

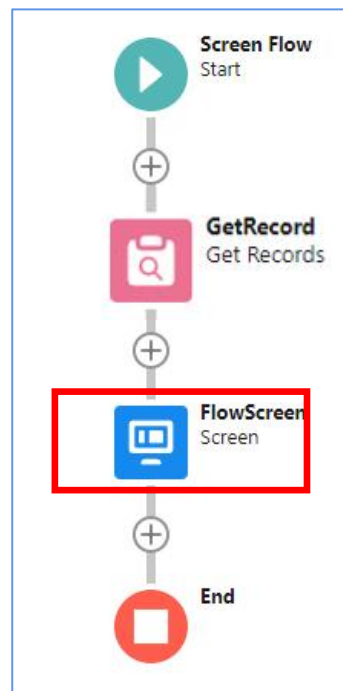
知识点(What):

①创建Flow Screen组件(主要提供交互界面)

创建一个FlowScreen组件

※Demo: 取得一条数据, 并指定Attribute

```
<aura:component  
implements="lightning:availableForFlowScreens"  
>  
<!-- Your component markup here -->  
</aura:component>
```



DreamITSchool.com

DreamITSchool.com

DreamITSchool.com

DreamITSchool.com

知识点(What):

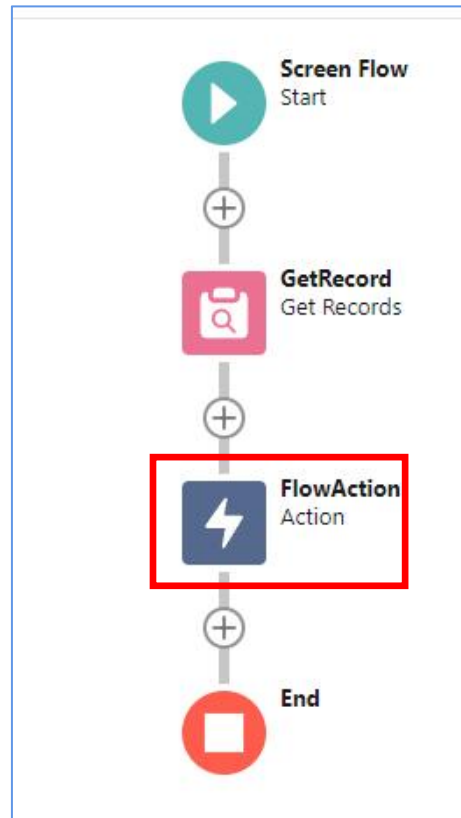
①创建Flow Action组件(不需要画面交互的处理, 比如查询数据库取得数据传给下一个画面)

创建一个FlowAction组件

※Demo: 取得一条数据, 并给Action指定Attribute, 最后执行Action

```
<aura:component  
implements="lightning:availableForFlowActions"  
>  
// declare attributes here  
</aura:component>
```

DreamITSchool.com



DreamITSchool.com

DreamITSchool.com

DreamITSchool.com

知识点(What):

①一部分Attribute可以在Flow中的Aura内使用

Aura在Flow中可使用的Attribute和Flow中的变量类型的对应关系

Flow Data Type	Lightning Component Attribute Type
Apex	Custom Apex Class
Boolean	Boolean
Currency	Number
Date	Date
Date/Time (API name is DateTime)	DateTime
Number	Number
Multi-Select Picklist	String
Picklist	String
Record, with a specified object	The API name of the specified object, such as Account or Case
Text	String

知识点(What):

①在Aura中调用Flow

需要使用Flow控件

```
<aura:component>
<aura:handler name="init" value="{!this}"
action="{!c.init}"/>
<lightning:flow aura:id="flowData"/>
</aura:component>
```

```
{
  init : function (cmp) {
    var flow = cmp.find("flowData");
    flow.startFlow("YourFlowName");
  }
}
```

```
((
  init : function(component, event, helper) {
    var flow = component.find("myFlow");
    var inputVariables = [
      {
        name: "AccName",
        type: "String",
        value: "SF Happy Corner Account"
      }
    ];
    flow.startFlow("Sample_Flow", inputVariables);
  }
})
```

知识点(What):

①在Aura中调用Flow的时候, 传递参数

创建一个FlowAction组件

```
<aura:component>
<aura:handler name="init" value="{!this}"
action="{!c.init}"/>
<lightning:flow aura:id="flowData"/>
</aura:component>
```

```
{
  init : function (cmp) {
    var flow = cmp.find("flowData");
    flow.startFlow("YourFlowName");
  }
}
```

```
((
  init : function(component, event, helper) {
    var flow = component.find("myFlow");
    var inputVariables = [
      {
        name: "AccName",
        type: "String",
        value: "SF Happy Corner Account"
      }
    ]
    flow.startFlow("Sample_Flow", inputVariables);
  }
})
```

知识点(What):

①从Flow中取得Output变量

CMP内容

```
<lightning:flow aura:id="flowData"  
onstatuschange="{!c.handleStatusChange}" />
```

DreamITSchool.com

Controller内容

```
handleStatusChange : function (component, event) {  
  if(event.getParam("status") === "FINISHED") {  
    // Get the output variables and iterate over them  
    var outputVariables =  
    event.getParam("outputVariables");  
    var outputVar;  
    for(var i = 0; i < outputVariables.length; i++) {  
      outputVar = outputVariables[i];  
      // Pass the values to the component's attributes  
      if(outputVar.name === "accountName") {  
        component.set("v.accountName", outputVar.value);  
      }  
    }  
  }  
},
```

实践(How):

①从Flow中创建一条数据, 并取得创建的数据Output变量, 并跳转到数据的详细页面

CMP内容

```
<lightning:flow aura:id="flowData"  
onstatuschange="{!c.handleStatusChange}" />
```

DreamITSchool.com

DreamITSchool.com

DreamITSchool.com

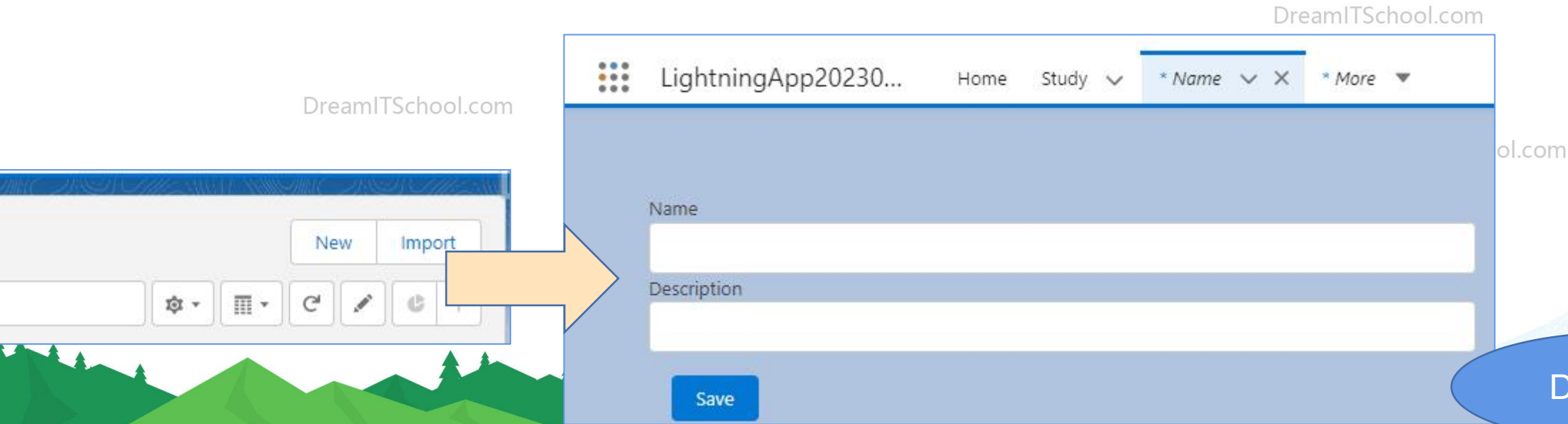
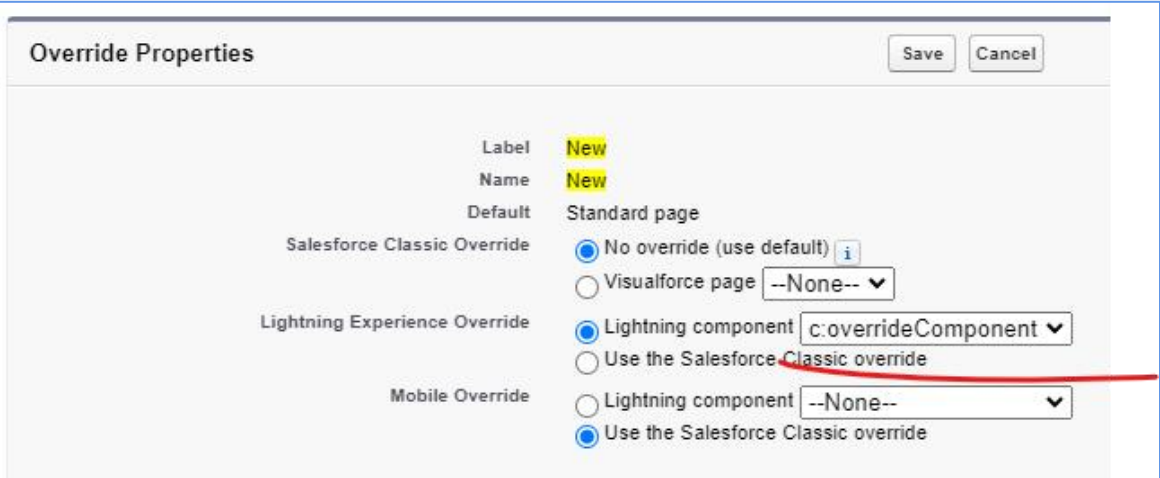
DreamITSchool.com

DreamITSchool.com

知识点(What):
使用Aura覆盖标准按钮

创建一个能够覆盖标准按钮的组件

前提条件
需要有
lightning:actionOverride



知识点(What):

①使用lightning:messageChannel传消息

CMP内容

```
<lightning:messageChannel  
type="SampleMessageChannel__c"  
aura:id="sampleMessageChannel"/>
```

```
// publisherController.js  
{  
  handleClick: function(cmp, event, helper) {  
    var payload = {  
      recordId: "some string",  
      recordData: {  
        value: "some value"  
      }  
    };  
    cmp.find("sampleMessageChannel").publish(payload);  
  }  
}
```

DreamITSchool.com

DreamITSchool.com

DreamITSchool.com

```
<-- subscriberComponent.cmp -->
<aura:component>
  <aura:attribute name="recordValue"
type="String"/>
  <lightning:formattedText value="{!v.recordValue}" />
  <lightning:messageChannel
type="SampleMessageChannel__c"
  onMessage="{!c.handleChanged}"
scope="APPLICATION"/>
</aura:component>
```

```
((
  handleChanged: function(cmp, message, helper) {
    // Read the message argument to get the values in the
    message payload
    if (message != null &&
message.getParam("recordData") != null) {
      cmp.set("v.recordValue",
message.getParam("recordData").value);
    }
  }
})
```