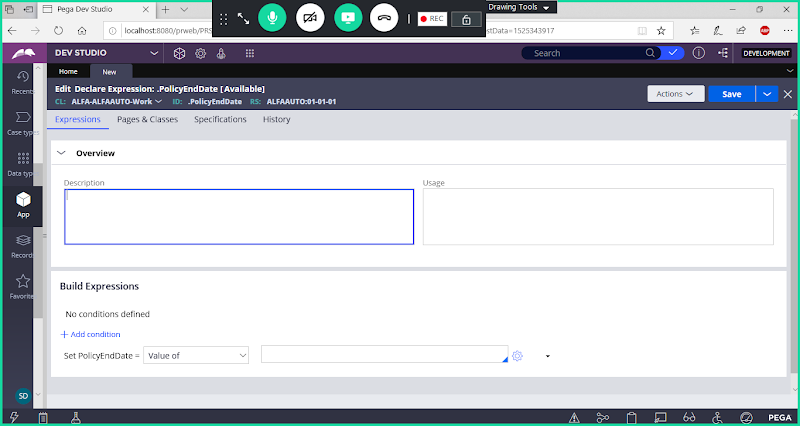
**Create Declare Expression Rule:**

**Right Click on PolicyEndDate Property & Select Define Expression:**

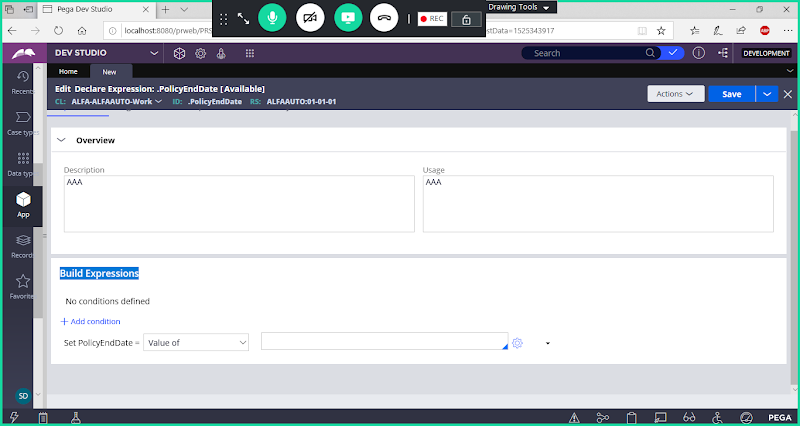




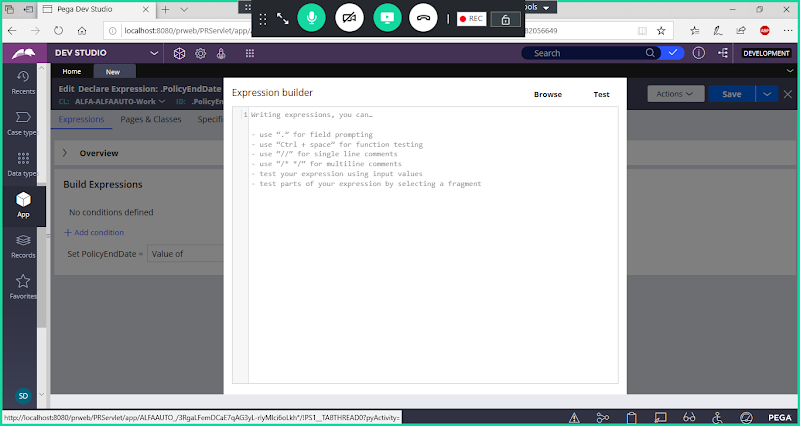
**Open**



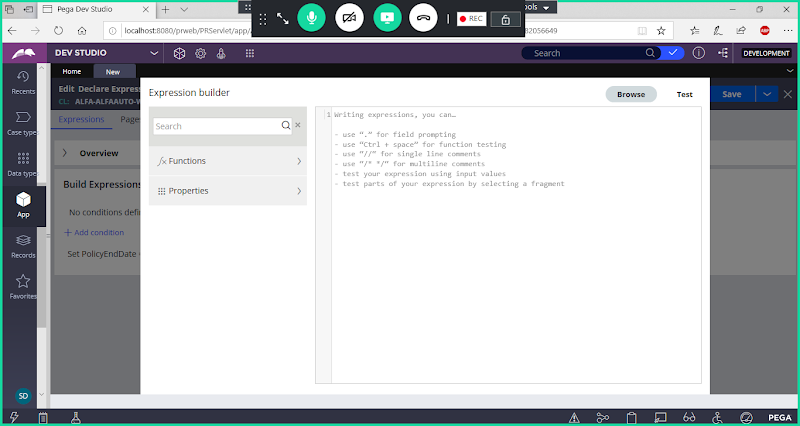
**Add Description & Usage**



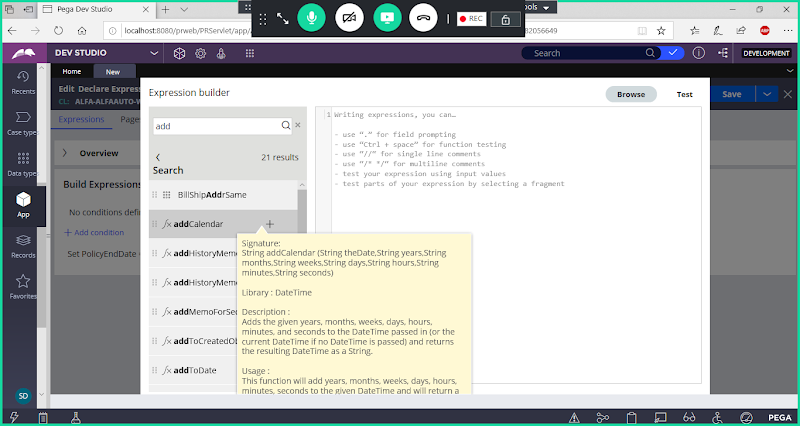
**Open Expression Builder**

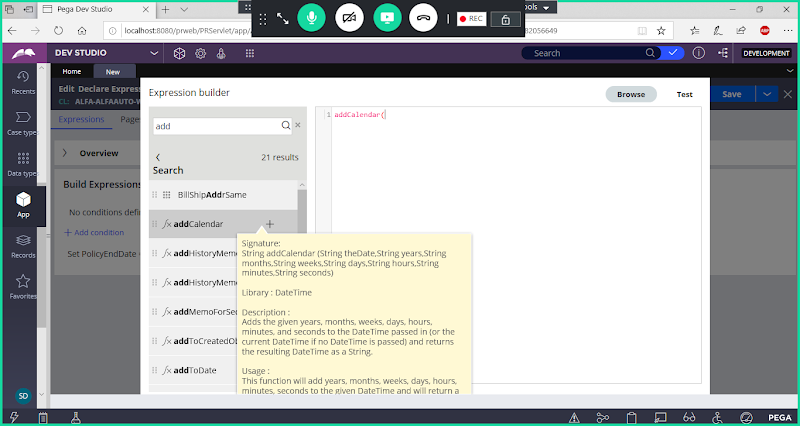


**Click on Browse**

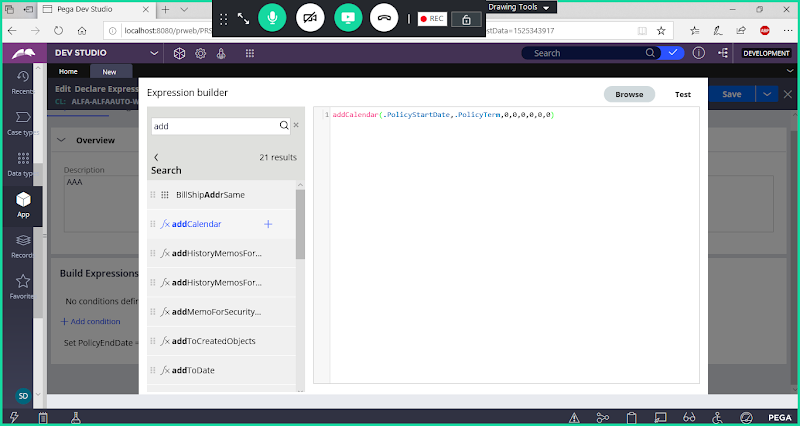


**Function: addCalender**



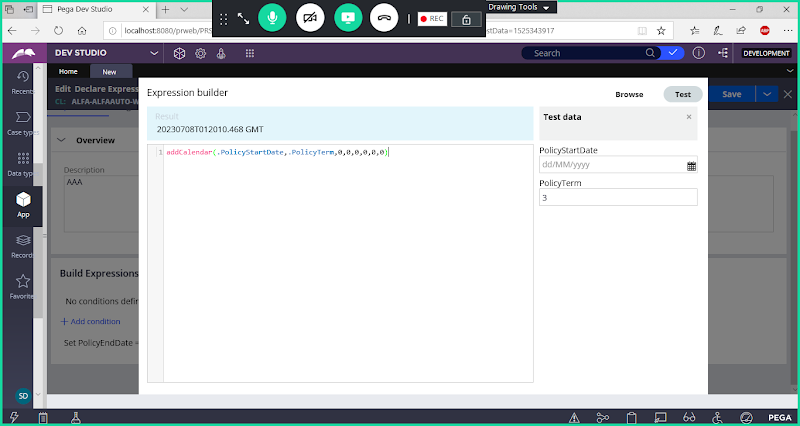


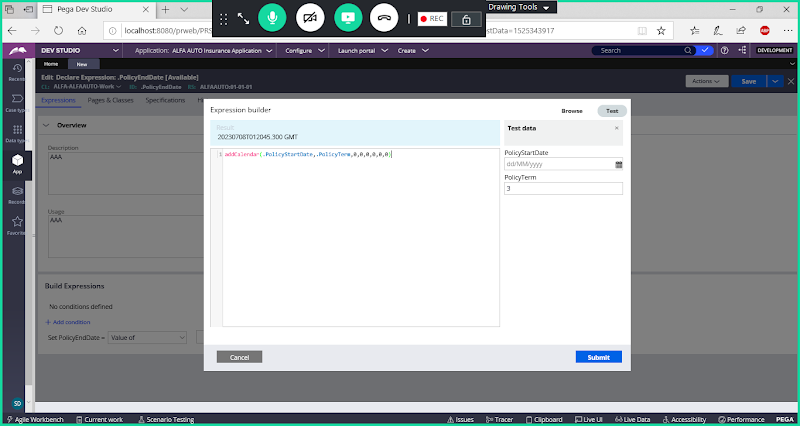
**According to Function Parameters fill your Property Names**



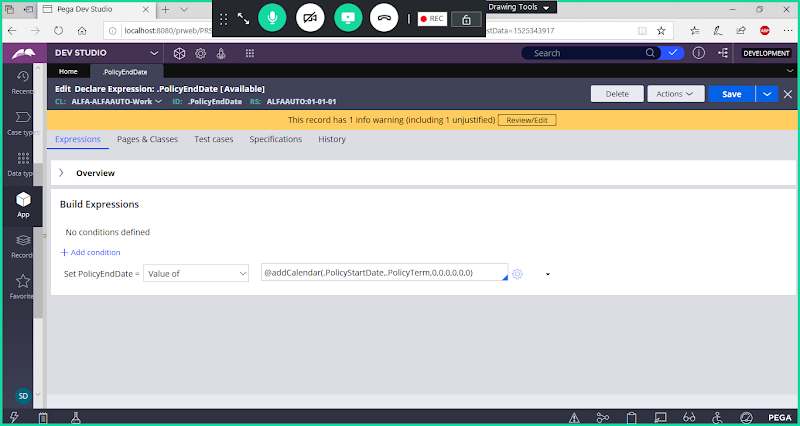
**addCalendar(.PolicyStartDate,.PolicyTerm,0,0,0,0,0,0)**

**Click on Test Button**



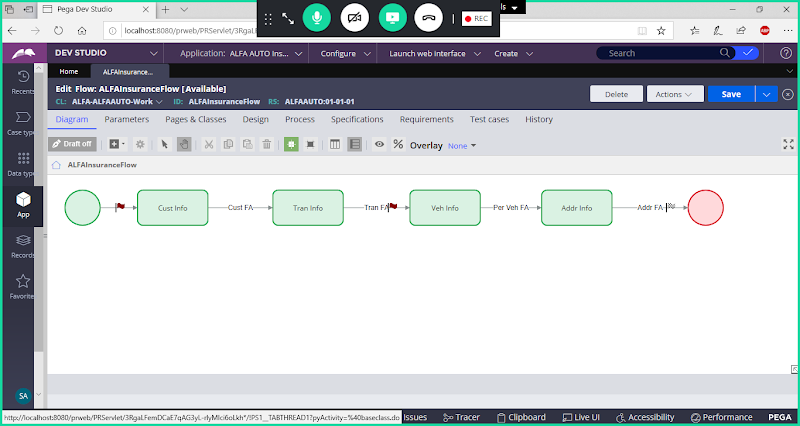


**Submit**

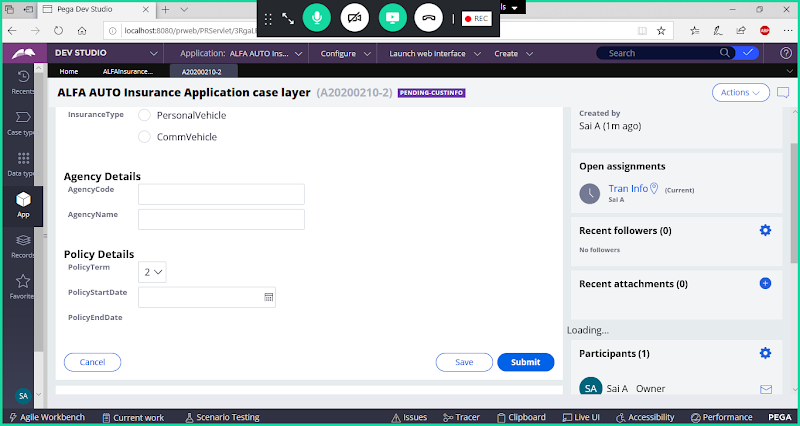


**Save**

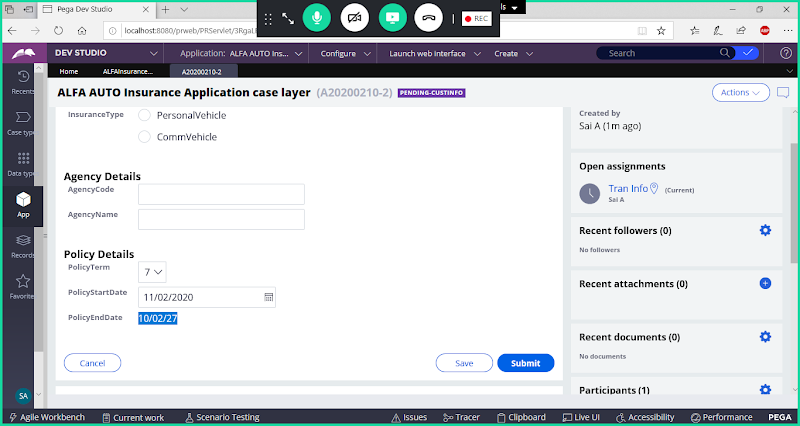
**Open Insurance Flow:**



**Run the flow**



**Add Source for the Policy Start date & Policy Term**



**====================**

**What is Forward Chaining & Backward Chaining?**

**Forward Chaining:**

**Ex: C= A +B**

**A & B - Source Properties**

**C - Target Property**

**If Inputs A & B Changes then C Value Changes.**

**It is called as Forward Chaining**

**Backward Chaining:**

**Ex: F=C+D**

**Here C Value is Retrieving from Data Storage, where value is already saved to data base table.**

**If Target Value if you use in another expression as source it is called as Backward Chaining**

**It is called a Backward Chaining**

**Technical Definitions:**

**Forward Chaining** :- Forward chaining is an internal mechanism that provides the automatic propagation of changes in one property value to changes in other property values.

In a Declare Expression rule, constraints rule, or Declare Index rule, you can establish required relationships among properties. When an activity step changes the value of a property that is defined in such rules, the system automatically consults an internal dependency network to see if other values are affected, and performs property value computations. This is known as forward chaining.

**Backward Chaining**  : - In Backward Chaining Process Commander allows a computation to advance even when the value of an input or parameter property is not available.

Declare Expression rules and the Property-Seek-Value method work together to provide this capability. The system uses the internal dependency network of property relationships to develop inferences about how to obtain the missing property value.

**==========**