Working and usage of force:refreshView in lightning

Posted on November 23, 2018 | By **Nigam**

Understanding:

Force:refreshView is an event that can be used to refresh the view. To elaborate, consider a scenario where you have a custom component on the Account detail page and this component consists of a form that updates a field on the account. Now, after you save the record the field will be updated and will be reflected on the component itself but not on the account detail page.To achieve that we have **force:refreshView**. There may be 2 cases in this scenario:-

Case 1: Updating standard view from custom component

Let's say, we have a component which creates a related contact of an account. Now after filling in all details and clicking on save the contact will be created but the related list component of account will not update. Here we can achieve this by firing the force:refreshView event after the save. Consider the following component:

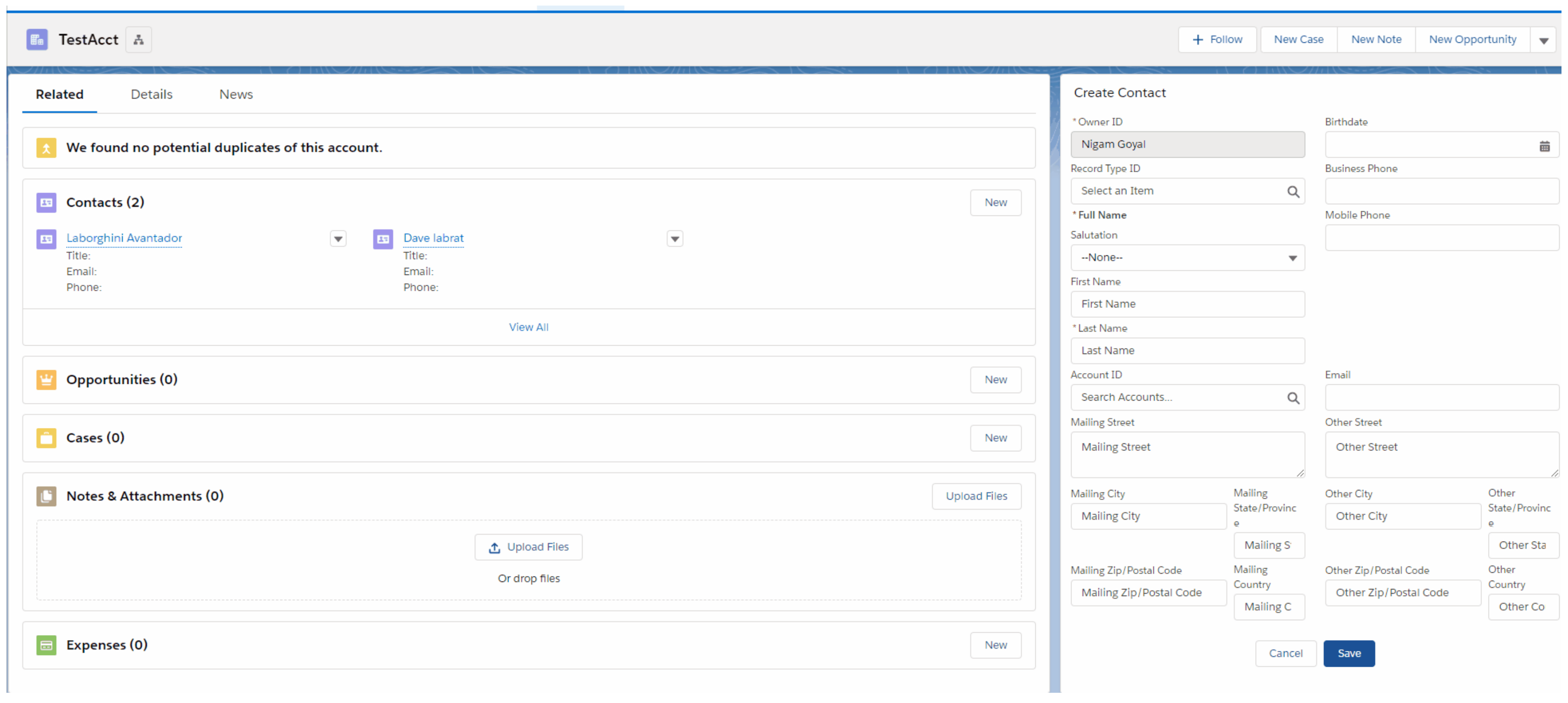
```
<aura:component implements="force:appHostable,flexipage:availableForRecordHome">
        <!--forceRefreshViewFireComp-->
    <aura:attribute name="fieldList" type="List" default="LastName, FirstName" />
    <lightning:card title="Create Contact">
        <div class="slds-m-horizontal_small">
            <lightning:recordForm
                                  objectApiName="Contact"
                                  layoutType="Compact"
                                  columns="2"
                                  mode="edit"
                                  onsubmit="{!c.createContact}" />
        </div>
    </lightning:card>

</aura:component>
```

forceRefreshViewFireComp hosted with ❤ by **GitHub** **view raw**

```
({
        createContact : function(component, event, helper) {
                console.log('inside create contact');
        $A.get('e.force:refreshView').fire();
        console.log('event fired');
        }
})
```

forceRefreshViewFireCont hosted with ❤ by **GitHub** **view raw**

Output:


We found no potential duplicates of this account.
Laborghini Avantador
Dave labrat
Owner ID: Nigam Goyal

Case 2: Updating Custom component view from Standard component

This case is the opposite of the above case. For example, let's say we have a component that displays some fields of the account like Name, Type etc. Now, when we update those fields on the standard component those changes will not be reflected on our custom component unless we handle the force:refreshView event by creating handler to refresh the component. In this scenario we initialize the component in the handler of force:refreshView.

Consider the following code:

```
<aura:component controller="FetchData" implements="force:appHostable,flexipage:availableForRecordHome,force:hasRecordId">
    <!--forceRefreshViewComp-->
    <aura:handler name="init" value="{! this}" action="{! c.doInit}" />
    <aura:handler event="force:refreshView" action="{! c.doInit}" />

    <aura:attribute name="account" type="Account" />


    <lightning:card title="Account Detail">
        <div class="slds-m-horizontal_small">
            <Strong>Account Name:</Strong> {! v.account.Name}   <br/>
            <Strong>Clean Status:</Strong> {! v.account.Type}
        </div>
    </lightning:card>

</aura:component>
```

forceRefreshViewComp hosted with ❤ by **GitHub** **view raw**

```
({
        doInit : function(component, event, helper) {
                helper.getData(component, event);
        },

})
```

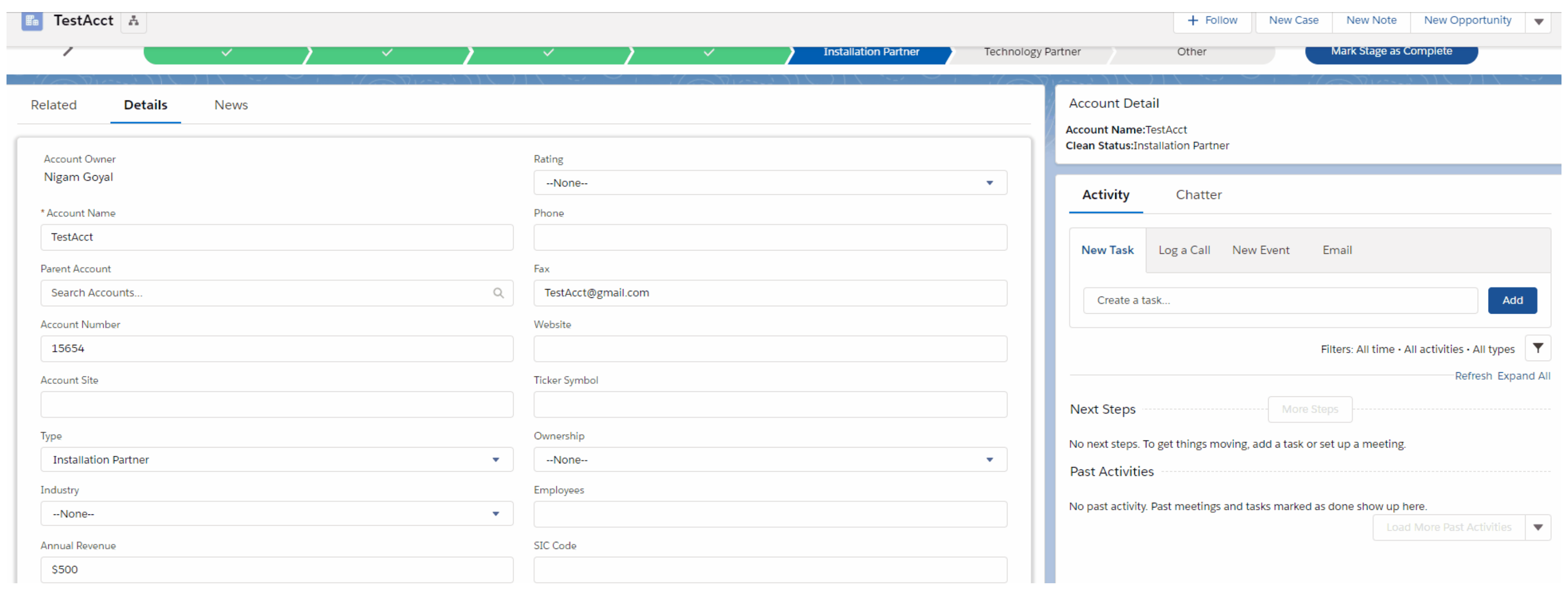
forceRefreshViewCont hosted with ❤ by **GitHub** **view raw**

```
({
        getData : function(component, event) {
        console.log('inside init getData');
        var recId = component.get("v.recordId");
```

```
                var dataAction = component.get("c.getAccount");
            dataAction.setParams({accountId : recId});
            dataAction.setCallback(this, function(response){
                var state = response.getState();
                if(state=='SUCCESS')
                {
                    var retVal = response.getReturnValue();
                    console.log('retVal---'+retVal.Name);
                    component.set("v.account",retVal);
                }

            });
                $A.enqueueAction(dataAction);
            }
    })
```

forceRefreshViewHelper hosted with ❤ by **GitHub** **view raw**

Output:



TestAcct

Installation Partner | Technology Partner | Other | Mark Stage as Complete

+ Follow | New Case | New Note | New Opportunity

Related | **Details** | News

Account Owner: Nigam Goyal	Rating: --None--
* Account Name: TestAcct	Phone:
Parent Account: Search Accounts...	Fax: TestAcct@gmail.com
Account Number: 15654	Website:
Account Site:	Ticker Symbol:
Type: Installation Partner	Ownership: --None--
Industry: --None--	Employees:
Annual Revenue: $500	SIC Code:

Account Detail

Account Name: TestAcct
Clean Status: Installation Partner

Activity | Chatter

New Task | Log a Call | New Event | Email

Create a task... Add

Filters: All time • All activities • All types

Refresh Expand All

Next Steps — More Steps

No next steps. To get things moving, add a task or set up a meeting.

Past Activities

No past activity. Past meetings and tasks marked as done show up here.

Load More Past Activities

Share this: