**Lightning Events examples**

**Components events example:**

*LightningEventsComp1:*

|  |
| --- |
| <aura:component>  <aura:attribute name="incomes" type="Object" />  <aura:attribute name="myColumns" type="List" />  <aura:attribute name="outputVal" type="Decimal" />  <aura:registerEvent name="totalIncomeComponentEvent" type="c:LightningCompEvent" />  <aura:handler name="totalIncomeComponentEvent" event="c.LightningCompEvent"  action="{!c.handleRegisteredCompEvent}" phase="bubble" />  <aura:handler name="init" value="{!this}" action="{!c.doInit}" />  <lightning:card title="SFDC income calculator" iconName="standard:account">  <aura:set attribute="actions">  <lightning:button label="Show/hide Income form" onclick="{!c.toggleIncomeForm}" />  </aura:set>  <p class="slds-p-horizontal\_small">  <div aura:Id="incomeForm" class="hide">  <lightning:input aura:id="source" name="source" required="true" label="Income source"></lightning:input>  <lightning:input aura:id="amount" name="amount" required="true" label="Amount"></lightning:input>  <br />  <lightning:button name="addIncome" label="Add Income" onclick="{!c.addIncome}" variant="brand" />  </div>  <br />  <lightning:datatable data="{!v.incomes}" columns="{!v.myColumns}" keyField="sno" />  <br />  <lightning:button name="calculateIncome" label="Calculate total income" onclick="{!c.fireTotalIncomeCompEvent}" variant="brand"></lightning:button>  </p>  </lightning:card>  </aura:component> |

*Controller:*

|  |
| --- |
| ({  doInit : function(component, event, helper) {  component.set('v.myColumns',[  {label: 'SNo', fieldName:'sno', type: 'number'},  {label:'Name of Source', fieldName:'source', type: 'text'},  {label:'Amount', fieldName:'amount', type: 'number'}  ]);  component.set('v.incomes', [{  sno: 1,  source: 'Regular Job',  amount: 10000  },  {  sno: 2,  source: 'Part time Job',  amount: 2000  }  ]);  },  handleRegisteredCompEvent : function(component, event, helper){  alert("event handler in source component");  //event.stopPropagation();  },  toggleIncomeForm : function(component, event, helper){  var incomeForm = component.find('incomeForm');  $A.util.toggleClass(incomeForm, 'hide');  },  addIncome : function(component, event, helper){  var incomes = component.get('v.incomes');  var newIncomes = {  sno : incomes.length+1,  source : component.find('source').get('v.value'),  amount : parseFloat(component.find('amount').get('v.value'))  }  if(newIncomes.source!='' && newIncomes.amount!=''&& newIncomes.source!=null && newIncomes.amount!=null){  incomes.push(newIncomes);  component.set('v.incomes', incomes);  component.set('v.source','');  component.set('v.amount','');  }  },  fireTotalIncomeCompEvent : function(component, event, helper){  var incomes=component.get('v.incomes');  var totIncome = 0;  for(var i=0; i<incomes.length; i++){  totIncome = totIncome+incomes[i].amount;  }  component.set('v.outputVal', totIncome);  var myEvent = component.getEvent('totalIncomeComponentEvent');  myEvent.setParams({totalIncome: totIncome});  myEvent.fire();  }  }) |

*LightningEventsCompContainer.cmp*

|  |
| --- |
| <aura:component>  <aura:attribute name="incomes" type="Object" />  <aura:attribute name="myColumns" type="List" />  <aura:attribute name="outputVal" type="Decimal" />  <aura:registerEvent name="totalIncomeComponentEvent" type="c:LightningCompEvent" />  <aura:handler name="totalIncomeComponentEvent" event="c.LightningCompEvent"  action="{!c.handleRegisteredCompEvent}" phase="bubble" />  <aura:handler name="init" value="{!this}" action="{!c.doInit}" />  <lightning:card title="SFDC income calculator" iconName="standard:account">  <aura:set attribute="actions">  <lightning:button label="Show/hide Income form" onclick="{!c.toggleIncomeForm}" />  </aura:set>  <p class="slds-p-horizontal\_small">  <div aura:Id="incomeForm" class="hide">  <lightning:input aura:id="source" name="source" required="true" label="Income source"></lightning:input>  <lightning:input aura:id="amount" name="amount" required="true" label="Amount"></lightning:input>  <br />  <lightning:button name="addIncome" label="Add Income" onclick="{!c.addIncome}" variant="brand" />  </div>  <br />  <lightning:datatable data="{!v.incomes}" columns="{!v.myColumns}" keyField="sno" />  <br />  <lightning:button name="calculateIncome" label="Calculate total income" onclick="{!c.fireTotalIncomeCompEvent}" variant="brand"></lightning:button>  </p>  </lightning:card>  </aura:component> |

*Controller*

|  |
| --- |
| ({  doInit : function(component, event, helper) {  component.set('v.myColumns',[  {label: 'SNo', fieldName:'sno', type: 'number'},  {label:'Name of Source', fieldName:'source', type: 'text'},  {label:'Amount', fieldName:'amount', type: 'number'}  ]);  component.set('v.incomes', [{  sno: 1,  source: 'Regular Job',  amount: 10000  },  {  sno: 2,  source: 'Part time Job',  amount: 2000  }  ]);  },  handleRegisteredCompEvent : function(component, event, helper){  alert("event handler in source component");  //event.stopPropagation();  },  toggleIncomeForm : function(component, event, helper){  var incomeForm = component.find('incomeForm');  $A.util.toggleClass(incomeForm, 'hide');  },  addIncome : function(component, event, helper){  var incomes = component.get('v.incomes');  var newIncomes = {  sno : incomes.length+1,  source : component.find('source').get('v.value'),  amount : parseFloat(component.find('amount').get('v.value'))  }  if(newIncomes.source!='' && newIncomes.amount!=''&& newIncomes.source!=null && newIncomes.amount!=null){  incomes.push(newIncomes);  component.set('v.incomes', incomes);  component.set('v.source','');  component.set('v.amount','');  }  },  fireTotalIncomeCompEvent : function(component, event, helper){  var incomes=component.get('v.incomes');  var totIncome = 0;  for(var i=0; i<incomes.length; i++){  totIncome = totIncome+incomes[i].amount;  }  component.set('v.outputVal', totIncome);  var myEvent = component.getEvent('totalIncomeComponentEvent');  myEvent.setParams({totalIncome: totIncome});  myEvent.fire();  }  }) |

*LightningEventsApp.app*

|  |
| --- |
| <aura:application extends="force:slds">  <!--<aura:handler event="c:LightningApplicationEvent" action="{!c.handleAppliationEvent}" phase="default"/>-->  <div class="slds-box">  <div class="slds-text-heading\_medium slds-text-align\_center">Component Event Below</div>  <c:LightningEventsCompContainer />  </div>  </aura:application> |

**Application Event:**

*LightningAppComp1.cmp:*

|  |
| --- |
| <aura:component implements="flexipage:availableForAllPageTypes,flexipage:availableForRecordHome,force:hasRecordId" access="global" >  <aura:attribute name="id" type="Integer" />  <aura:attribute name="message" type="String" />    <aura:registerEvent name="myAppEvent" type="c:LightningApplicationEvent" />  <aura:handler event="c:LightningApplicationEvent" action="{!c.handleAppEvent}" phase="default" />  <div class="innerComponent">  <lightning:button label="Fire Event" onclick="{!c.handleFireEvent}" />  </div>  </aura:component> |

*Controller.js:*

|  |
| --- |
| ({  handleFireEvent : function(component, event, helper){  var appEvent = $A.get('e.c:LightningApplicationEvent');  appEvent.setParams({  message:'hello'  });  appEvent.fire();  },    handleAppEvent : function(component, event, helper) {  var id=component.get('v.id');  alert('handler in source component' +id);  //event.stopPropagation();  }  }) |

*LightningEvntAppContainer.cmp*

|  |
| --- |
| <aura:component >  <aura:attribute name="id" type="Integer" />  <aura:attribute name="message" type="String" default="none" />  <aura:handler event="c:LightningApplicationEvent" action="{!c.handleAppliationEvent}" phase="default"/>    <div class="outerComponent">  <br />  Event container id={!v.id}  <br />  Message in event container is {!v.message}  <br />  <c:LightningAppComp1 id="{!v.id}" />  </div>    </aura:component> |

Controller.js:

|  |
| --- |
| ({  handleAppliationEvent : function(component, event, helper) {  var id=component.get('v.id');  alert('handler in container'+id);  component.set('v.message',event.getParam('message'));  var message = component.get('v.message');  //event.stopPropagation();  }  }) |

LightningEventsApp.app

|  |
| --- |
| <aura:application extends="force:slds">  <aura:handler event="c:LightningApplicationEvent" action="{!c.handleAppliationEvent}" phase="default"/>  <!--<div class="slds-box">  <div class="slds-text-heading\_medium slds-text-align\_center">Component Event Below</div>  <c:LightningEventsCompContainer />  </div>-->  <div class="slds-box">  <div class="slds-text-heading\_medium slds-text-align\_center">Application Event Below</div>  <c:LightningEvntAppContainer id="1" />  <br />  <c:LightningEvntAppContainer id="2" />  </div>  </aura:application> |

*Controller.js*

|  |
| --- |
| ({  handleAppliationEvent : function(component, event, helper) {  alert('event in app root');  }  }) |