

Salesforce Trigger and controllers Assignment

Trigger Assignment-

1. How to create triggers? Explain with Example
 - Create a Apex trigger

```

1 trigger AccountTrigger on Account (before insert, before update) {
2     if((Trigger.isbefore) && (Trigger.isInsert) || (Trigger.isUpdate))
3
4     for (Account acc : Trigger.new) {
5         if( acc.Industry=='Biotechnology' || acc.Industry=='Education'){
6             acc.Rating='Hot';
7         }
8     }
9 }

```

Save it
Activate it

Apex Triggers

This page allows you to view and modify all the triggers in your organization. To create a new trigger, navigate to the appropriate sObject triggers page.

Percent of Apex Used: 0.01%
You are currently using 404 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 8,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Compile all triggers

View: All

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	AccountTrigger		Account	59.0	Active	308	Veena G. 10/17/2023, 2:38 PM	<input type="checkbox"/>

Check for result

Account Trigger

Type Phone Website Account Owner Account Site Industry Biotechnology

Related **Details** News

Account Owner		Rating	Hot
Account Name	Trigger	Phone	
Parent Account		Fax	
Account Number		Website	
Account Site		Ticker Symbol	
Type		Ownership	
Industry	Biotechnology	Employees	
Annual Revenue		SIC Code	
Account Currency	INR - Indian Rupee		
Billing Address		Shipping Address	

Salesforce PD- 1 and 2 Certification Course

2. Write a trigger example to insert contact when you create account

Ans-

The screenshot displays the Salesforce Developer Console interface. At the top, the browser address bar shows the URL: `veenaenterprise-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The console tabs include `Account.obj`, `Account@2:13 PM`, `AccountTrigger.apxt`, `Log executeAnonymous @10/17/2023, 2:15:57 PM`, and `TriggerAccount1.apxt`. The `Code Coverage: None` and `API Version: 59` are displayed.

```
1 trigger TriggerAccount1 on Account (before insert) {
2     List<contact> c= new list<contact>();
3     for (Account acc: trigger.new)
4     {
5         contact c1=new contact();
6         c1.LastName=acc.Name;
7         c.add(c1);
8     }
9     insert c;
10 }
11
12 }
```

Below the code editor, the **Apex Triggers** page is visible. It includes a status bar indicating **Percent of Apex Used: 0.02%** and a table listing triggers.

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	AccountTrigger		Account	59.0	Active	305	Veena G, 10/17/2023, 2:38 PM	<input type="checkbox"/>
Edit Del	TriggerAccount1		Account	59.0	Active	240	Veena G, 10/17/2023, 3:11 PM	<input type="checkbox"/>

Below the table, two sections are shown:

- Accounts Recently Viewed**: 16 items • Updated a minute ago. The list includes a checkbox for **Account Name** and a row with a checkbox and the name **Radha**.
- Contacts Recently Viewed**: 8 items • Updated a few seconds ago. The list includes a checkbox for **Name** and a row with a checkbox and the name **Radha**.

3. Explain Trigger.oldmap with example.

Ans-

Apex Triggers

Percent of Apex Used: 0.02%

You are currently using 1,123 characters of Apex Code (excluding comments and @sTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Compile all triggers

View: All Create New View

Developer Console

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	oldTrigger		Opportunity	59.0	Active	334	Veneta G 10/17/2023, 8:11 PM	<input type="checkbox"/>

le Edit Debug Test Workspace Help

oldTrigger.apxt Log executeAnonymous @10/17/2023, 8:18:08 PM Log executeAnonymous @10/17/2023, 8:17:56 PM

Code Coverage: None API Version: 59

```
1 trigger oldTrigger on Opportunity (before update) {
2
3 for (Opportunity newopp : trigger.new) {
4     String newstage = newopp.StageName;
5     string oldstage = trigger.oldMap.get(newopp.id).StageName;
6     if(oldstage.equals('closed Won') || oldstage.equals('closed Lost')){
7         system.debug('Old Stage Value:'+ oldStage +'new value:'+ newStage);
8     }
9 }
10 }
```

ops Tests Checkpoints Query Editor View State Progress Problems

ser	Application	Operation	Time	Status	Read	Size
tena G	Browser	/aura	10/17/2023, 8:18:08 PM	Success		4.76 KB
tena G	Browser	/aura	10/17/2023, 8:17:56 PM	Success		4.76 KB
tena G	Browser	/aura	10/17/2023, 8:13:01 PM	Success		1.08 KB
tena G	Browser	/aura	10/17/2023, 8:12:11 PM	Success		1.09 KB

20:17:56:004	VARIABLE_SCO...	[4] newstage String false false
20:17:56:004	VARIABLE_ASSI...	[4] newstage "Closed Lost"
20:17:56:004	STATEMENT_EX...	[5]
20:17:56:004	VARIABLE_SCO...	[5] oldstage String false false
20:17:56:004	VARIABLE_ASSI...	[5] oldstage "Perception Analysis"
20:17:56:004	HEAP_ALLOCATE	[6] Bytes:10
20:17:56:004	HEAP_ALLOCATE	[6] Bytes:11
20:17:56:004	STATEMENT_EX...	[6]

Salesforce PD- 1 and 2 Certification Course

4. Write an example to create list in Apex Trigger.

Ans-

Apex Trigger

TriggerList

[Back to List: Email Alerts](#)

Apex Trigger Detail

[Edit](#) [Delete](#) [Download](#) [Show Dependencies](#)

Name	TriggerList	Object Type	Contact
Code Coverage	0% (0/7)	Status	Active
Created By	Veena G	Last Modified By	Veena G
Created	10/18/2023, 12:51 PM	Last Modified	10/18/2023, 1:13 PM
Namespace Prefix			

Apex Trigger [Version Settings](#) [Trace Flags](#)

```
1 trigger TriggerList on Contact (after insert,after update) {
2   List<Contact> contactsToUpdate = new List<Contact>();
3
4   for (Contact updatedContact : Trigger.new) {
5     if (updatedContact.LeadSource == 'Purchased List') {
6       contactsToUpdate.add(updatedContact);
7     }
8   }
9
10  for (Contact contact : contactsToUpdate) {
11
12    contact.Description = 'Contact Lead source status changed: ' + contact.Name;
13  }
14
15  update contactsToUpdate;
16 }
```

Salesforce - Developer Edition

[Edit](#) [Delete](#) [Download](#) [Show Dependencies](#)

Developer Console - Google Chrome

veenaenterprises-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

TriggerList.apxt Log executeAnonymous @10/18/2023, 1:14:21 PM

Code Coverage: None API Version: 59

```
1 trigger TriggerList on Contact (after insert,after update) {
2   List<Contact> contactsToUpdate = new List<Contact>();
3
4   for (Contact updatedContact : Trigger.new) {
5     if (updatedContact.LeadSource == 'Purchased List') {
6       contactsToUpdate.add(updatedContact);
7     }
8   }
9
10  for (Contact contact : contactsToUpdate) {
11
12    contact.Description = 'Contact Lead source status changed: ' + contact.Name;
13  }
14
15  update contactsToUpdate;
16 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
veena G	Browser	/aura	10/18/2023, 1:14:21 PM	Success		1.3 KB

Controller Assignment

1. Design VisualForce Page using Custom Controller Class

Ans-

The screenshot shows the Salesforce Visualforce Pages editor. At the top, there's a 'Visualforce Page' section with a 'VisualController' label. Below this, there's a 'Page Detail' section with various settings like 'Namespace Prefix', 'Require CSRF protection on GET requests', and 'Last Modified By'. The main part of the editor shows the 'Visualforce Markup' with the following code:

```
<apex:page controller="CustomController">
  <apex:pageBlock title="Contact List">
    <apex:pageBlockTable value="{!contactList}" var="contact">
      <apex:column value="{!contact.FirstName}" headerValue="First Name"/>
      <apex:column value="{!contact.LastName}" headerValue="Last Name"/>
      <apex:column value="{!contact.Email}" headerValue="Email"/>
      <apex:column value="{!contact.Phone}" headerValue="Phone"/>
    </apex:pageBlockTable>
  </apex:pageBlock>
</apex:page>
```

Below the markup, there's a 'Contact List' table with the following data:

First Name	Last Name	Email	Phone
Veena	Giri		
Veena	Giri		
Sayali	Giri	veena1104giri@gmail.com	
Veebha	p	veebhav2013@gmail.com	
Veena	Radha		
Sean	Giri		
Jack	Forbes	sean@edge.com	(512) 757-6000
Pat	Rogers	jrogers@burlington.com	(336) 222-7000
Edna	Stummiller	pat@gvramid.net	(014) 427-4427
	Frank	efrank@genepoint.com	(650) 867-3450

2. Design VisualForce Page using Custom Controller Extensions Class.

Ans- Class-

The screenshot shows the Salesforce IDE with the following code for a custom controller extension class:

```
public with sharing class MyControllerExtension {
    private final Account acc;

    public MyControllerExtension(ApexPages.StandardController stdController) {
        this.acc = (Account)stdController.getRecord();
    }

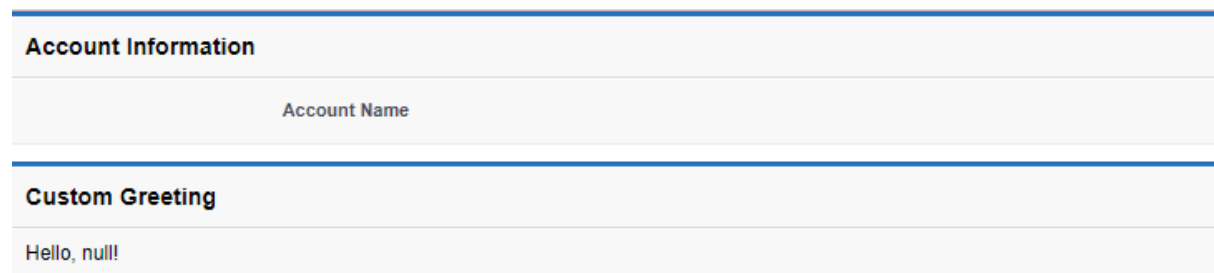
    public String getGreeting() {
        return 'Hello, ' + acc.Name + '!';
    }
}
```

Visual Page-



The screenshot shows the Salesforce Visual Page Editor interface. The browser address bar displays the URL: `veenaentepries-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apex`. The editor has tabs for `MyControllerExtension.apxc` and `ControllerExtension.vfp`. The `ControllerExtension.vfp` tab is active, showing the following Apex code:

```
1 <apex:page standardController="Account" extensions="MyControllerExtension">
2   <apex:form>
3     <apex:pageBlock title="Account Information">
4       <apex:pageBlockSection>
5         <apex:outputField value="{!Account.Name}" />
6       </apex:pageBlockSection>
7     </apex:pageBlock>
8     <apex:pageBlock title="Custom Greeting">
9       <apex:outputText value="{!greeting}" />
10    </apex:pageBlock>
11  </apex:form>
12 </apex:page>
```

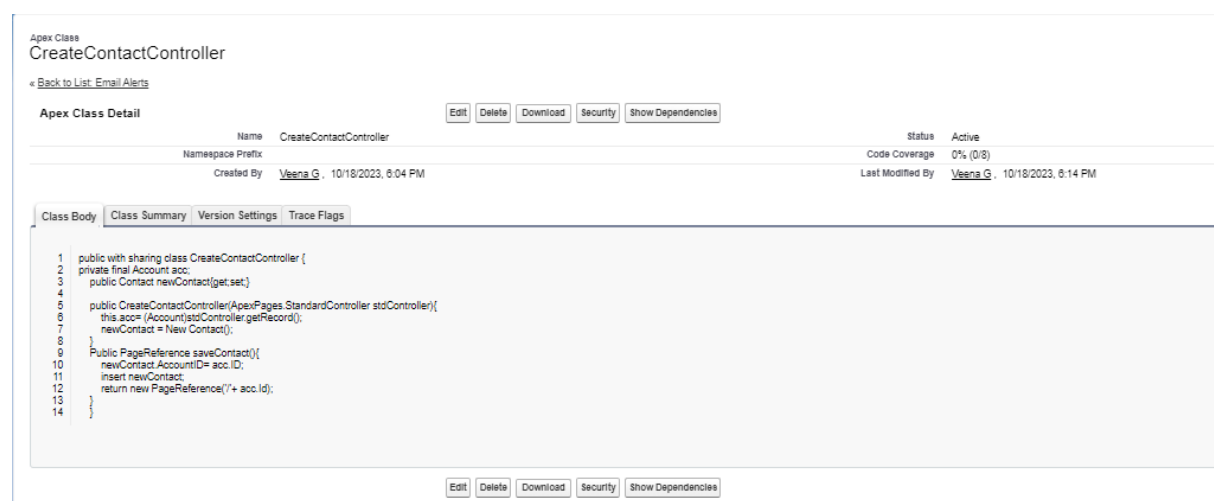


The screenshot shows the preview of the Visual Page. It consists of two main sections:

- Account Information**: A section with a sub-section titled "Account Name".
- Custom Greeting**: A section displaying the text "Hello, null!".

3. Write an example to create new records using custom controller.

Ans- Created Apex class-



The screenshot shows the Salesforce Apex Class Editor for the class `CreateContactController`. The class is in the `veena` namespace. The class body is as follows:

```
1 public with sharing class CreateContactController {
2   private final Account acc;
3   public Contact newContact(get: set)
4
5   public CreateContactController(ApexPages.StandardController stdController){
6     this.acc= (Account)stdController.getRecord();
7     newContact = New Contact();
8   }
9   Public PageReference saveContact(){
10     newContact.AccountID= acc.ID;
11     insert newContact;
12     return new PageReference("/"+ acc.Id);
13   }
14 }
```

Salesforce PD- 1 and 2 Certification Course

Visual page-

The screenshot shows the Salesforce Visualforce Pages editor interface. At the top, there's a 'SETUP' button and the 'Visualforce Pages' header. Below this, the page is titled 'Visualforce Page Contactcontrollervisual'. A 'Page Detail' section shows the page name 'Contactcontrollervisual' and various settings like 'Namespace Prefix', 'Available for Lightning Experience, Experience Builder sites, and the mobile app', 'Require CSRF protection on GET requests', 'Last Modified By', and 'Created By'. Below the details, there are tabs for 'Visualforce Markup' and 'Version Settings'. The 'Visualforce Markup' tab is active, displaying the following code:

```
<apex:page standardController="Account" extensions="CreateContactController">
<apex:form>
  <apex:pageBlock title="New Contact">
    <apex:pageBlockSection>
      <apex:inputField value="{!newContact.FirstName}" />
      <apex:inputField value="{!newContact.LastName}" />
    </apex:pageBlockSection>
    <apex:pageBlockButtons>
      <apex:commandButton action="{!saveContact}" value="Save" />
    </apex:pageBlockButtons>
  </apex:pageBlock>
</apex:form>
</apex:page>
```

Below the code, there are buttons for 'Edit', 'Delete', 'Clone', 'Where is this used?', 'Show Dependencies', and 'Preview'. Below the editor, a browser window shows the preview of the 'New Contact' page. It has a 'Save' button at the top right. The form has two input fields: 'First Name' and 'Last Name'. There is a 'Save' button at the bottom center.

4. Write an example to manage list of records by using standard controller

Ans- create apex class-

The screenshot shows the Salesforce Apex Class editor interface. At the top, there's a 'manageextension' title. Below this, there's a 'Back to List Email Alerts' link. The 'Apex Class Detail' section shows the class name 'manageextension' and various settings like 'Name', 'Namespace Prefix', 'Created By', 'Last Modified By', 'Status', 'Code Coverage', and 'Last Modified By'. Below the details, there are tabs for 'Class Body', 'Class Summary', 'Version Settings', and 'Trace Flags'. The 'Class Body' tab is active, displaying the following code:

```
1 public class manageextension {
2   public List<Opportunity> Opportunities { get; set; }
3
4   public manageextension(ApexPages.StandardController stdController) {
5     Opportunities = [SELECT Id, Name, StageName, Amount FROM Opportunity];
6   }
7 }
```

Below the code, there are buttons for 'Edit', 'Delete', 'Download', 'Security', and 'Show Dependencies'.

Salesforce PD- 1 and 2 Certification Course

Visualforce Page

AccountListPage

Page Detail

Edit Delete Clone Where is this used? Show Dependencies Preview

Label	AccountListPage	Name	AccountListPage
Namespace Prefix		Available for Lightning Experience, Experience Builder sites, and the mobile app	<input type="checkbox"/>
Require CSRF protection on GET requests	<input type="checkbox"/>	Description	
Last Modified By	Veena G. 10/18/2023, 9:38 PM	Created By	Veena G. 10/18/2023, 8:10 PM

Visualforce Markup Version Settings

```
<apex:page standardController="Opportunity" extensions="manageextension">
  <apex:form>
    <apex:pageBlock title="Opportunity List">
      <apex:pageBlockButtons location="top">
        <apex:commandButton action="{!URLFOR($Action.Opportunity.New)}" value="New Opportunity"/>
      </apex:pageBlockButtons>
      <apex:pageBlockTable value="{!Opportunities}" var="opp">
        <apex:column value="{!opp.Name}" headerValue="Opportunity Name"/>
        <apex:column value="{!opp.StageName}" headerValue="Stage"/>
        <apex:column value="{!opp.Amount}" headerValue="Amount"/>
        <apex:column headerValue="Actions">
          <apex:commandButton action="{!URLFOR($Action.Opportunity.Edit, opp.Id)}" value="Edit"/>
          <br/>
          <apex:commandLink action="{!URLFOR($Action.Opportunity.Delete, opp.Id)}" value="Delete"/>
        </apex:column>
      </apex:pageBlockTable>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

Opportunity List [New Opportunity](#)

Opportunity Name	Stage	Amount	Actions
New	Qualification		Edit Delete
VIP CustomerModule 5- op 1	Closed Won	INR 100,000.00	Edit Delete
Record	Closed Won		Edit Delete
VIP CustomerSayali	Qualification	INR 10,000.00	Edit Delete
Edge Emergency Generator	Closed Won	INR 75,000.00	Edit Delete
Edge Installation	Closed Won	INR 50,000.00	Edit Delete
Edge SLA	Closed Won	INR 60,000.00	Edit Delete
Edge Emergency Generator	Id. Decision Makers	INR 35,000.00	Edit Delete
VIP CustomerAction for update	Prospecting	INR 1,001.00	Edit Delete
Burlington Textiles Weaving Plant Generator	Closed Won	INR 235,000.00	Edit Delete
Pyramid Emergency Generators	Prospecting	INR 100,000.00	Edit Delete
Dickenson Mobile Generators	Qualification	INR 15,000.00	Edit Delete
Grand Hotels Kitchen Generator	Id. Decision Makers	INR 15,000.00	Edit Delete
Grand Hotels Guest Portable Generators	Value Proposition	INR 250,000.00	Edit Delete
Grand Hotels Generator Installations	Closed Won	INR 350,000.00	Edit Delete
Grand Hotels SLA	Closed Won	INR 90,000.00	Edit Delete
Grand Hotels Emergency Generators	Closed Won	INR 210,000.00	Edit Delete
United Oil SLA	Closed Won	INR 120,000.00	Edit Delete
United Oil Office Portable Generators	Negotiation/Review	INR 125,000.00	Edit Delete

5. Explain Rendered Function with example.

Ans-

Visualforce Page

VisualRendered

Page Detail

Edit Delete Clone Where is this used? Show Dependencies Preview

Label	VisualRendered	Name	VisualRendered
Namespace Prefix		Available for Lightning Experience, Experience Builder sites, and the mobile app	<input type="checkbox"/>
Require CSRF protection on GET requests	<input type="checkbox"/>	Description	
Last Modified By	Veena G. 10/19/2023, 12:04 AM	Created By	Veena G. 10/18/2023, 11:52 PM

Visualforce Markup Version Settings

```
<apex:page standardController="Account">
  <apex:form>
    <apex:pageBlock title="Account Information">
      <apex:pageBlockSection title="Account Details" columns="2">
        <apex:outputField value="{!Account.Name}" rendered="{!Account.Type == 'Customer - Channel'}"/>
        <apex:outputField value="{!Account.Type}" rendered="{!Account.Type != 'Customer - Channel'}"/>
      </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

Edit Delete Clone Where is this used? Show Dependencies Preview

Salesforce PD- 1 and 2 Certification Course

The screenshot displays the Salesforce Visualforce page editor interface. On the left, a preview of the 'Account Information' page is shown, featuring a tab labeled 'Account Details'. The right pane displays the Visualforce code for the page, which is a Visualforce Page (VFP) titled 'Account Information'. The code uses the `standardController="Account"` and includes a `pageBlockSection` titled 'Account Details' with two columns. The first column displays the account name using `{!Account.Name}`, and the second column displays the account type using `{!Account.Type}`. The code is as follows:

```
1 <apex:page standardController="Account">
2   <apex:form>
3     <apex:pageBlock title="Account Information">
4       <apex:pageBlockSection title="Account Details" columns="2">
5         <apex:outputField value="{!Account.Name}" rendered="{!Account.Name != null}">
6           <apex:outputField value="{!Account.Type}" rendered="{!Account.Type != null}">
7         </apex:pageBlockSection>
8       </apex:pageBlock>
9     </apex:form>
10  </apex:page>
```