Apex Form Component

Apex Form Component

Apex Form Component is used to create Forms in Visualforce page. Apex input and output components are added inside <apex:form> Component. It is required to add only one <apex:form> tag instead of adding many form tags. The body of the form determines the data that is displayed and the way it is processed.

Apex Form Component Attributes.

Accept	Accept is an attribute and it's a string type, the text to display as a tooltip when the user's mouse pointer hovers over this component.
Accept charset	Its a string type, it's a comma separated list of characters encoding that a server processing this form can handle.
forceSSL	The form will be submitted using SSL.
onclick	A javascript invoked if the onclick event occurs.
ondclick	A javascript invoked if the ondblclick event occurs.
target	This attribute includes target names like "_blank", "_parent","_self" and "_top".
prependId	It is a boolean type that specifies whether this form should prepend its ID to the IDS of it child component.

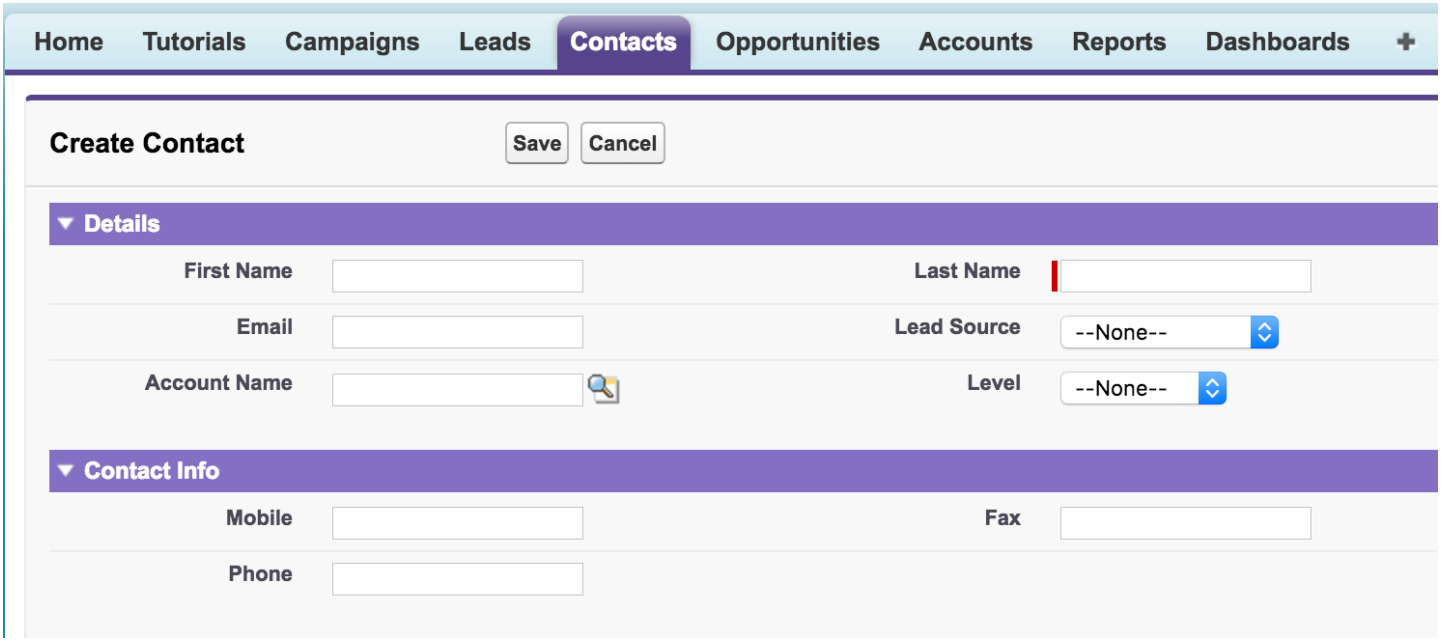
Visualforce page

```
<apex:page
```

```
<apex:page standardController="contact" sidebar="false">
    <apex:form >
        <apex:pageBlock mode="detail" title="{!if($CurrentPage.parameters.id == null,
                                'Create Contact', 'Edit Contact')}">
            <apex:pageBlockButtons >
                <apex:commandButton value="Save" action="{!Save}"/>
                <apex:commandButton value="Cancel" action="{!Cancel}"/>
            </apex:pageBlockButtons>
                <apex:pageBlockSection Title="Details">
                <apex:inputField value="{!contact.firstname}"/>
                <apex:inputField value="{!contact.lastname}"/>
                <apex:inputField value="{!contact.Email}"/>
                <apex:inputField value="{!contact.Leadsource}"/>
                <apex:inputField value="{!contact.Accountid}"/>
                <apex:inputField value="{!contact.Level__c}"/>
                </apex:pageBlockSection>
            <apex:pageBlockSection Title="Contact Info">
                <apex:inputField value="{!contact.mobilephone}"/>
                <apex:inputField value="{!contact.Fax}"/>
                <apex:inputField value="{!contact.Phone}"/>
            </apex:pageBlockSection>

        </apex:pageBlock>
    </apex:form>
</apex:page>
```

Output



Apex Form component supports HTML pass-through attributes[https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_html_features_pass_through_attributes.htm] using the “html-” prefix. Pass-through

attributes are attached to the generated <form> tag. You can add arbitrary attributes to many Visualforce components that are “passed through” to the rendered HTML. This is useful, for example, when using Visualforce with JavaScript frameworks, such as jQuery Mobile, AngularJS, and Knockout, which usedata-* or other attributes as hooks to activate framework functions

Visualforce Tutorials.

- What is Visualforce?
- Difference between Salesforce.com, Force.com and Salesforce1.
- Why should we learn Visualforce?.
- Creating first visualforce page

Visualforce basics

- Building Visualforce page using MVC Model.
- Visualforce Standard Controllers
- Visualforce Custom Controllers
- Visualforce Controller extensions.
- Visualforce standard list controllers.

Visualforce components.

- apex:page.
- apex:pagemessage.
- apex:Form
- apex:pageblock.
- apex:Commandbutton.
- apex:pageblockbuttons.
- apex:pageblockSection.
- apex:pageblockSectionItem.
- apex:detail.
- apex:tab.
- apex:tabpanel.
- apex:panelbar
- apex:panelbarItem.
- apex:facetname

Visualforce - Input Components.

- apex:inputText.
- apex:inputSecret.

- apex:inputHidden.
- apex:inputCheckbox
- apex:inputTextArea.
- apex:selectList
- apex:selectoption
- apex:inputField
- apex:selectRadio
- apex:selectCheckBoxes

Enchancing your Organization.

Using Developer Console.

Unit Test