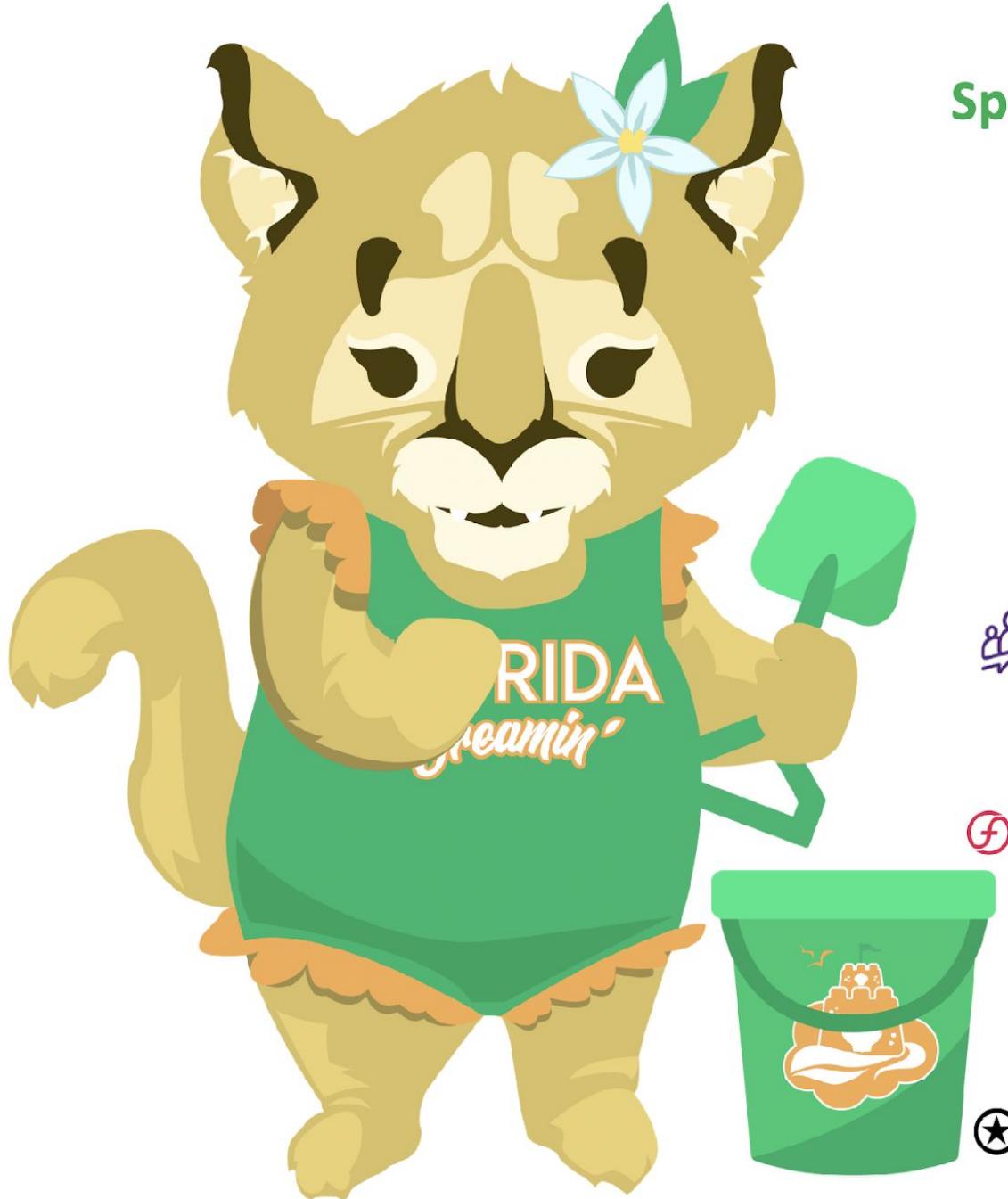


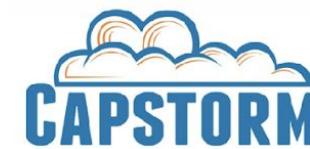
Lightning Components made easy with Lightning Data Service and Base Lightning Components

Phillip Southern
phillip@clicks2code.com
@phil7s





Special Thanks to Our Platinum and Gold Sponsors



Who is this?

and other interesting facts



phillip@clicks2code.com
@phil7s



Salesforce MVP



Salesforce Certified



A bit about me.....



visual*force*TM



Love at first site



- Standard Controller
- Extensions
- Tables
- Styling
- Javascript



Love at first site



Portals / Communities

Edge Cases

Complex Functionality



Then what happened was....



Lightning Component - It's Easy!

(not really)



1. Apex Controller - FOR ANY DATA
2. Pass any data to JS Controller
3. Build HTML Tags
4. Iterate Data

```
<apex:page lightningStylesheets="true">  
    <apex:slds />
```



Let's Talk

1. Terminology
2. What the Component
3. Base Components
4. Lightning Data Service
5. Demo Time!



Terminology

Lightning Component

Aura/Web

Lightning Data Service / Lightning Service

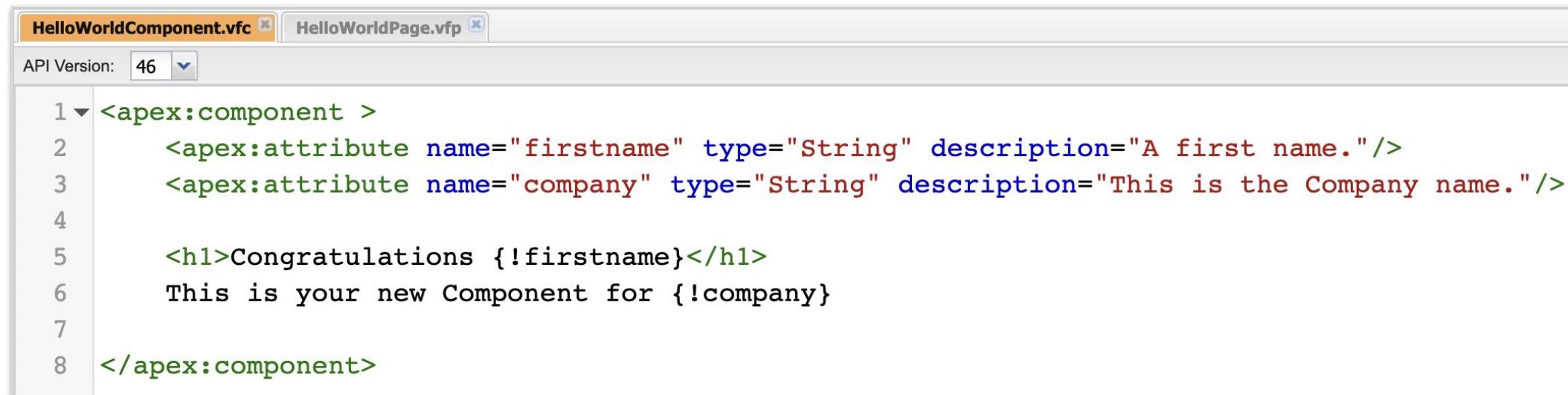
Base Component / Base Lightning Component

Visualforce Component / Visualforce Standard Component



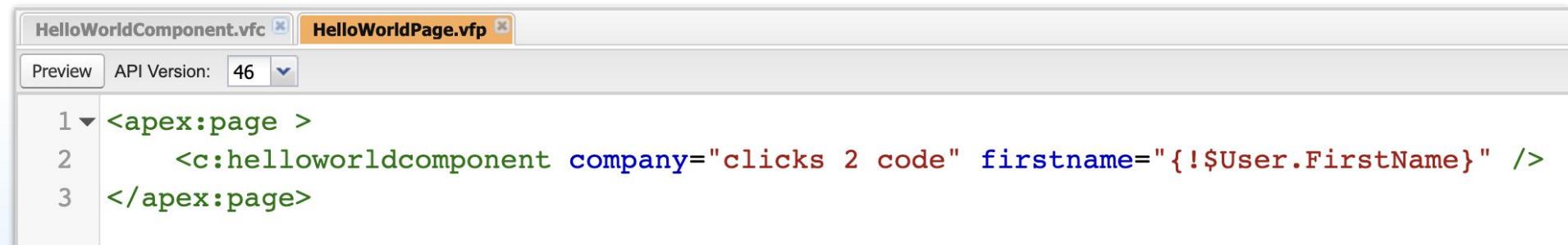
What the component....

Visualforce



The screenshot shows the Visualforce Component Editor interface. The tab bar at the top has 'HelloWorldComponent.vfc' selected. Below the tabs is a dropdown for 'API Version' set to 46. The main area contains the following Apex code:

```
1 <apex:component>
2     <apex:attribute name="firstname" type="String" description="A first name."/>
3     <apex:attribute name="company" type="String" description="This is the Company name."/>
4
5     <h1>Congratulations {!firstname}</h1>
6     This is your new Component for {!company}
7
8 </apex:component>
```



The screenshot shows the Visualforce Page Editor interface. The tab bar at the top has 'HelloWorldPage.vfp' selected. Below the tabs is a dropdown for 'API Version' set to 46. The main area contains the following Apex code:

```
1 <apex:page>
2     <c:helloworldcomponent company="clicks 2 code" firstname="{!$User.FirstName}" />
3 </apex:page>
```



What the component....

Visualforce Standard Components

Standard Component Reference

- analytics:reportChart
- apex:actionFunction
- apex:actionPoller
- apex:actionRegion
- apex:actionStatus
- apex:actionSupport
- apex:areaSeries
- apex:attribute
- apex:axis
- apex:barSeries
- apex:canvasApp
- apex:chart
- apex:chartLabel
- apex:chartTips
- apex:column

AccountContacts.vfp

Preview API Version: 46

```
1 <apex:page standardController="Account">
2   <apex:pageBlock title="Contacts">
3     <apex:pageBlockTable value="{!account.Contacts}" var="contact">
4       <apex:column value="{!!contact.name}"/>
5       <apex:column value="{!!contact.email}"/>
6       <apex:column value="{!!contact.phone}"/>
7     </apex:pageBlockTable>
8   </apex:pageBlock>
9 </apex:page>
```

What the component....

Visualforce Standard Components

```
AccountContacts.vfp
Preview API Version: 46
1 <apex:page standardController="Account">
2   <apex:pageBlock title="Contacts">
3     <apex:pageBlockTable value="{!account.Contacts}" var="contact">
4       <apex:column value="{!contact.name}" />
5       <apex:column value="{!contact.email}" />
6       <apex:column value="{!contact.phone}" />
7     </apex:pageBlockTable>
8   </apex:pageBlock>
9 </apex:page>
```

Contacts		
Name	Email	Phone
Stella Pavlova	spavlova@uog.com	(212) 842-5500
Lauren Boyle	lboyle@uog.com	(212) 842-5500
Arthur Song	asong@uog.com	(212) 842-5500
Avi Green	agreen@uog.com	(212) 842-5500

Wait...Aura Web what?

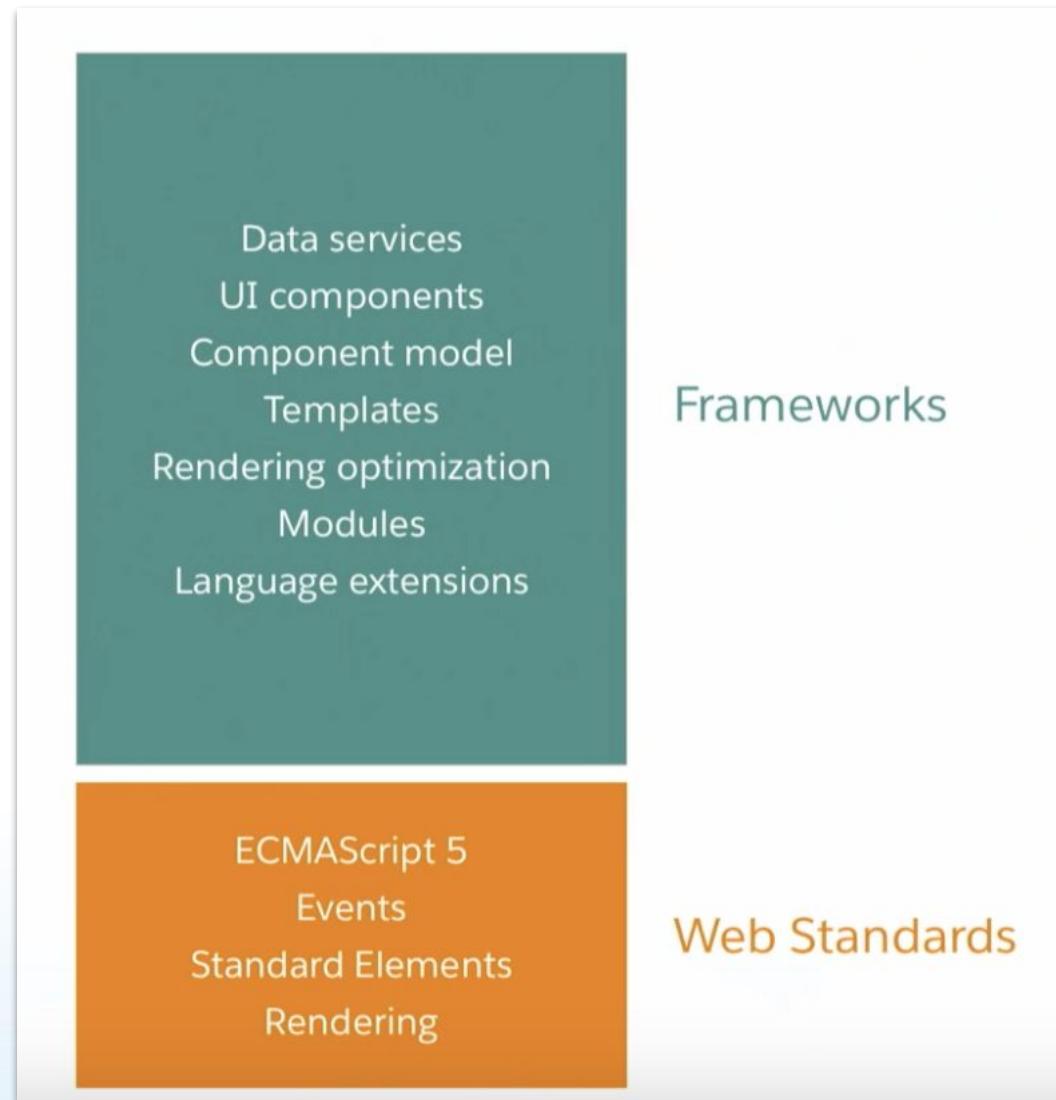
Lightning Component

	Aura	Web
Introduced	Dreamforce '13	Spring '19
Tooling	Developer Console and multiple IDE's	VS Code
Base Components	Yes	Yes
Data Service	Yes	Yes
Open Source	Yes	Yes
Web Standards	Some	Many



Wait...Aura Web what?

2014 Web Stack



Wait...Aura Web what?

Today



Base Lightning Components

Building blocks that make up the modern Lightning Experience, Salesforce app, and Lightning Communities user interfaces.

Incorporate Lightning Design System.

Handle the details of HTML and CSS for you.



Base Lightning Components

Namespace

All have the “lightning” namespace.

Framework	Button
Visualforce	<code><apex:commandButton action="{!save}" value="Save" id="theButton"/></code>
Aura	<code><lightning:button label="Neutral" title="Neutral action" onclick="{!! c.handleClick }"/></code>
Web	<code><lightning-button label="Neutral" title="Neutral action" onclick={handleClick}></lightning-button></code>



Base Lightning Components

A Timeline

Component	Release
Beta Release	Winter '17
lightning:datatable	Winter '18
lightning:inputField	Spring '18
lightning:recordForm	Summer '18



Lightning Component

Wait we had these.....

```
<force:inputField value=" {!v.contact.Name}"/>
```



Lightning Data Service

Use to load, create, edit, or delete a record in your component without requiring Apex code.

Handles sharing rules and field-level security for you.

Improves performance and user interface consistency.



Lightning Data Service

How do we do this?

```
<aura:attribute name="accountRecord" type="Object"/>

<force:recordData aura:id="recordLoader"
  recordId="{!!v.recordId}"
  fields="Name,Description,Phone,Industry"
  targetFields="{!!v.accountRecord}"
/>
```

Lightning Data Service

How do we do this?

```
<aura:attribute name="accountRecord" type="Object"/>

<force:recordData aura:id="recordLoader"
recordId="{!!v.recordId}"
fields="Name,Description,Phone,Industry"
targetFields="{!!v.accountRecord}"
/>

<div>
  <lightning:card iconName="standard:account" title="{!!v.accountRecord.Name}" >
    <div class="slds-p-horizontal--small">
      <p class="slds-text-heading--medium"><lightning:formattedPhone title="Phone" value="{!!v.accountRecord.Phone}" /></p>
      <p class="slds-truncate"><lightning:formattedText title="Description" value="{!!v.accountRecord.Description}" /></p>
      <p class="slds-truncate"> <lightning:formattedText title="Industry" value="{!!v.accountRecord.Industry}" /></p>
    </div>
  </lightning:card>
</div>
```

Lightning Data Service

How do we do this?

lightning:recordForm

lightning:recordViewForm

lightning:recordEditForm



Lightning Data Service

Which do we use?

Do I need a form?

Yes

`lightning:recordForm`

No

`lightning:recordViewForm`

`force:recordData`

`lightning:recordEditForm`



Code

Record Component

Base Components + Data Service

So what's the big deal?

Before we had to:

1. Apex Controller
2. Pass any data to JS Controller
3. Build HTML Tags
4. Iterate Data



!(Base Components + Data Service)

So what's the big deal?

Apex Class

```
public with sharing class ContactController {  
    @AuraEnabled  
    public static Contact getContact() {  
        return [select Id, Name from Contact Limit 1];  
    }  
}
```



!(Base Components + Data Service)

So what's the big deal?

Component Javascript Controller

```
{  
    doInit : function(component, event, helper) {  
        var action = component.get("c.getContact");  
        action.setCallback(this, function(response) {  
            var state = response.getState();  
            if (state === "SUCCESS") {  
                component.set("v.contact", response.getReturnValue());  
                console.log(response.getReturnValue());  
            }  
        });  
        $A.enqueueAction(action);  
    }  
}
```



!(Base Components + Data Service)

So what's the big deal?

Component

```
<aura:component controller="ContactController">
    <aura:attribute name="contact" type="Contact"
        default="{ 'sobjectType': 'Contact' }"/>
    <aura:handler name="init" value="{!this}" action=" {!c.doInit} " />
    <force:inputField value=" {!v.contact.Name} "/>
</aura:component>
```



Base Components + Data Service

So what's the big deal?

```
<aura:component implements="flexipage:availableForRecordHome,force:hasRecordId">
    <aura:attribute name="fields" type="String[]" default="['Name','Title','Email']" />
    <lightning:recordForm
        recordId="{!v.recordId}"
        objectApiName="Contact"
        fields="{!v.fields}"
        columns="1"
        mode="edit"/>
</aura:component>
```



Base Components + Data Service

So what's the big deal?

```
<aura:component implements="flexipage:availableForRecordHome,force:hasRecordId">
  <lightning:recordEditForm recordId="{!v.recordId}" objectApiName="Contact">
    <lightning:messages />
    <lightning:inputField fieldName="Name" />
    <lightning:inputField fieldName="Title" />
    <lightning:inputField fieldName="Email" />
    <lightning:button class="slds-m-top_small" type="submit" name="update" label="Update" />
  </lightning:recordEditForm>
</aura:component>
```

Base Components + Data Service

So what's the big deal?

* Full Name

Salutation

Mr.

First Name

Jack

* Last Name

Rogers

Title

VP, Facilities.

Email

jrogers@burlington.com

Base Components + Data Service

So what's the big deal?

```
<template>
  <lightning-record-form
    record-id={recordId}
    object-api-name={objectApiName}
    fields={fields}
    columns="1"
    mode="edit"
  >
</lightning-record-form>
</template>
```



Base Components + Data Service

So what's the big deal?

```
import { LightningElement, api } from 'lwc';

import NAME_FIELD from '@salesforce/schema/Contact.Name';
import TITLE_FIELD from '@salesforce/schema/Contact.Title';
import EMAIL_FIELD from '@salesforce/schema/Contact.Email';

export default class ContactQuickEditLWC extends LightningElement {
    @api recordId;
    @api objectApiName;

    fields = [NAME_FIELD, TITLE_FIELD, EMAIL_FIELD];
}
```



Base Components + Data Service

So what's the big deal?

* Full Name

Salutation

Mr.

First Name

Jack

* Last Name

Rogers

Title

VP, Facilities.

Email

jrogers@burlington.com

Cancel

Save

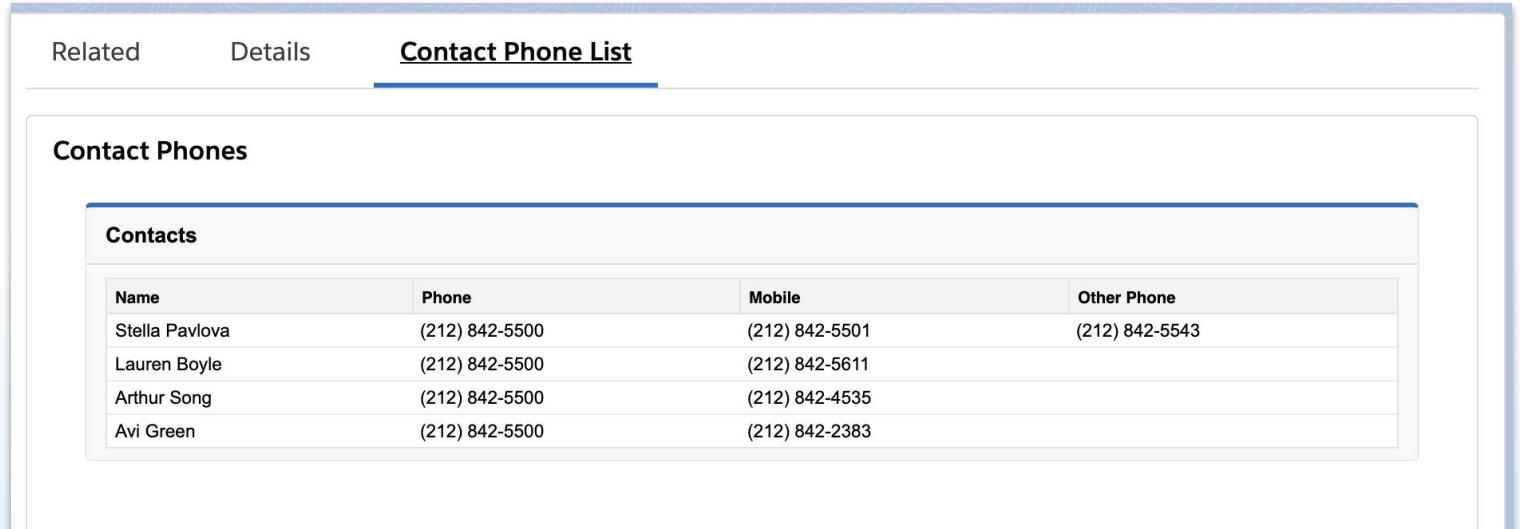


Code

Visualforce Convert

Migrate Visualforce

```
<apex:page standardController="Account">
    <apex:pageBlock title="Contacts">
        <apex:pageBlockTable value="{!!account.Contacts}" var="contact">
            <apex:column value="{!!contact.name}"/>
            <apex:column value="{!!contact.phone}"/>
            <apex:column value="{!!contact.mobilephone}"/>
            <apex:column value="{!!contact.otherphone}"/>
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:page>
```



The screenshot shows a Salesforce Visualforce page for a contact record. The page has a tab bar with 'Related', 'Details', and 'Contact Phone List' tabs. The 'Contact Phone List' tab is active, indicated by a blue underline. Below the tabs, there's a section titled 'Contact Phones'. A table titled 'Contacts' lists four contacts with their respective phone numbers:

Name	Phone	Mobile	Other Phone
Stella Pavlova	(212) 842-5500	(212) 842-5501	(212) 842-5543
Lauren Boyle	(212) 842-5500	(212) 842-5611	
Arthur Song	(212) 842-5500	(212) 842-4535	
Avi Green	(212) 842-5500	(212) 842-2383	

Migrate Visualforce

```
<apex:page standardController="Account" lightningStylesheets="true">
    <apex:pageBlock title="Contacts">
        <apex:pageBlockTable value="{!!account.Contacts}" var="contact">
            <apex:column value="{!!contact.name}"/>
            <apex:column value="{!!contact.phone}"/>
            <apex:column value="{!!contact.mobilephone}"/>
            <apex:column value="{!!contact.otherphone}"/>
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:page>
```

NAME	PHONE	MOBILE	OTHER PHONE
Stella Pavlova	(212) 842-5500	(212) 842-5501	(212) 842-5543
Lauren Boyle	(212) 842-5500	(212) 842-5611	
Arthur Song	(212) 842-5500	(212) 842-4535	
Avi Green	(212) 842-5500	(212) 842-2383	

Migrate Visualforce

Apex Class

```
public with sharing class ContactPhoneListController
{
    @AuraEnabled(cacheable=true)
    public static List<Contact> getContacts(Id accountId)
    {
        return [Select Id, Name, Phone, MobilePhone, OtherPhone From Contact Where AccountId = :accountId];
    }
}
```



Migrate Visualforce

Component Javascript Controller

```
{{
    myAction : function(component, event, helper) {
        var action = component.get("c.getContacts");
        component.set("v.Columns", [
            {label:"Name", fieldName:"Name", type:"text"},
            {label:"Phone", fieldName:"Phone", type:"phone"},
            {label:"Mobile", fieldName:"MobilePhone", type:"phone"},
            {label:"Other", fieldName:"OtherPhone", type:"phone"}
        ]);

        action.setParams({
            accountId: component.get("v.recordId")
        });
        action.setCallback(this, function(data) {
            component.set("v.Contacts", data.getReturnValue());
        });
        $A.enqueueAction(action);
    }
}}
```

Migrate Visualforce

Component

```
<aura:component implements="flexipage:availableForRecordHome,force:hasRecordId" controller="ContactPhoneListController" access="global" >
    <aura:attribute name="Contacts" type="Contact" />
    <aura:attribute name="Columns" type="List" />

    <aura:handler name="init" value="{!this}" action="{!c.myAction}" />

    <lightning:datatable data="{!!v.Contacts}" columns="{!! v.Columns}" keyField="Id" hideCheckboxColumn="true"/>

</aura:component>
```



Migrate Visualforce

Account
United Oil & Gas Corp.

Type Customer - Direct	Phone (212) 842-5500	Website http://www.uos.com	Account Owner  Phillip Southern	Account Site
---------------------------	-------------------------	--	---	--------------

Related **Details** **Contact Phone List**

NAME	PHONE	MOBILE	OTHER
Stella Pavlova	(212) 842-5500	(212) 842-5501	(212) 842-5543
Lauren Boyle	(212) 842-5500	(212) 842-5611	
Arthur Song	(212) 842-5500	(212) 842-4535	
Avi Green	(212) 842-5500	(212) 842-2383	

Migrate Visualforce

Apex Class

```
public with sharing class ContactPhoneListController
{
    @AuraEnabled(cacheable=true)
    public static List<Contact> getContacts(Id accountId)
    {
        return [Select Id, Name, Phone, MobilePhone, OtherPhone From Contact Where AccountId = :accountid];
    }
}
```



Migrate Visualforce

Javascript

```
import { LightningElement, track, wire, api } from 'lwc';
import getContacts from '@salesforce/apex/ContactPhoneListController.getContacts';

const columns = [
    { label: 'Name', fieldName: 'Name' },
    { label: 'Phone', fieldName: 'Phone', type: 'phone' },
    { label: 'Mobile', fieldName: 'MobilePhone', type: 'phone' },
    { label: 'Other', fieldName: 'OtherPhone', type: 'phone' }
];

export default class ContactPhoneListLWC extends LightningElement {
    @api recordId;
    @track columns = columns;
    @wire(getContacts, {accountId:$recordId}) contacts;
}
```

Migrate Visualforce

HTML

```
<template>
    <lightning-datatable
        key-field="id"
        data={contacts.data}
        columns={columns}
        hide-checkbox-column=true>
        </lightning-datatable>
</template>
```



Migrate Visualforce

Account
United Oil & Gas Corp.

Type Customer - Direct	Phone (212) 842-5500	Website http://www.uos.com	Account Owner  Phillip Southern	Account Site
---------------------------	-------------------------	--	---	--------------

Related	Details	<u>Contact Phone List</u>		
NAME	▼ PHONE	▼ MOBILE	▼ OTHER	▼
Stella Pavlova	(212) 842-5500	(212) 842-5501	(212) 842-5543	
Lauren Boyle	(212) 842-5500	(212) 842-5611		
Arthur Song	(212) 842-5500	(212) 842-4535		
Avi Green	(212) 842-5500	(212) 842-2383		

Code

Complex Base Components

Aura Component
Tree Grid

[View as Lightning Web Component](#)

Displays a hierarchical view of data in a table. This component requires API version 42.0 and later.

Descriptor
lightning:treeGrid

Experience
Lightning Experience, Lightning Communities, Lightning Out / Visualforce, Standalone Lightning App

Related	Details	Contact Phone List	<u>Case Hierarchy</u>
NAME	SUBJECT	STATUS	
▼ United Oil & Gas Corp.			▼
> 00001001	Performance inadequate for second consecutive w...	Closed	
> 00001002	Seeking guidance on electrical wiring installation f...	New	
> 00001021	Generator GC3060 platform structure is weakening	Closed	
> 00001022	Signal panel on GC5060 blinks intermittently	Closed	
> 00001023	Electric surge damaging adjacent equipment	Closed	
> 00001024	Design issue with mechanical rotor	New	
> 00001025	Motor design hindering performance	Closed	
▼ Burlington Textiles Corp of ...			
> 00001019	Structural failure of generator base	Closed	
> 00001020	Power generation below stated level	Closed	



```
<aura:component controller="AccountCaseHierarchyController" implements="flexipage:availableForRecordHome,force:hasRecordId" access="global" >
    <aura:handler name="init" value="{!this}" action=" {!c.init}" />
    <aura:attribute name="gridColumns" type="List" access="PRIVATE" />
    <aura:attribute name="gridData" type="Object" access="PRIVATE" />

    <lightning:treeGrid
        columns=" {!v.gridColumns}"
        data=" {!v.gridData}"
        keyField="recordId"
        hideCheckboxColumn="true"
        aura:id="casetree"
    />
</aura:component>
```



Lightning Web Component **Map**

[View as Aura Component](#)

Displays a map with markers for one or more locations.

Descriptor
lightning-map

Experience
Lightning Experience, Lightning Communities, Salesforce Mobile App, Standalone Lightning App



```
public class contactLocationsController
{
    @AuraEnabled(cacheable=true)
    public static List<Contact> getContacts(Id recordId)
    {
        return [Select Id, Name, MailingStreet, MailingCity, MailingState, MailingPostalCode
                from Contact
                Where AccountId = :recordId
                AND MailingStreet != null
                AND MailingState != null
                AND MailingPostalCode != null];
    }
}
```



```
<template>
  <lightning-map map-markers={mapMarkers} zoom-level={zoomLevel} list-view={listView}>
    </lightning-map>
</template>
```



```
import { LightningElement, api, track, wire } from 'lwc';
import getContacts from '@salesforce/apex/contactLocationsController.getContacts';

export default class ContactLocationsWebComponent extends LightningElement {
    @api recordId;
    @track mapMarkers = [];
    @track errors;
    listView = 'visible';

    @wire(getContacts, {recordId:'$recordId'})
    wiredContacts({error,data})
    {
        if(data)
        {
            data.forEach(dataItem => {
                this.mapMarkers = [...this.mapMarkers ,
                {
                    location: {
                        Street: dataItem.MailingStreet,
                        City: dataItem.MailingCity,
                        State: dataItem.MailingState,
                        PostalCode: dataItem.MailingPostalCode,
                    },
                    icon: 'standard:account',
                    title: dataItem.Name,
                    description: dataItem.Name + ' in ' + dataItem.MailingState,
                }];
            });
            this.errors = undefined;
        }
        else if(error)
        {
            window.console.log(error);
        }
    }
}
```

```
import { LightningElement, api, track, wire } from 'lwc';
import getContacts from '@salesforce/apex/contactLocationsController.getContacts';

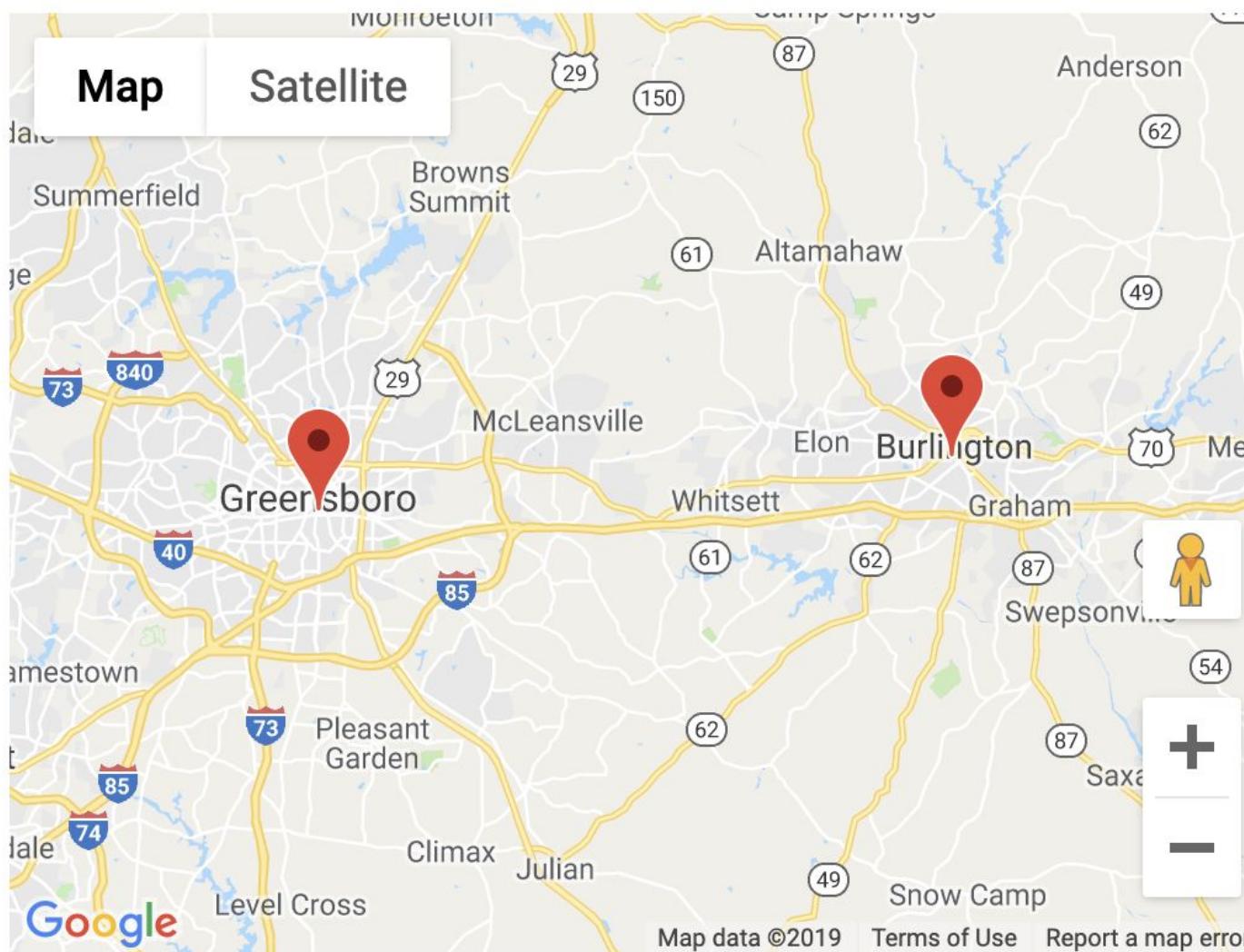
export default class ContactLocationsWebComponent extends LightningElement {
    @api recordId;
    @track mapMarkers = [];
    @track errors;
    listView = 'visible';

    @wire(getContacts, {recordId:$recordId})
    wiredContacts({error,data})
    {
        if(error)
        {
            this.errors.push(error);
        }
        else
        {
            this.mapMarkers = data;
        }
    }
}
```



```
@wire(getContacts, {recordId:'$recordId'})
wiredContacts({error,data})
{
    if(data)
    {
        data.forEach(dataItem => {
            this.mapMarkers = [...this.mapMarkers ,
            {
                location: {
                    Street: dataItem.MailingStreet,
                    City: dataItem.MailingCity,
                    State: dataItem.MailingState,
                    PostalCode: dataItem.MailingPostalCode,
                },
                icon: 'standard:account',
                title: dataItem.Name,
                description: dataItem.Name + ' in ' + dataItem.MailingState,
            };
        });
    }
}
```



[Related](#)[Details](#)[Contact Phone List](#)[Case Hierarchy](#)[**Map Contacts**](#)

Markers (2)

**Jack Rogers**

525 S. Lexington Ave, Burlington, NC

**Phillip Southern**

111 W Lewis St, Greensboro, NC

Resources

<https://developer.salesforce.com/docs/component-library/overview/components>

<https://github.com/phillipsouthern/LC-with-LDS-and-BLC>

<https://github.com/trailheadapps/lwc-recipes>

<https://trailhead.salesforce.com/content/learn/modules/lightning-web-components-basics>



#SED2020

southeastdreamin.com

SOUTHEAST DREAMIN'

SALESFORCE SOUTHERN STYLE



WHEN

March 26-27, 2020



WHERE

Atlanta, GA
Atlanta Marriott Buckhead



[← Session Details](#)

Lightning Components made easy with Lightning Data Service and Base Components

September 13, 11:30 AM - 12:10 PM

Codey Theatre

★ ★ ★ ★ ★ No Ratings

DESCRIPTION

As developers, we've hesitated moving to Lightning Components for fear of rebuilding the wheel. Life with Visualforce has been nice, especially with StandardController and Visualforce Components. The good news is, Lightning Data Service and Base

[Show More](#)

SPEAKERS



Phillip Southern

Founder
clicks 2 code



Thank You

