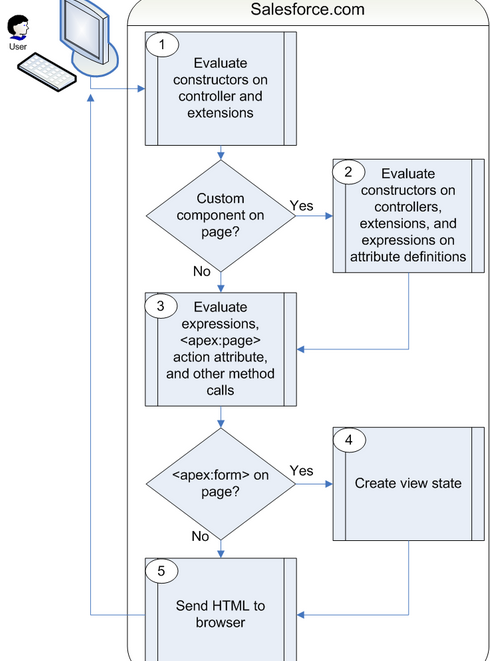
**Visualforce page life cycle in salesforce**

When a user views a Visualforce page, instances of the controller, extensions, and components associated with the page are created by the server. The order in which these elements are executed can affect how the page is displayed to the user   
  
There are two types of Visualforce page requests:

A get request is an initial request for a page either made when a user enters an URL or when a link or button is clicked that takes the user to a new page.

A postback request is made when user interaction requires a page update, such as when a user clicks on a Save button and triggers a save action.

**Order of Execution for Visualforce Page Get Requests**   
  
A get request is an initial request for a page either made when a user enters an URL or when a link or button is clicked that takes the user to a new page. The following diagram shows how a Visualforce page interacts with a controller extension or a custom controller class during a get request:

[](http://4.bp.blogspot.com/-RgCQtI0Ky0I/VPL30roMvEI/AAAAAAAAAMk/IrkjhubevLg/s1600/VF1.png)

In the diagram above the user initially requests a page, either by entering a URL or clicking a link or button. This initial page request is called the get request.

**Constructor methods** on the associated **controller / extension** classes are called, instantiating the controller objects.

If the page contains any **custom components** , they are created and the constructor methods on any associated custom controllers or controller extensions are executed. If attributes are set on the custom component using expressions, the expressions are evaluated after the constructors are evaluated.

The page then executes any **assignTo attributes on any custom components** on the page. After the assignTo methods are executed, expressions are evaluated, the **action attribute** on the **<apex:page>** component is evaluated, and all other method calls, such as **getting or setting** a property value, are made.

If the page contains an **<apex:form>** component, all of the information necessary to maintain the state of the database between page requests is saved as an encrypted view state. The view state is updated whenever the page is updated.

The resulting HTML is sent to the browser. If there are any client-side technologies on the page, such as JavaScript, the browser executes them.

As the user interacts with the page, the page contacts the controller objects as required to execute action, getter, and setter methods.

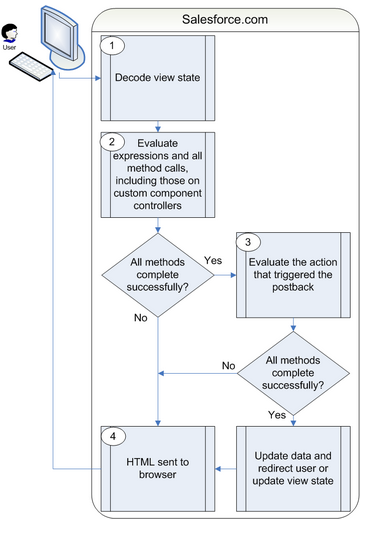
Once a new get request is made by the user, the view state and controller objects are deleted

**Note** :-  If the user is redirected to a page that uses the same controller and the same or a proper subset of controller extensions, a postback request is made. When a postback request is made, the view state is maintained

Constructor of controller /extension - >  constructor of custom components  -> AssignTo any Custom Components -> Action method->Getter or Setter -> Form method -> HTML

**Order of Execution for Visualforce Page Postback Requests**

A postback request is made when user interaction requires a page update, such as when a user clicks on a Save button and triggers a save action. The following diagram shows how a Visualforce page interacts with a controller extension or a custom controller class during a postback request:

[](http://2.bp.blogspot.com/-S2T6YGX3rRw/VPL3qqJzh8I/AAAAAAAAAMc/Nniunf97wRA/s1600/VF2.png)

During a postback request, the view state is decoded and used as the basis for updating the values on the page.

After the view state is decoded, expressions are evaluated and set methods on the controller and any controller extensions, including set methods in controllers defined for custom components, are executed.

The action that triggered the postback request is executed. If that action completes successfully, the data is updated. If the postback request returns the user to the same page, the view state is updated

The resulting HTML is sent to the browser

If the postback request indicates a page redirect and the redirect is to a page that uses the same controller and a proper subset of controller extensions of the originating page, a postback request is executed for that page. Otherwise, a get request is executed for the page. If the postback request contains an **<apex:form>** component, only the ID query parameter on a postback request is returned

**NOTE :- Once the user is redirected to another page, the view state and controller objects are deleted.**