**Working with Declare Trigger Rule:**

**Declare Trigger is a Declarative created from Decision Category.**

**When an Action is performed on tables, it will call Trigger Activity. We can Set Activity Type as Trigger.**

**Declare Trigger Rule gets fired when a table operation is performed**

**Trigger Actions can be - Calling Activity ( Send Email Notifications), Generate Reports.**

**Declare Trigger gets executed based on DB Value Changes i.e Insert, Update or Delete**

**Declare Trigger we will define under Table Class. For every Action we create separate trigger Rule.**

**Insert Action - Trigger**

**Update Action- Trigger**

**Delete Action- Trigger**

**Page - Group:**

**Page List Structure:**

**Address (1)                     Address (2)**

**DNo                                  DNo**

**Area                                 Area**

**City                                  City**

**Pincode                          Pincode**

**Page Group Structure:**

**Address (Home)                     Address (Office)             Address( Permanent)**

**DNo                                          DNo                                  DNo**

**Area                                         Area                                  Area**

**City                                          City                                    City**

**Pincode                                  Pincode                            Pincode**

**In Page Group structure based on Unique record the search will happen.**

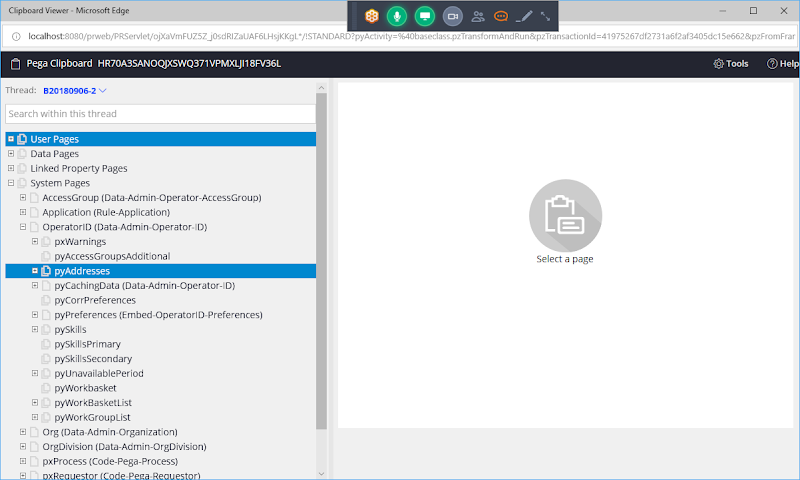
**In Page Group advantage is instead of Index, we can identify directly using name or Value.**

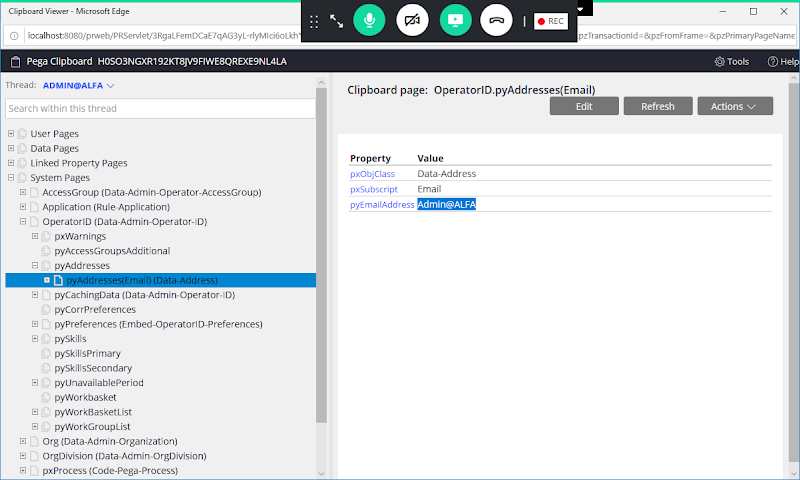
**If Page list we have to loop to search for index ( It will automatically Indexes)**

**Some predefined page group properties in pega: pyAddresses( Email), pyAddresses( fax),pyAddresses(Phone)**

**What is Page Group Structure of Email Property:**

**Go to Clipboard-> System Pages-> Operator ID-> pyAddresses**





**Fetch Multiple Email Id: Obj-Browse**

**OperatorsList.pxResults(<Current>).pyAddresses(Email).pyEmailAddress**

**Obj-Open:**

**XPage.pyAddresses(Email).pyEmailAddress**

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

**Req: Customize Manager Portal left Menu to a New Menu ( Add New State).**

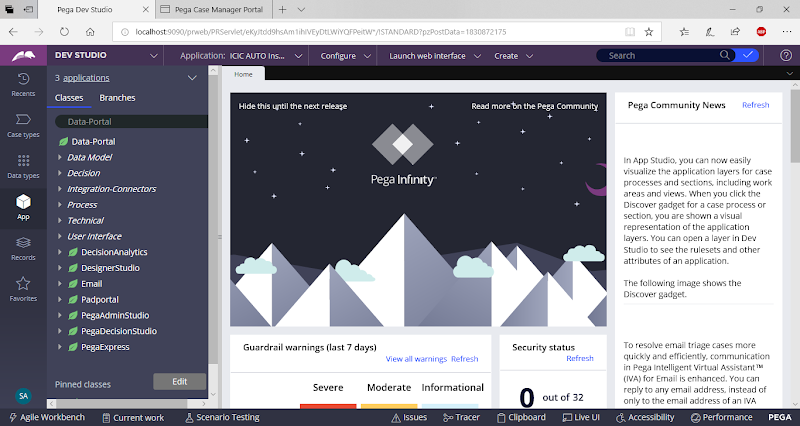
**- Add New Menu in Manager Portal. On click of add new state display page with State Code & State Name Properties & Add Button to insert Data.**

**- On submit, values should be inserted to States Table**

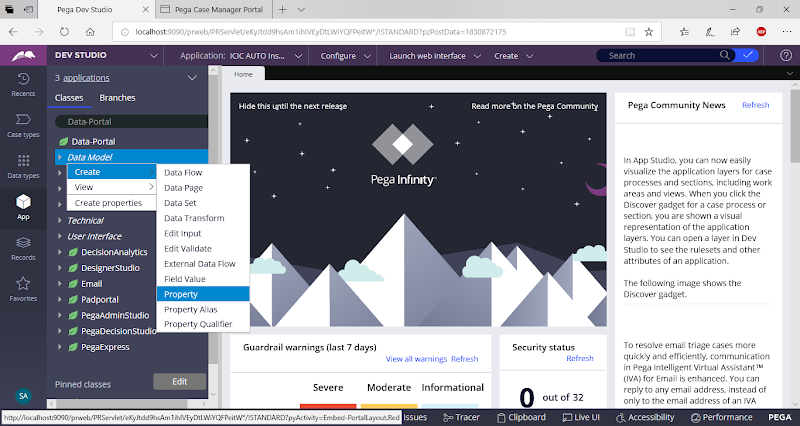
**- Once State is inserted an email notification should be sent to all the managers of our application.**

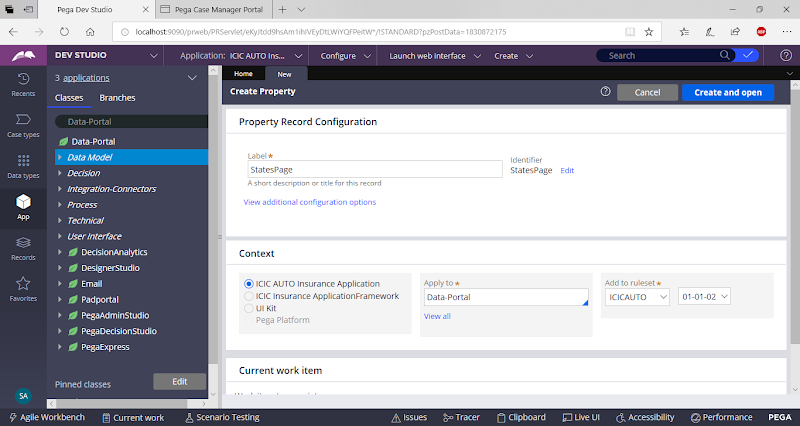
**- When Customizing any Portal Options any new rules we are going to add must be defined under Data-Portal Class.**

- **When customizing Manager portal anything we create add under Data-Portal Class.**

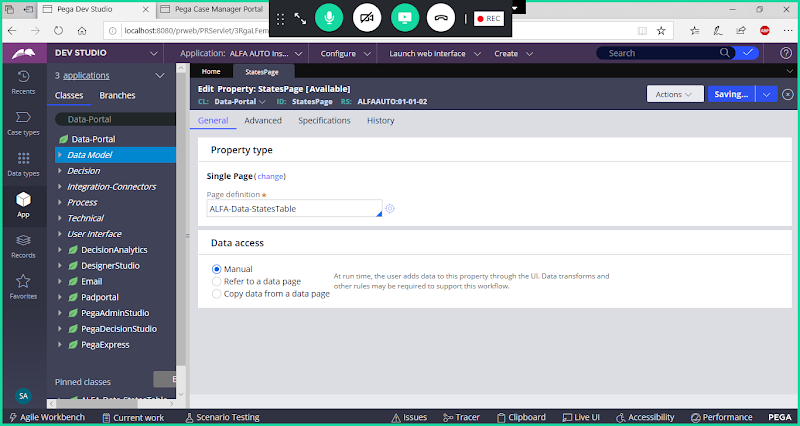


**Create a Page Property Property referring to StatesTable Class:**



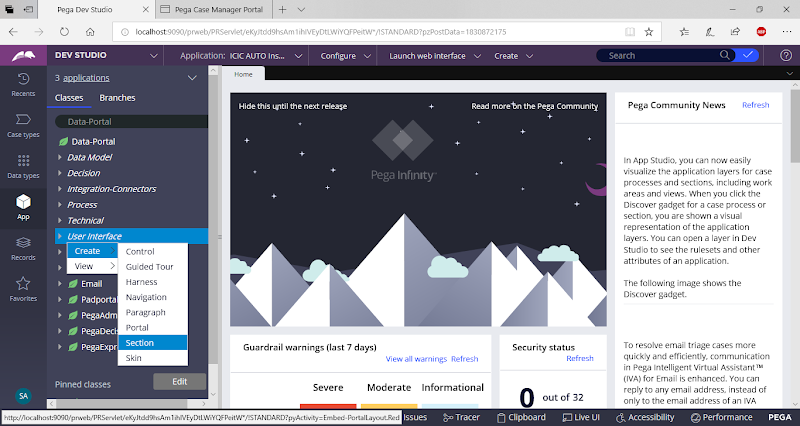


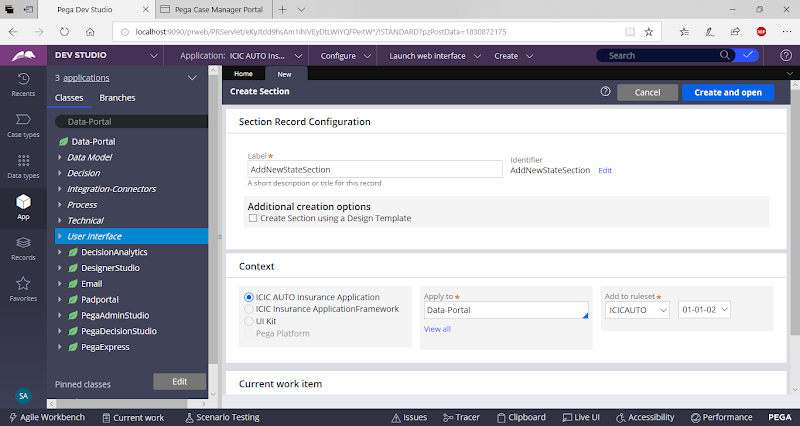
**open**



**Save Property**

**Create a Section under Portal Class:**

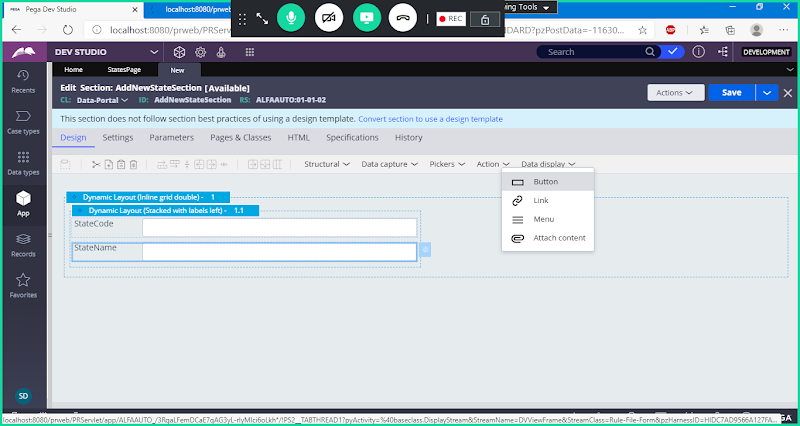


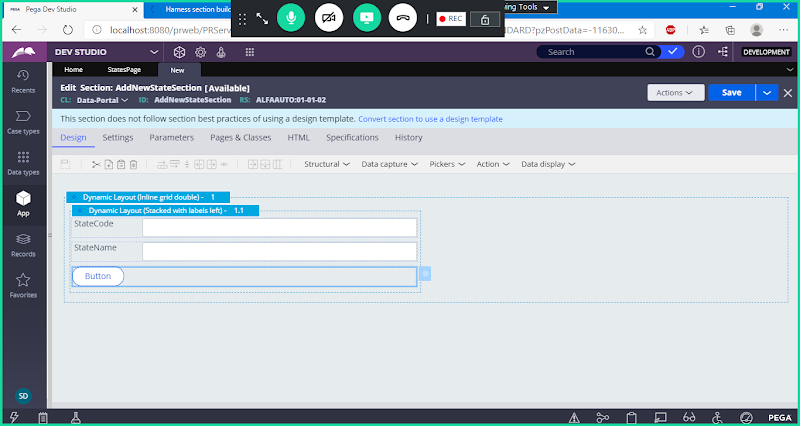


**open**

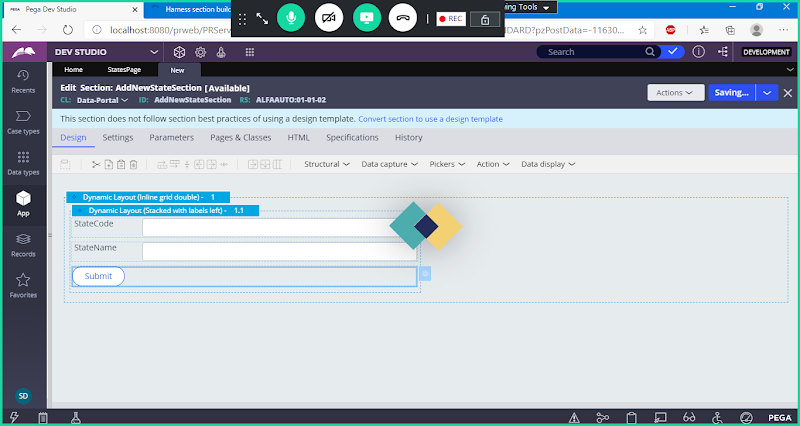


**Add Button Control from Action dropdown**



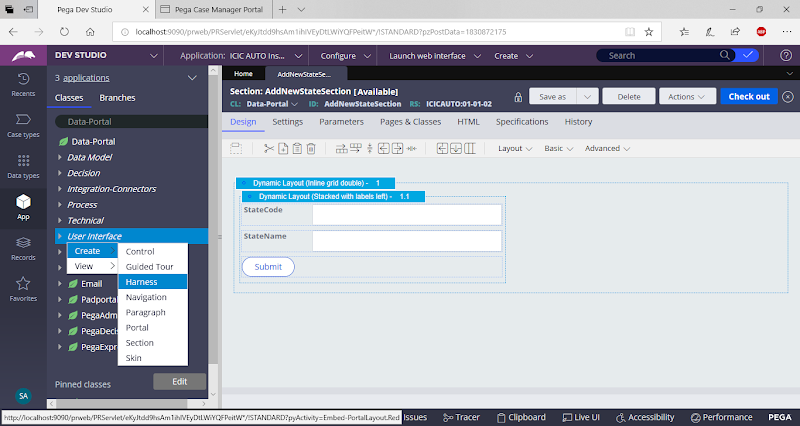


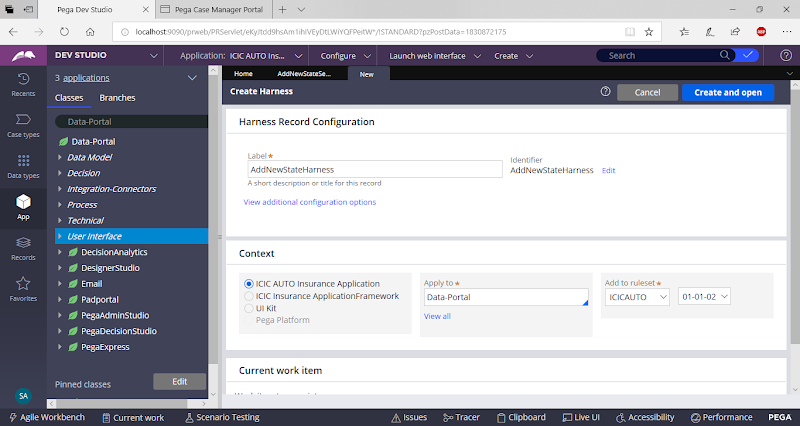
**Add Label for Button**



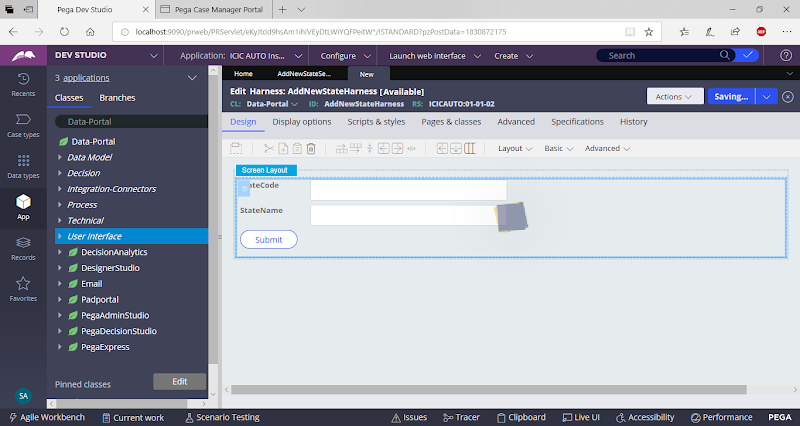
**Save section**

**Create a Harness Rule:**



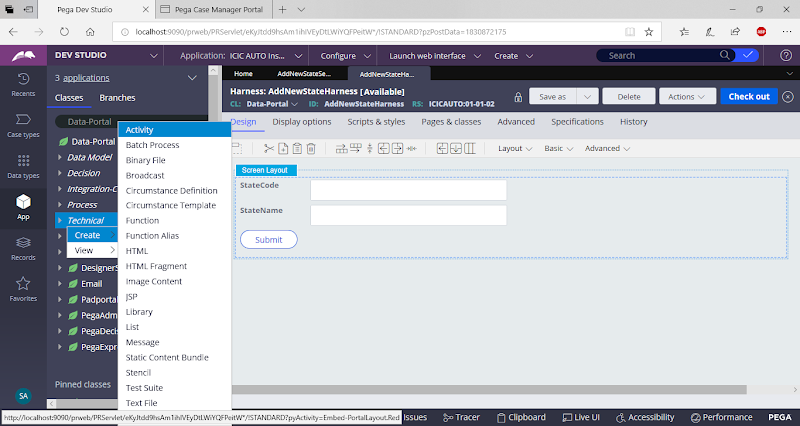


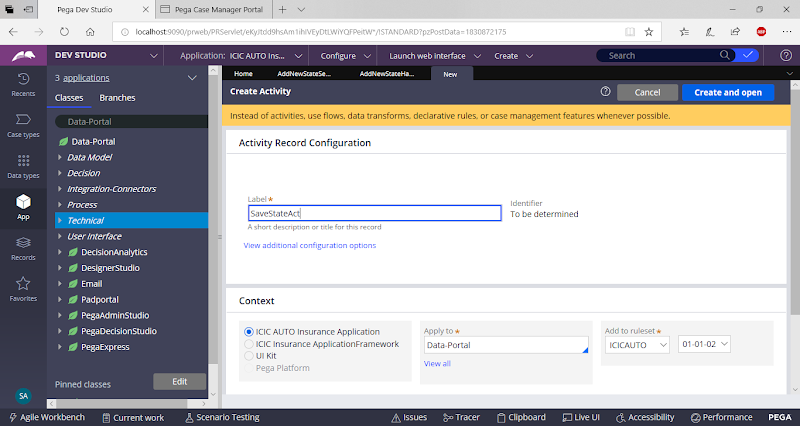
**Drag & Drop Section in Harness**



**Save Harness**

**Create Activity to insert New State Data to States table, add activity under Data-Portal Class.**

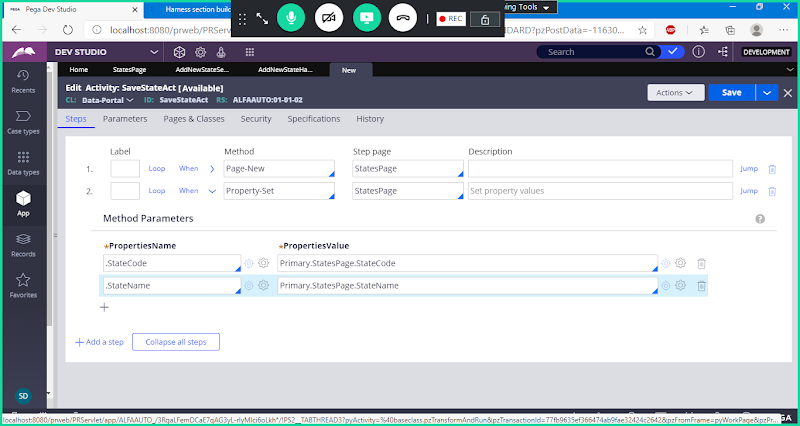


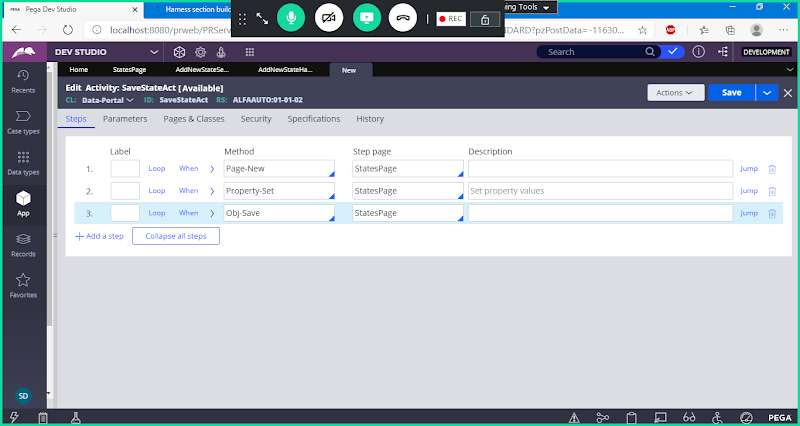


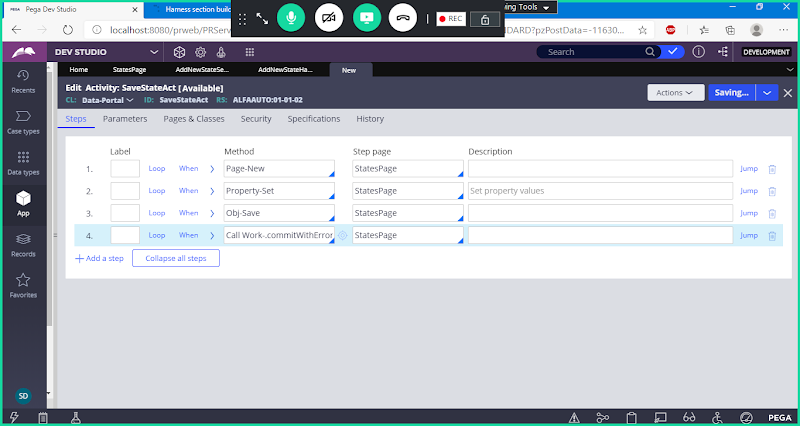
**Go to Pages & Classes**



**Go to Steps Tab:**



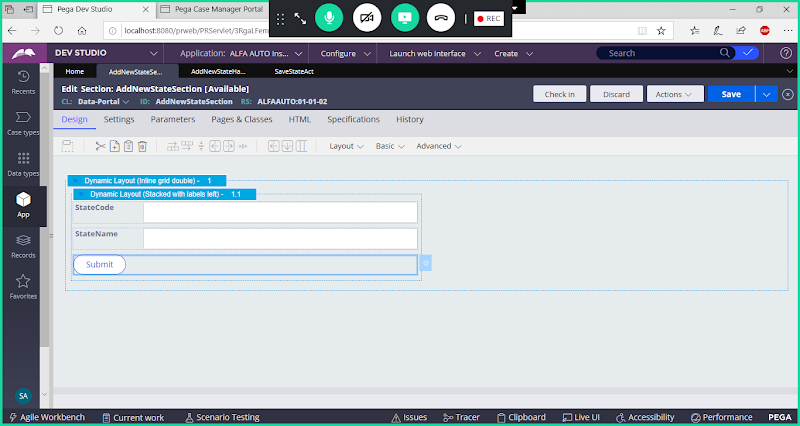




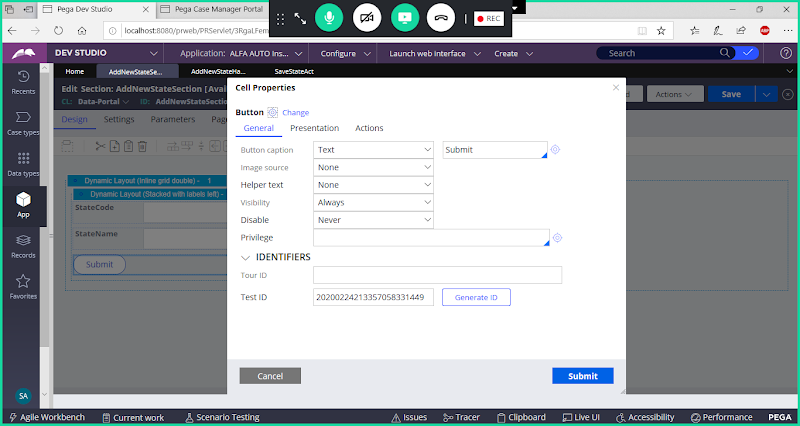
**Call Work-.commitWithErrorHandling**

**Save Activity**

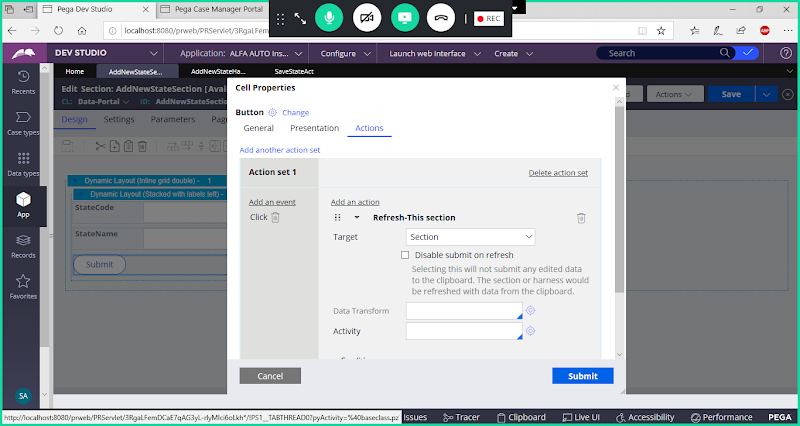
**Call this Activity in Button settings in Add New State Section:**



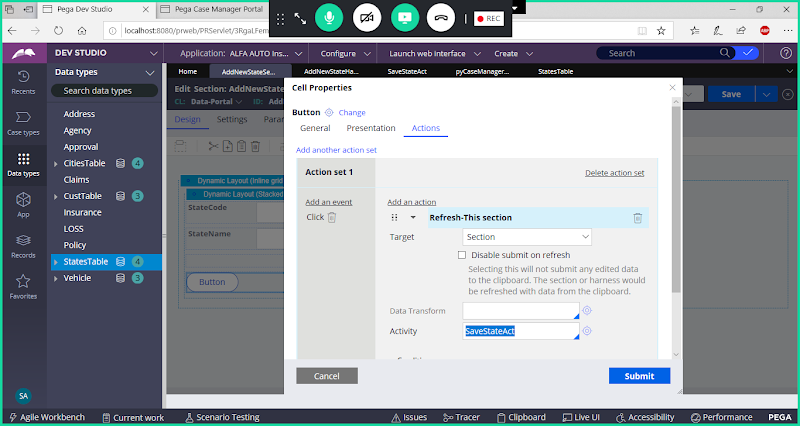
**Open Button settings**



**Set Event & Action:**



**Give Activity Name**



**Submit**

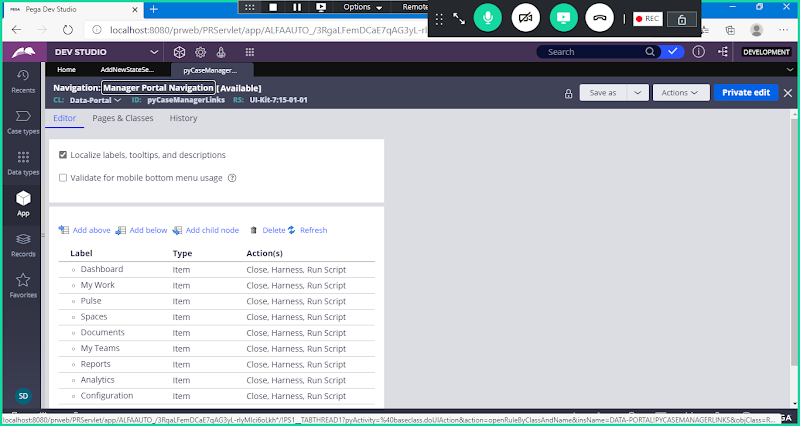
**Save section**

**Add New State Menu Item in Manager Portal:**

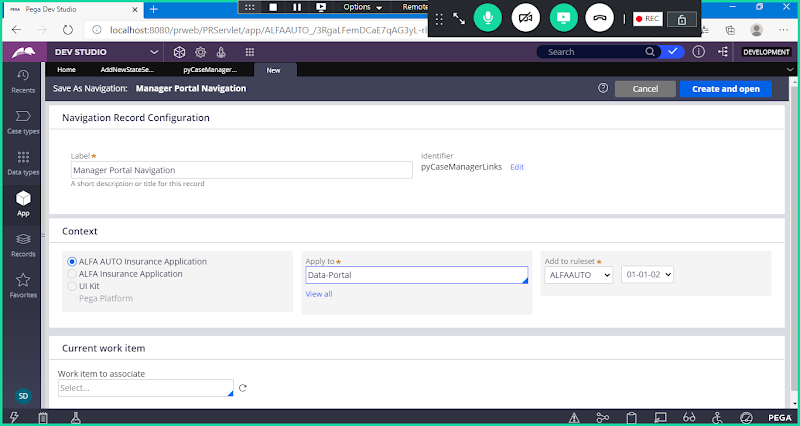
**Predefined Navigation Rule where left Menu available is pyCaseManagerLinks**

**Navigation:  Is a User Interface Rule, used to Add Menu Items & set Menu Actions( Menu Control)**

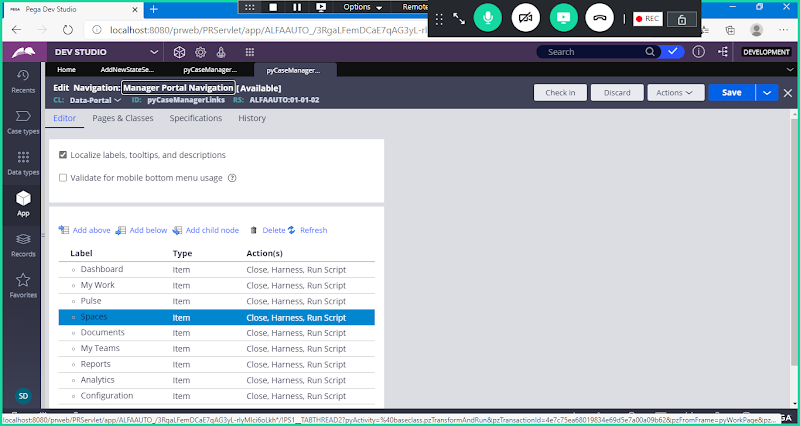
**Now Open & Customize pyCaseManagerLinks to add new Menu item ( Add New State)**



**Use Saveas**

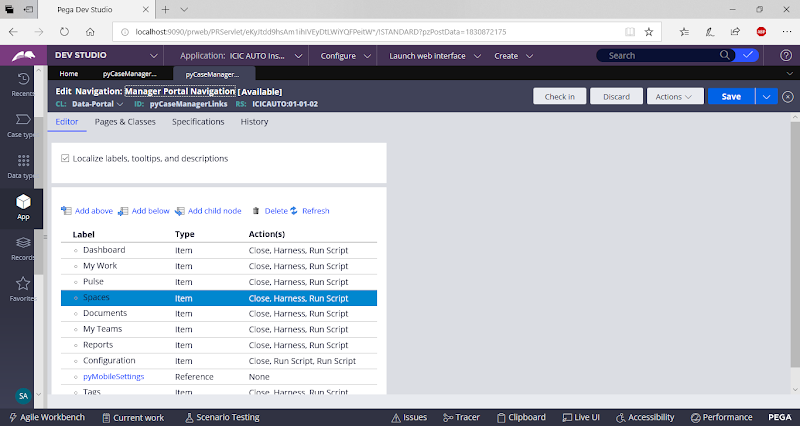


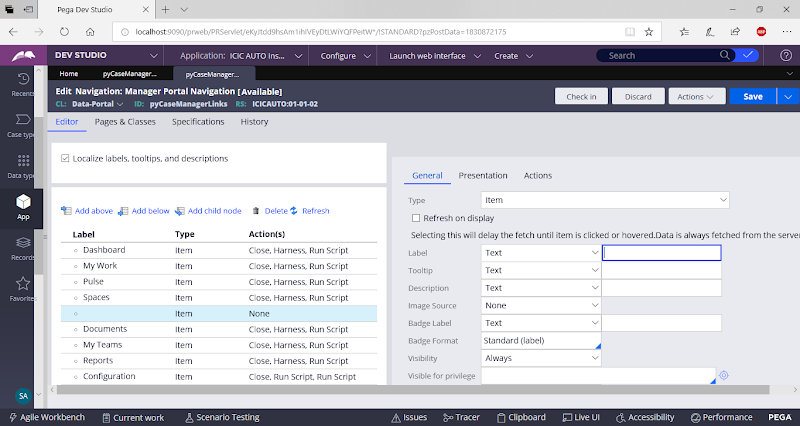
**Open**



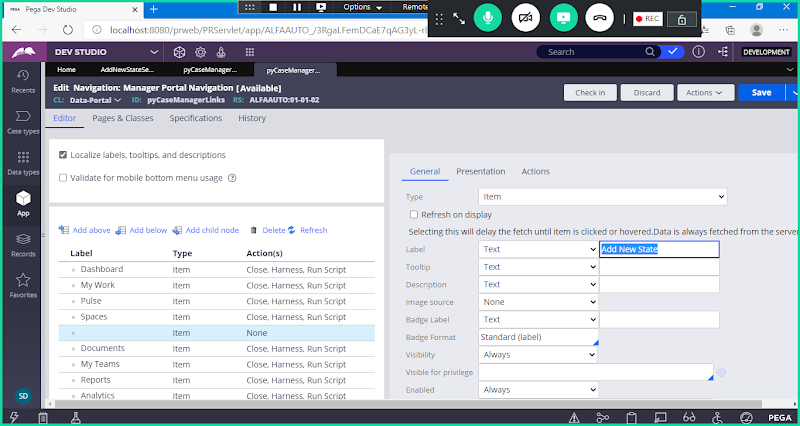
**Add New Menu item After Spaces**

**Select Spaces & Click on Add Below:**

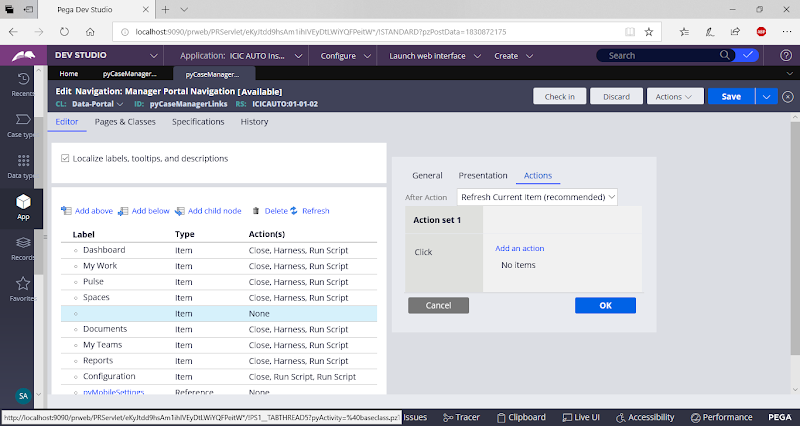




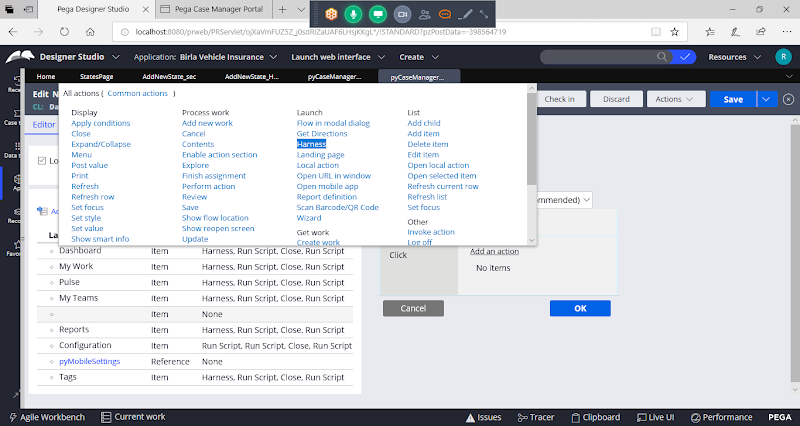
**Add Name**

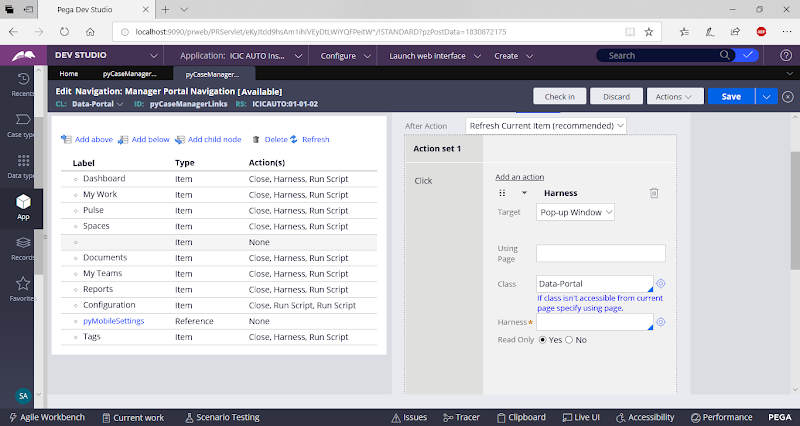


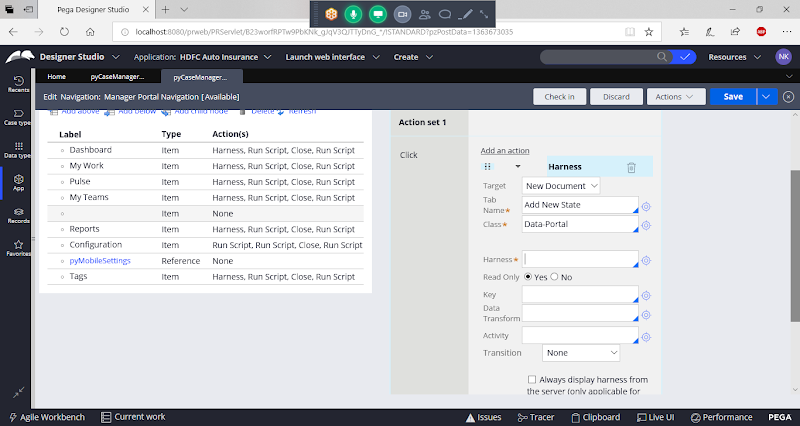
**Now go to Actions Tab**



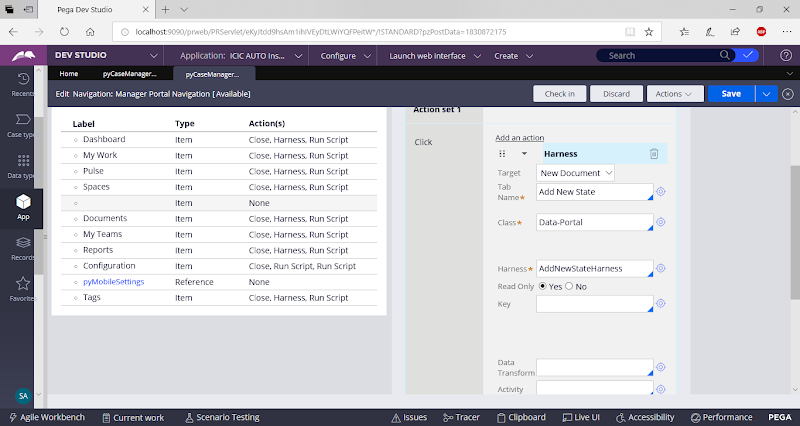
**Add an action: Launch Harness**



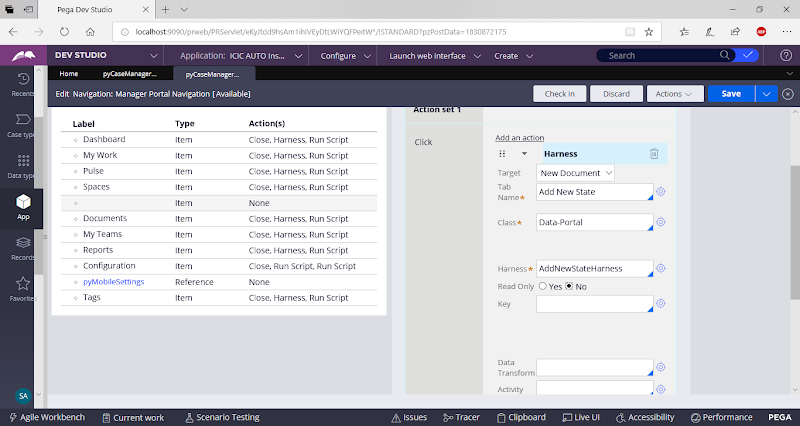




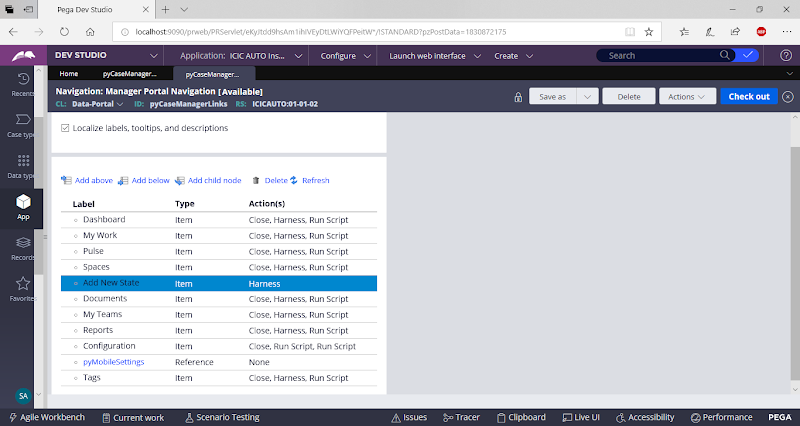
**Add Harness Name:**



**Select Read Only as No**

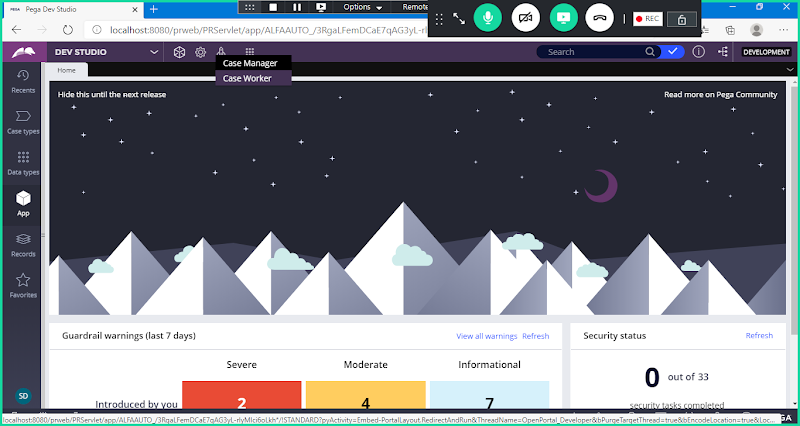


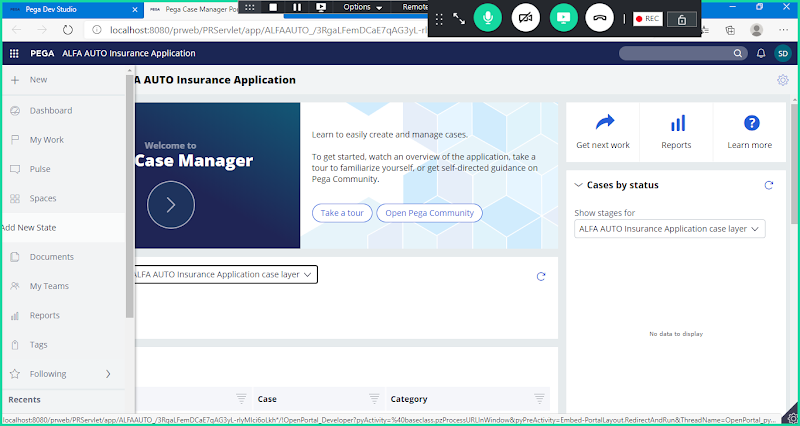
**Clcik on Ok**



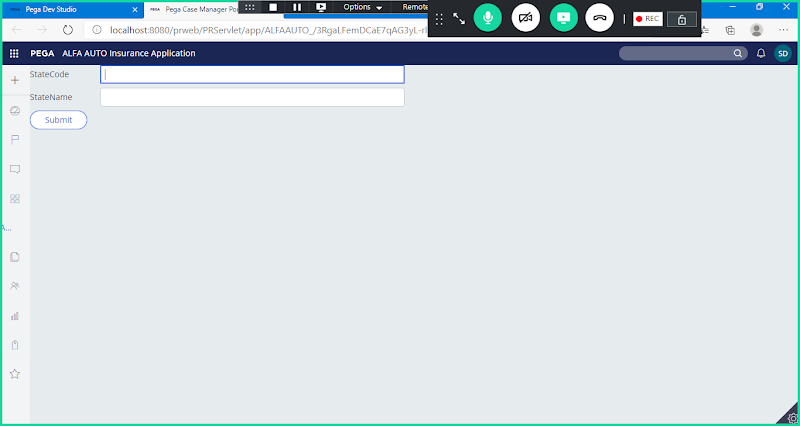
**Save Navigation Rule & Check in**

**Now Open Manager Portal & Add New State Values & Insert**





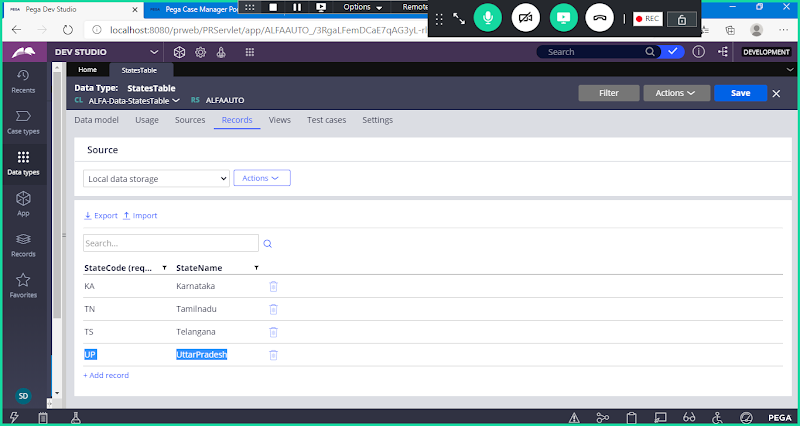
**Click on Add New State**





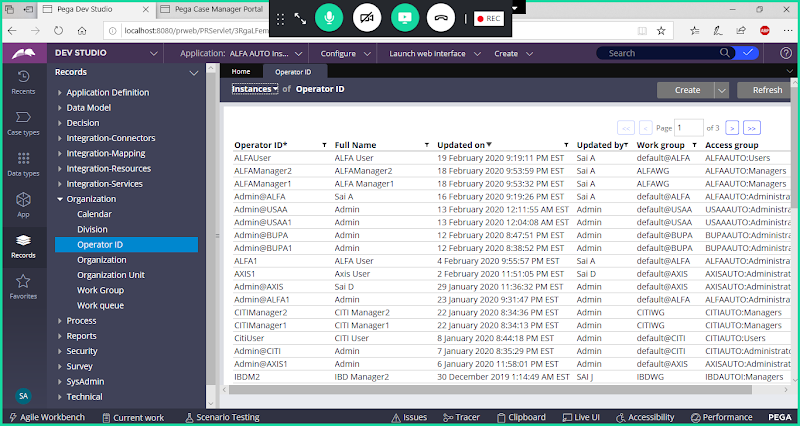
**Submit**

**So, Data Inserted to states table**

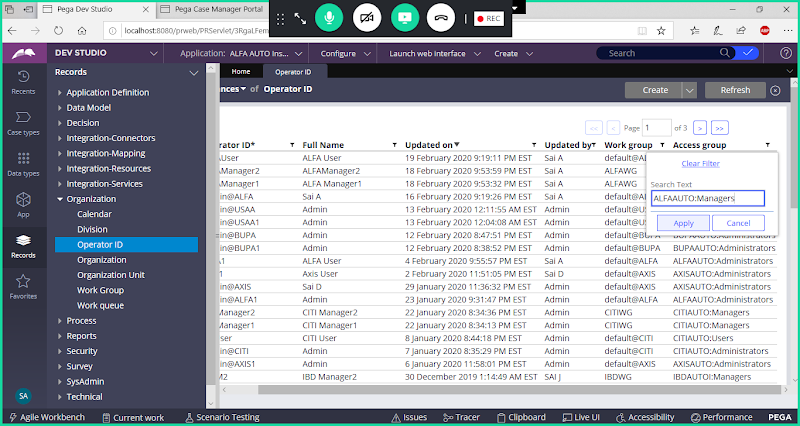


**Before Proceeding make sure all the Managers ID's have valid email id assigned.**

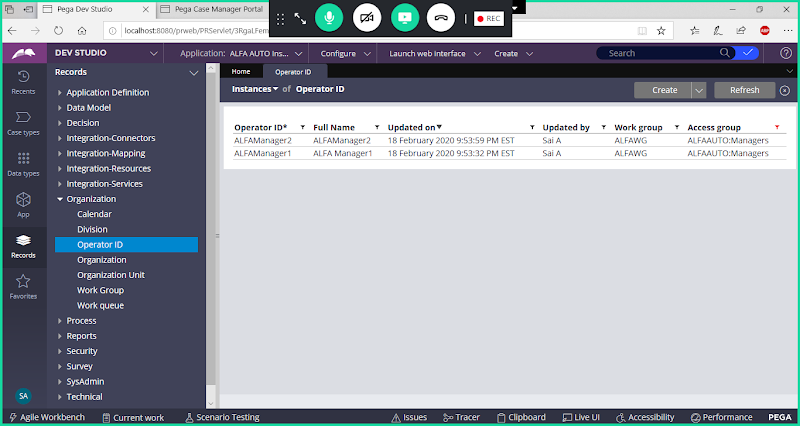
**Go to Records-> Organization -> Operator ID**



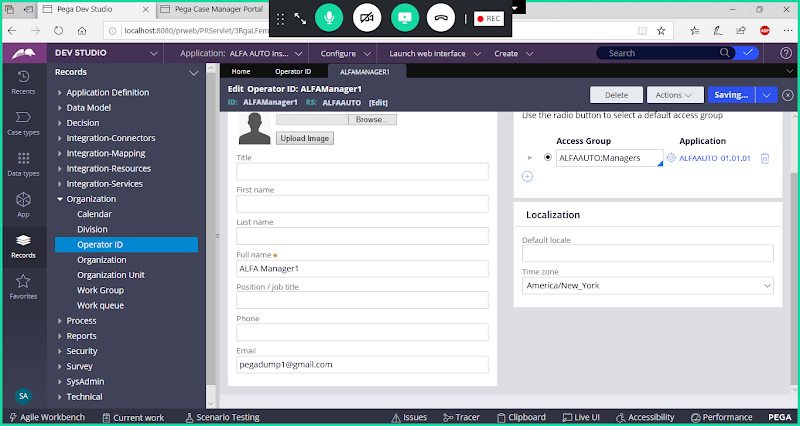
**Apply Filter with Manager Access Group**



**Click on Apply**



**Open Each ID & Assign Valid Email Address**



**Simiarly update email id for others managers ids.**

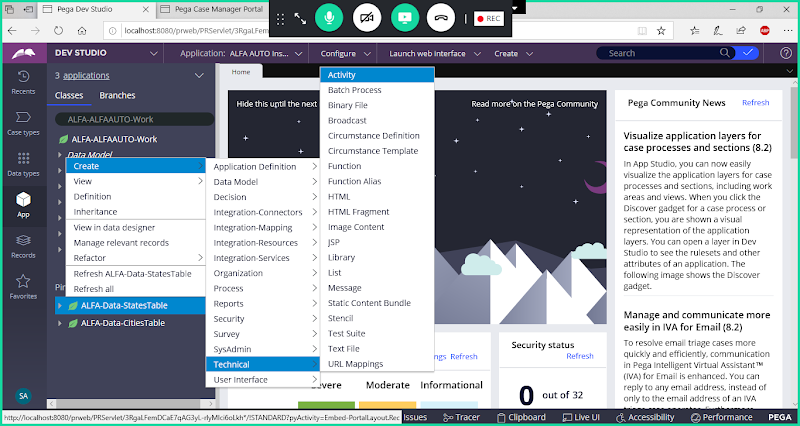
**============**

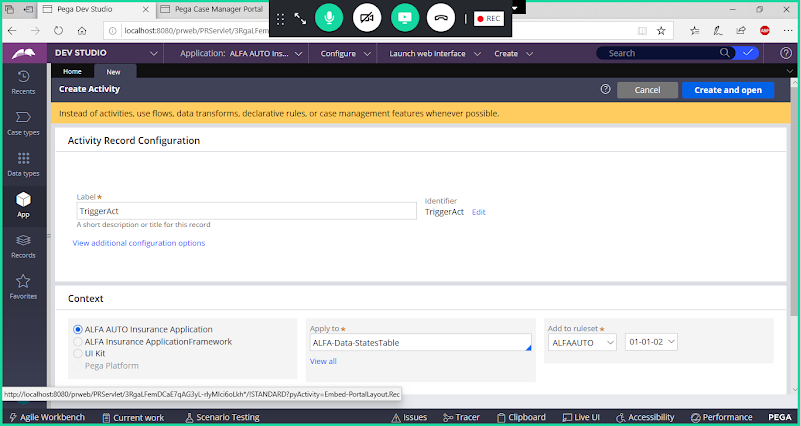
**Open StatesTable Data Class:**

**Create an Activity to Fetch Manager email ids & Send email notifications. Activity Type must be Trigger Activity.**

**Trigger Activity & Declare  Trigger Rule We need to Define under Data Class ( States Table)**

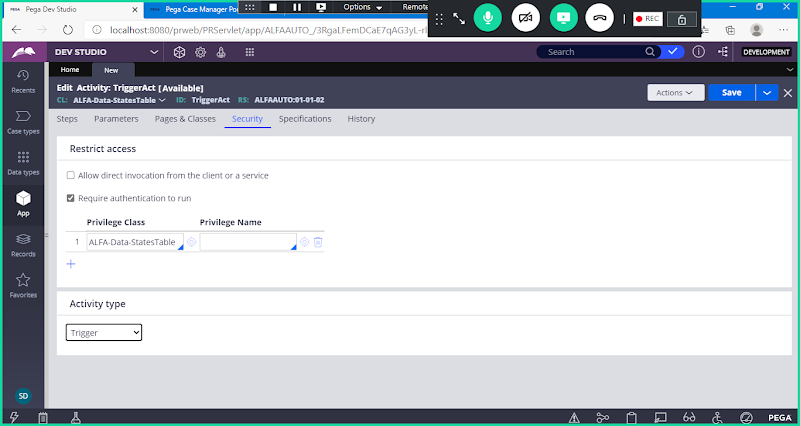
**Create Activity Under States Data Class:**



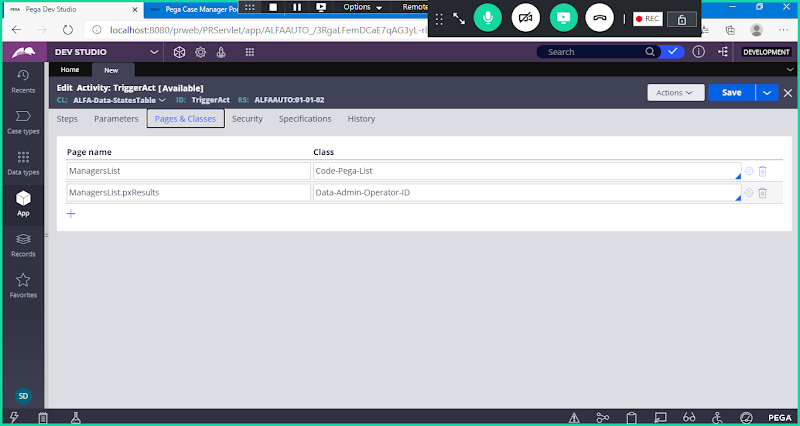


**Open**

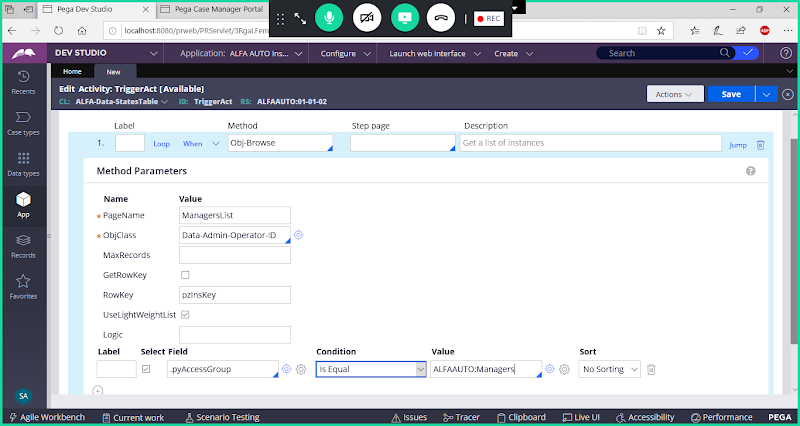
**Go to Security -> Set Activity Type: Trigger**

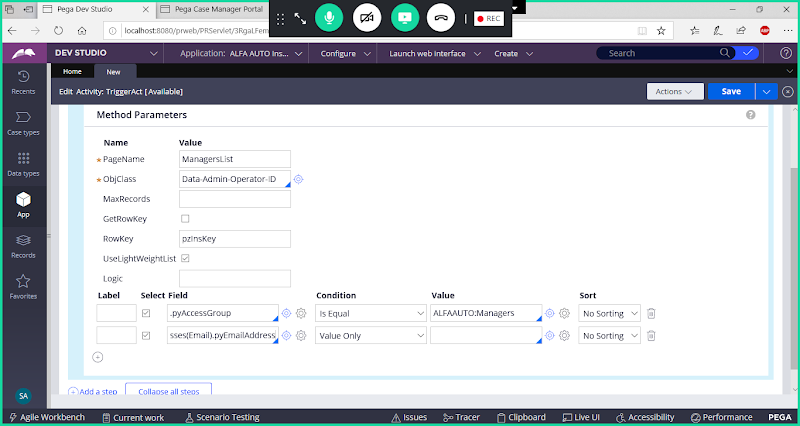


**Go to Pages & Classes**



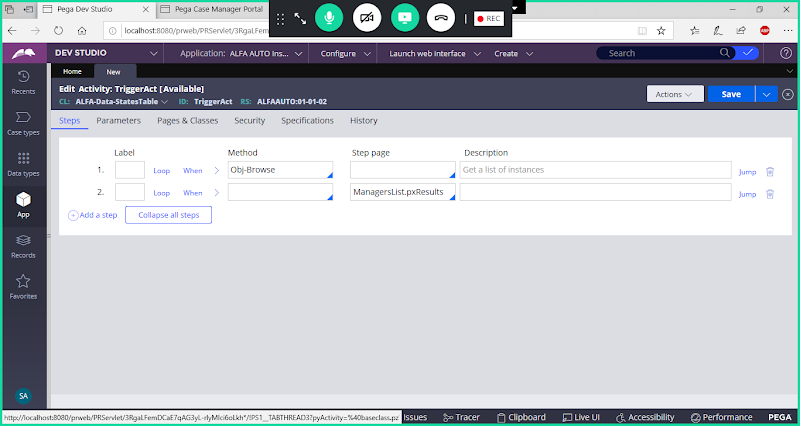
**Go t o Steps:**





**.pyAddresses(Email).pyEmailAddress**

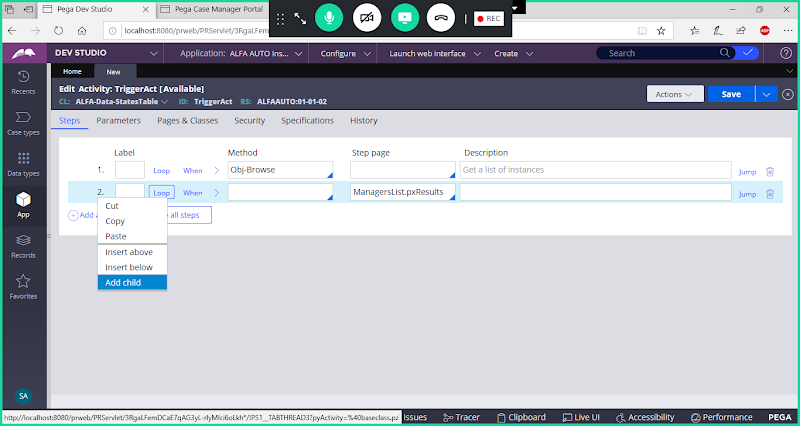
**Add Loop:**

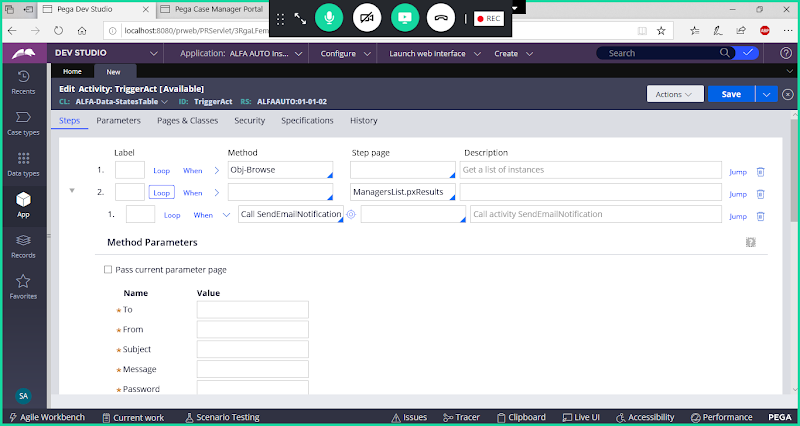


**Now Select Loop : For Each Embedded Page**

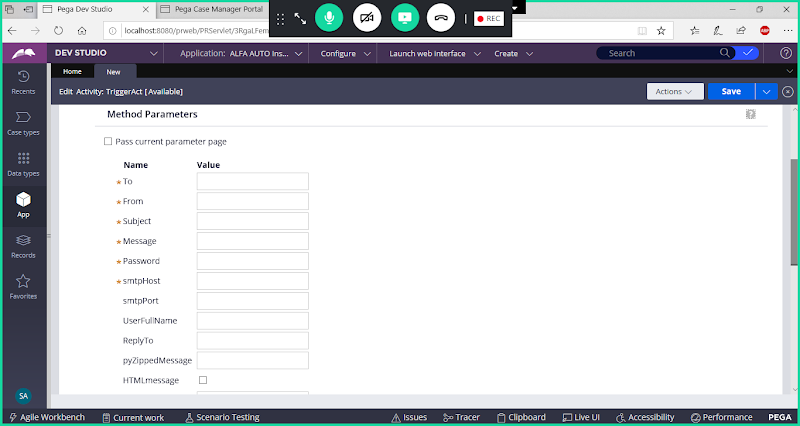


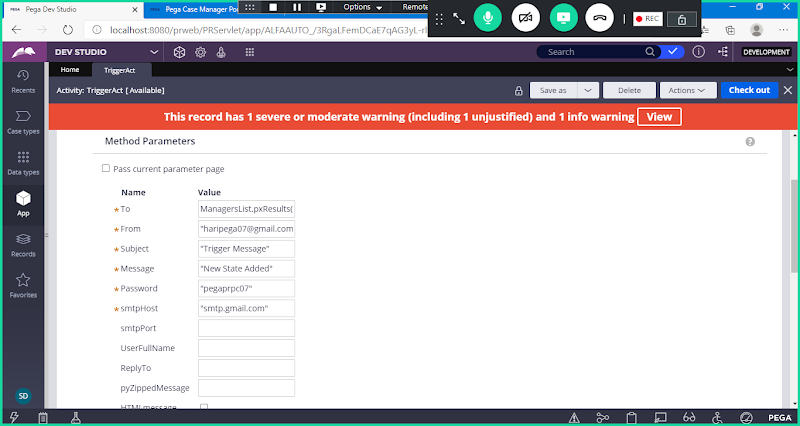
**Add Child under Loop:**



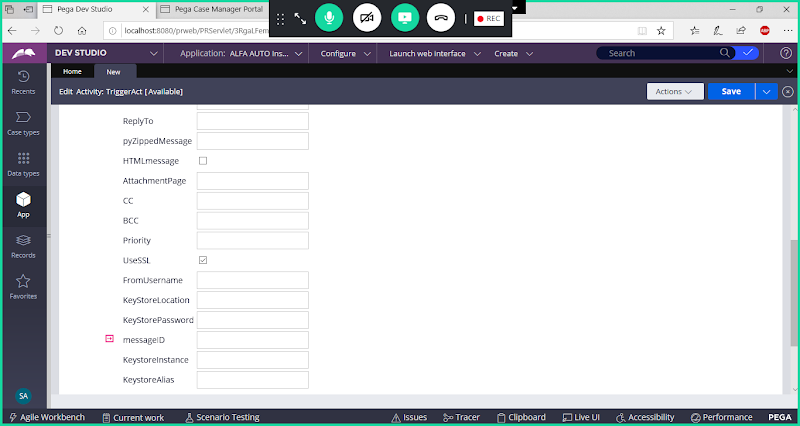


**Now Fill Parameters:**



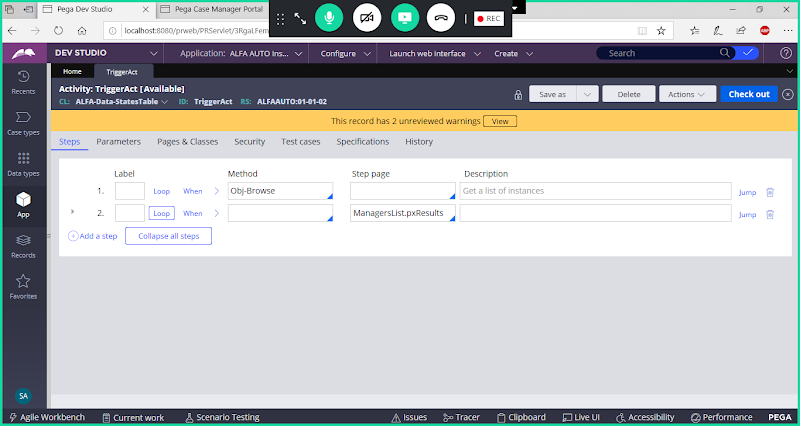


**Enable Use SSL**



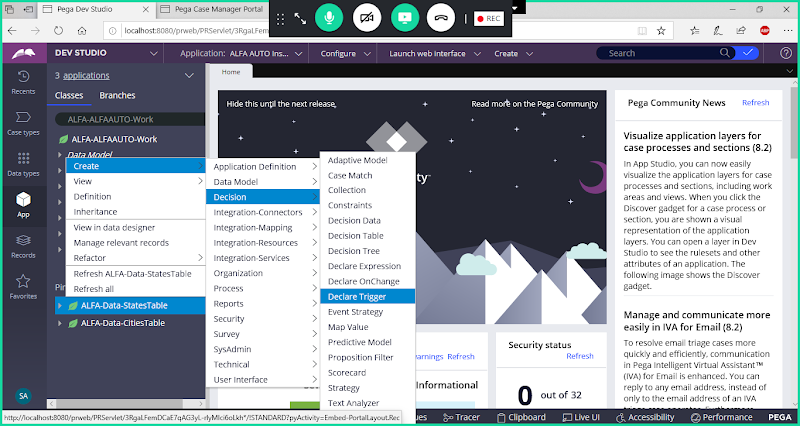
**To:**

**ManagersList.pxResults(<Current>).pyAddresses(Email).pyEmailAddress**



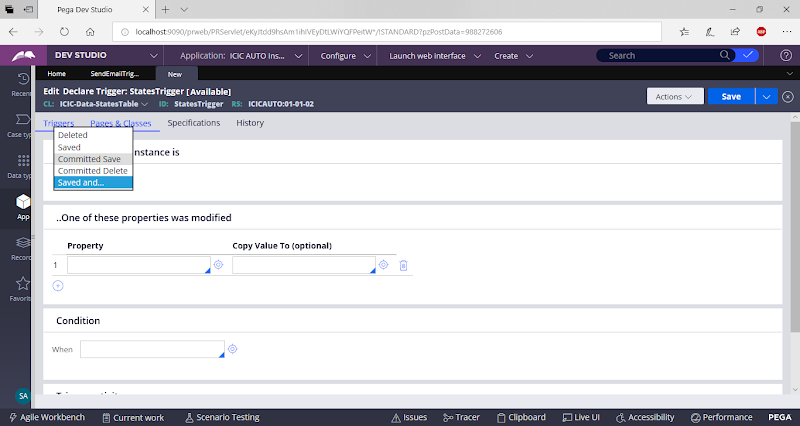
**Save Activity**

**Create Declare Trigger Rule:**





**Open**



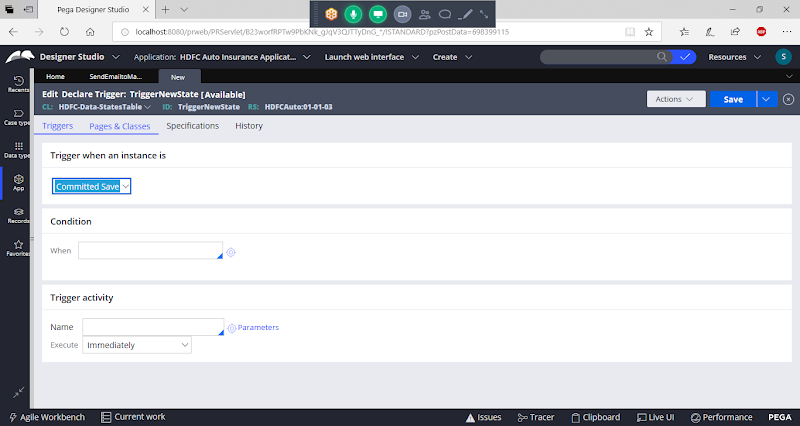
**Saved and..: When Updating data in table**

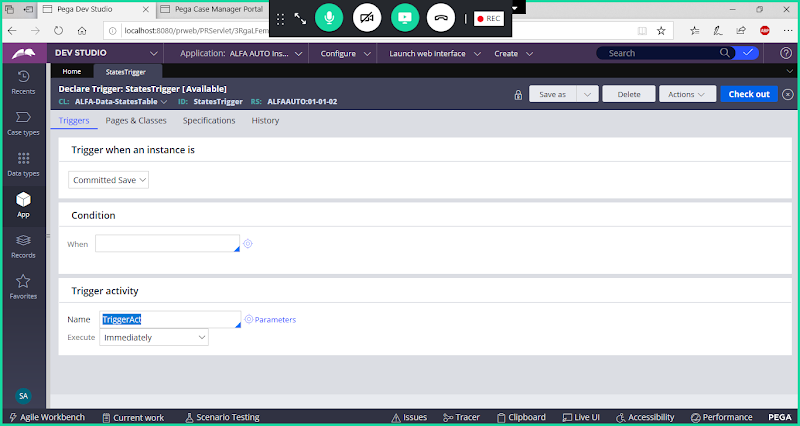
**Committed Save: If Save is Non Transactional Processing use this**

**Committed Delete: If delete is non transaction processing use this**

**Saved: If it is a Transaction processing & Inserting data from table use this**

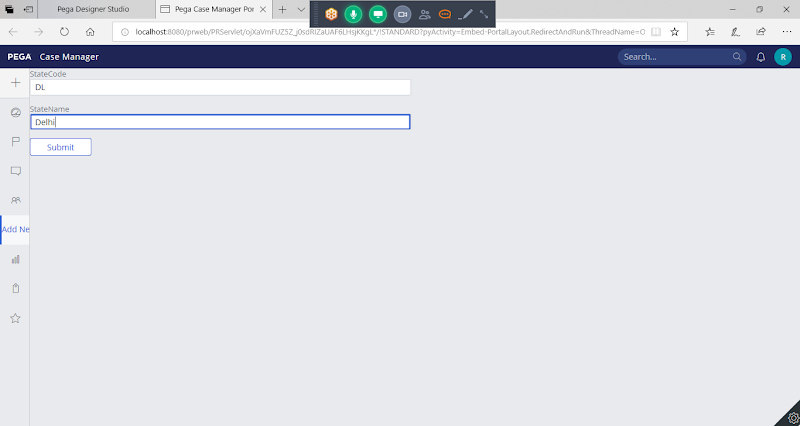
**Deleted: If it is a Transaction processing & deleting data from table use this**





**Save Trigger Rule**

**Open Manager Portal & Add New State & Check whether notification is sent or not.**



**Submit**