

# [Salese Force] Salese Force 의 Lightnig Platform 사용 방법\_1(페이지 설명) — 나무늘보의 개발 블로그

🌐 웹 클립

Base

## [Salese Force] Salese Force 의 Lightnig Platform 사용 방법\_1(페이지 설명)

2021. 5. 21. 21:46 수정 삭제 공개



## 1.Lightning Resource 설명

Resource

Resource Name

Usage

Component or Application	sample.cmp or sample.app	번들에서 유일한 필수 리소스입니다. 구성 요소 또는 앱에 대한 마크 업을 포함합니다. 각 번들에는 하나의 구성 요소 또는 앱 리소스 만 포함됩니다.
CSS Styles	sample.css	구성 요소의 스타일을 포함합니다.
Controller	sampleController.js	구성 요소의 이벤트를 처리하는 클라이언트 측 컨트롤러 메서드를 포함합니다.
Design	sample.design	Lightning App Builder, Lightning 페이지, Community Builder 또는 Cloud Flow Designer에서 사용되는 구성 요소에 필요한 파일입니다.
Documentation	sample.auradoc	설명, 샘플 코드 및 예제 구성 요소에 대한 하나 이상의 참
Renderer	sampleRenderer.js	구성 요소의 기본 렌더링을 재정의하는 클라이언트 측 렌더러입니다.
Helper	sampleHelper.js	구성 요소 번들의 JavaScript 코드에서 호출 할 수있는 JavaScript 함수
SVG File	sample.svg	Lightning 앱 빌더 또는 커뮤니티 빌더에서 사용되는 구성 요소에 대한 맞춤 아이콘 리소스입니다.

## Component page

```
1 <aura:component>
2     Hello, world!
3 </aura:component>
```

```
1 <aura:component>
2     <div class="container">
3         <!--Other HTML tags or components here-->
4     </div>
5 </aura:component>
```

구성 요소 리소스에는 마크 업이 포함되며 \* .cmp 접미사가 있습니다.

마크 업에는 다른 구성 요소에 대한 텍스트 또는 참조가 포함될 수 있으며 구성 요소에 대한 메타 데이터도 선언됩니다.

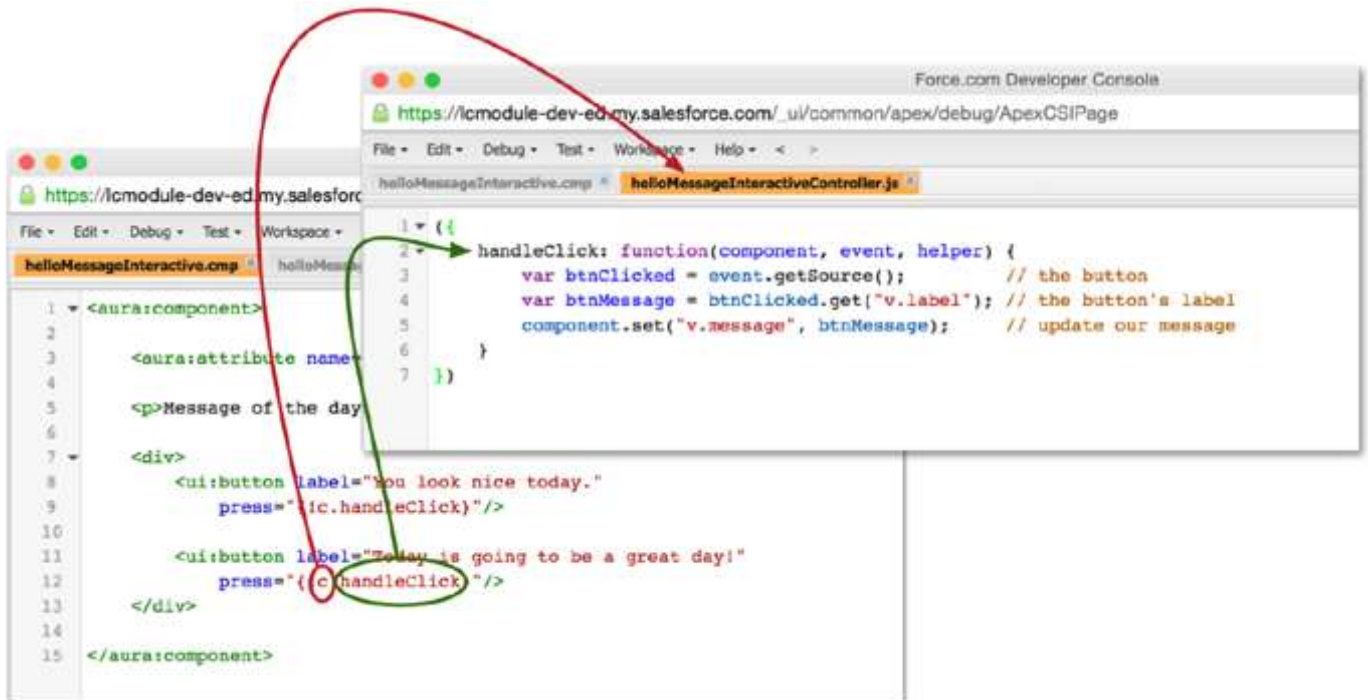
## Controller page

```
1 ({
2     myAction : function(cmp, event, helper) {
3         // add code for the action
4     },
5
6     anotherAction : function(cmp, event, helper) {
7         // add code for the action
8     }
9 })
```

클라이언트 측 컨트롤러는 구성 요소 내에서 이벤트를 처리하고 \* Controller.js 접미사를 갖습니다.

모든 구성 요소 작업에 대한 기능을 정의하는 자바 스크립트 리소스입니다.

## Controller page



`press = "{! c.handleClick}"`

c. 'handleClick'속성이있는 값 제공 업체입니다.

c는 구성 요소의 클라이언트 측 컨트롤러에 대한 값 공급자이고 handleClick은 해당 컨트롤러에 정의 된 함수입니다.

`{! c.handleClick}`은 구성 요소 컨트롤러의 작업 핸들러에 대한 참조입니다.

## Helper page

```
01 ({
02   helperMethod1 : function() {
03     // logic here
04   },
05   helperMethod2 : function(component) {
06     // logic here
07     this.helperMethod3(var1, var2);
08   },
09   helperMethod3 : function(var1, var2) {
10     // do something with var1 and var2 here
11   }
12 })
```

재사용하려는 함수를 구성 요소의 도우미에 넣고 \* Helper.js 접미사를 붙입니다.

도우미 기능은 또한 데이터 처리 및 서버 측 작업 실행과 같은 작업의 전문화를 가능하게 합니다.

## Style page

```
01 <aura:component>
02   <div class="white">
03     Hello, HTML!
04   </div>
05
06   <h2>Check out the style in this list.</h2>
07
08   <ul>
09     <li class="red">I'm red.</li>
10     <li class="blue">I'm blue.</li>
11     <li class="green">I'm green.</li>
12   </ul>
13 </aura:component>
```

```
01 .THIS {
02   background-color: grey;
03 }
04
05 .THIS.white {
06   background-color: white;
07 }
08
09 .THIS .red {
10   background-color: red;
11 }
12
13 .THIS .blue {
14   background-color: blue;
15 }
16
17 .THIS .green {
18   background-color: green;
19 }
```

CSS가 있고 \* .CSS 접미사가 있습니다.

## Design page

```
1 <design:component label="Hello World">
2   <design:attribute name="subject" label="Subject" description="Name of t
3   <design:attribute name="greeting" label="Greeting" />
4 </design:component>
```

디자인 리소스를 사용하여 Lightning App Builder, Community Builder 또는 Cloud Flow Designer와 같은 빌더 도구에 노출되는 속성을 제어합니다.

\* .design 접미사가 있습니다.

## APEX page

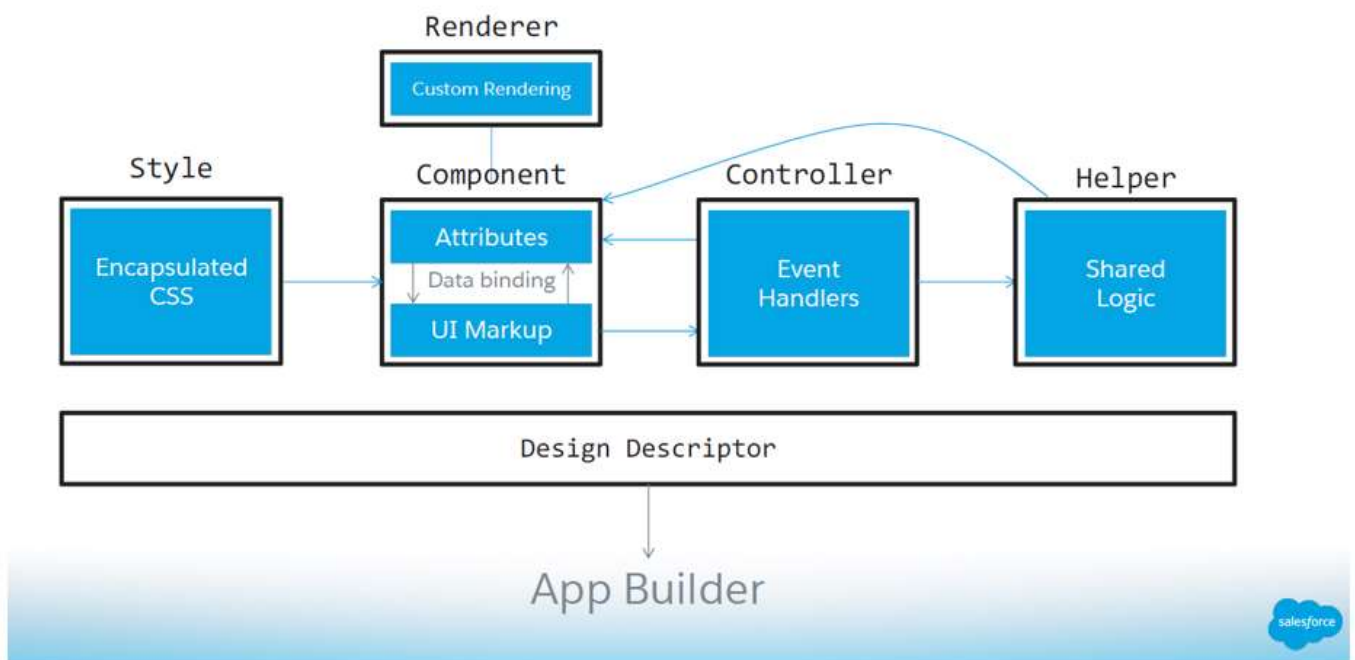
```

... /* *****_
... * Get Account Information_
... ***** */_
... @AuraEnabled_
... public static Account gf_GetAccInformation (String accId) {_
...     return [SELECT id, name FROM Account WHERE id = :accId];_
... }_

```

Apex는 개발자가 API 호출과 함께 Salesforce 서버에서 흐름 및 트랜잭션 제어 문을 실행할 수 있도록하는 강력한 형식의 개체 지향 프로그래밍 언어입니다.

## Component Framework Deep Dive



'Base' 카테고리의 다른 글

[Sales Force] Salese Force 의 Lightnig Platform 사용 방법\_1(페이지 설명)



나아무늘보

혼자 끄적끄적하는 블로그 입니다.

