

# Chuck Norris your Contacts

- My first lightning component

Simon Lawrence and Edith Valencia-Martinez



dreamOlé<sup>®</sup>

Madrid 1st June 2017



# Simon Lawrence

@srlawr

Senior Developer, Desynit  
Salesforce MVP, Bristol DUG Leader



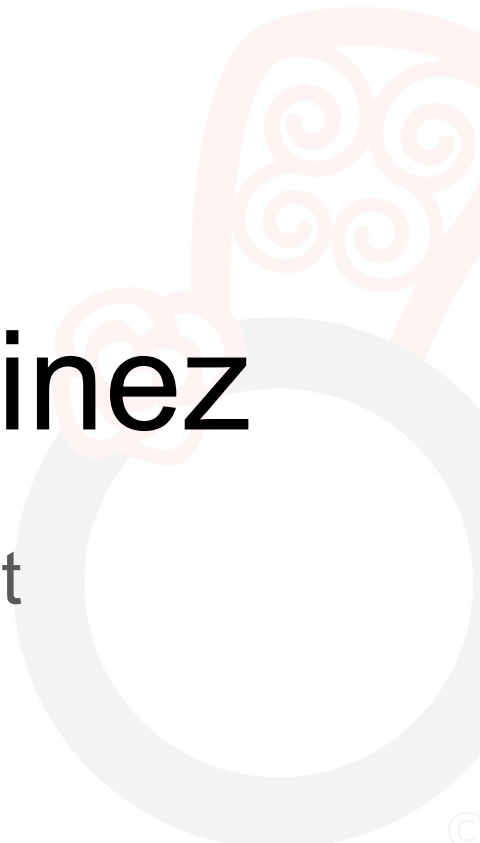


# Edith Valencia-Martinez

@edith\_mxuk

Salesforce Developer, Desynit

Spanish Speaker



# Welcome

## Baseline of understanding

### Comfortable with Salesforce UI

- Setup menu
- Developer console

### Familiar with basic programming

- Apex
- Visualforce
- Javascript

### Understand what a Lightning Component is

### Have a developer edition



# Running Order

Introduction

Lightning Applications

Worksheet time (30 minutes)

The Result

Q&A



# Introduction

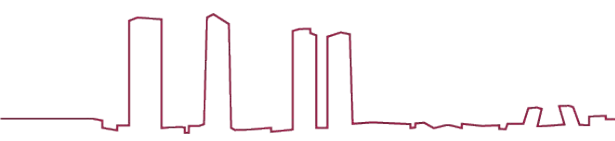
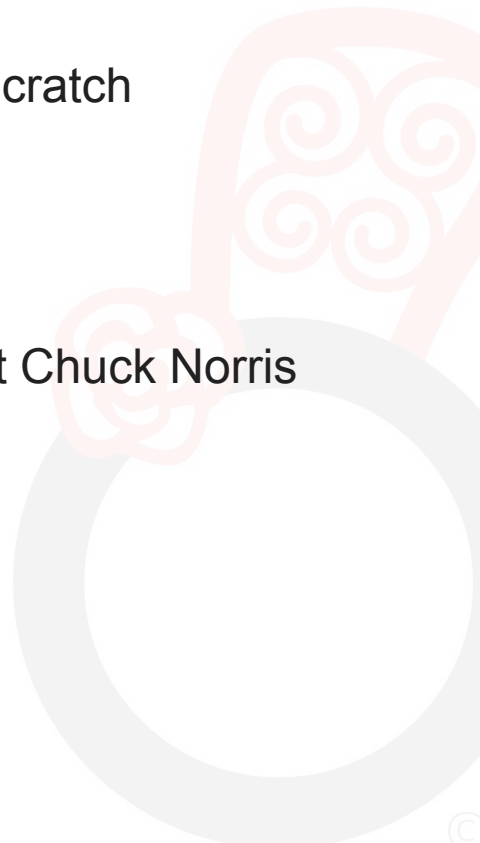
We are going to build a brand new Lightning Component from scratch

Appears on the Contact page

On initialisation, makes a synchronous callout to the Internet Chuck Norris Database API with the Contact's name

Displays the resultant quote on the Contact page

This result is in Salesforce classic - extention topic for LEX



# Introduction

What is a Chuck Norris quote?



**Chuck Norris Jokes** @ChuckNorrsJokes · 5 Feb 2016

"Chu



**Chuck Norris Jokes** @ChuckNorrsJokes · 4 Feb 2016

"Chuck Norris played Russian Roulette



**Chuck Norris Jokes** @ChuckNorrsJokes · 4 Feb 2016



**Chuck Norris Jokes** @ChuckNorrsJokes · 16 Jan 2016

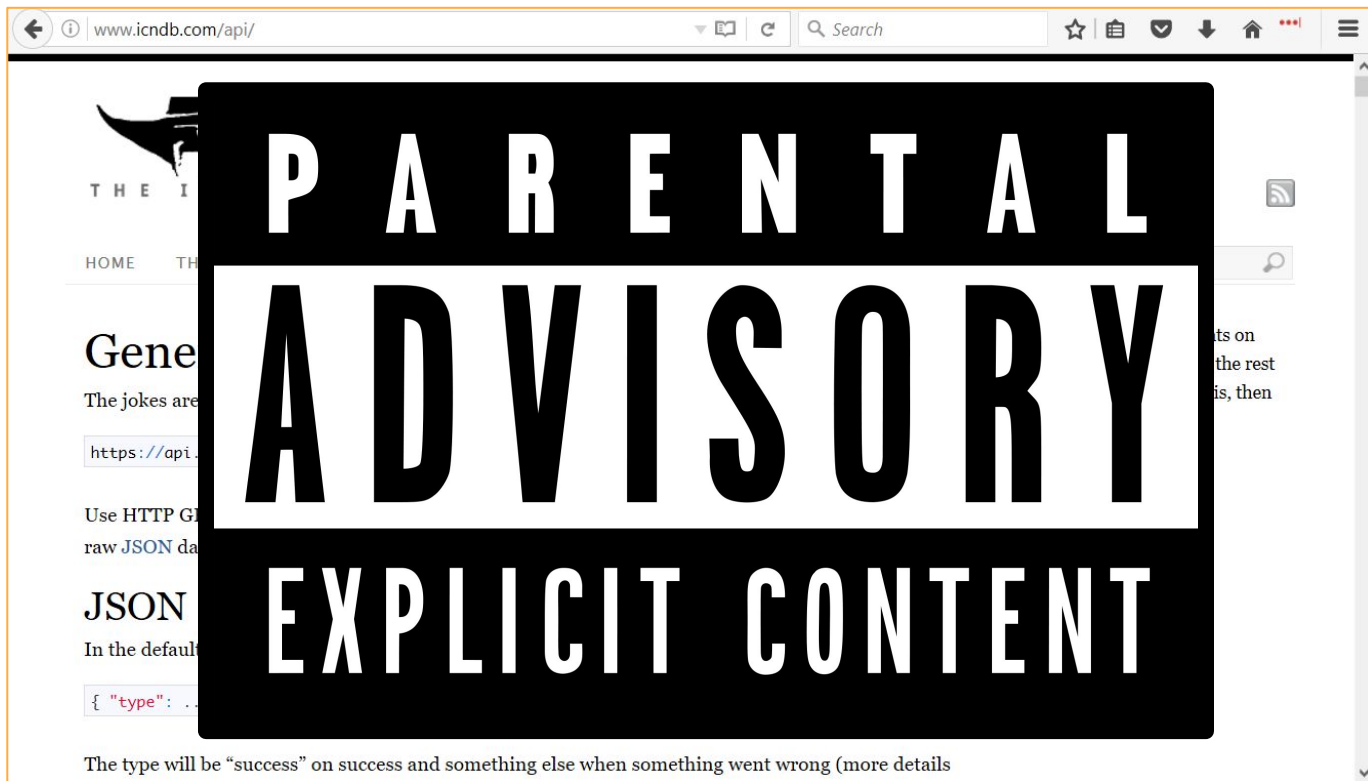
"iTunes pays Chuck Norris \$0.99 every time he listens to a song."



1



# Introduction



The image shows a browser window with the address bar displaying `www.icndb.com/api/`. The page content is partially obscured by a large black box with white text that reads "PARENTAL ADVISORY EXPLICIT CONTENT".

Visible text on the page includes:

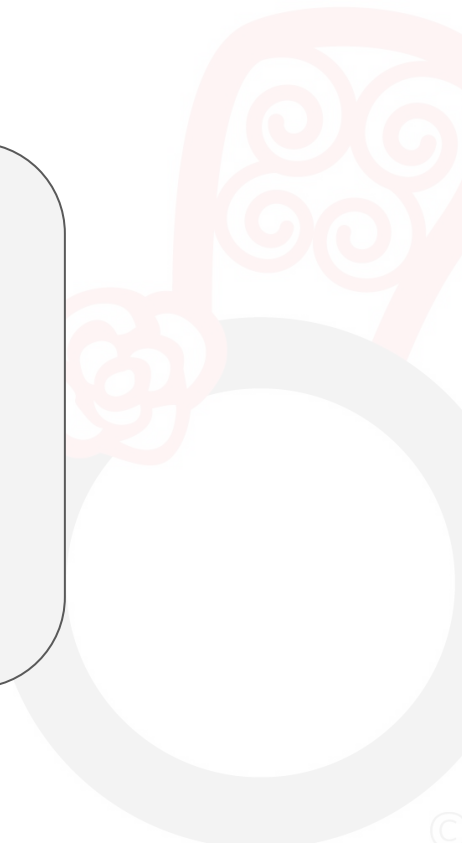
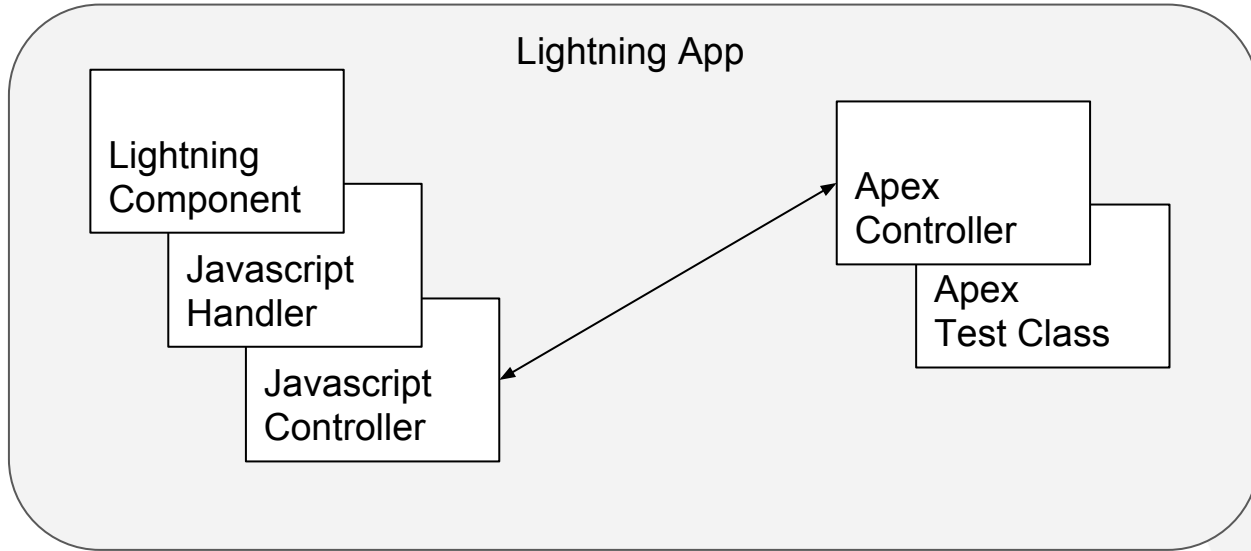
- THE I
- HOME TH
- Gene
- The jokes are
- `https://api.`
- Use HTTP G
- raw JSON da
- JSON
- In the default
- `{ "type": .`
- The type will be "success" on success and something else when something went wrong (more details)

The browser's address bar also shows a search icon and the word "Search". The page has a navigation menu with "HOME" and "TH". There is a search icon on the right side of the page. A large, faint, stylized owl logo is visible in the background on the right side of the image.

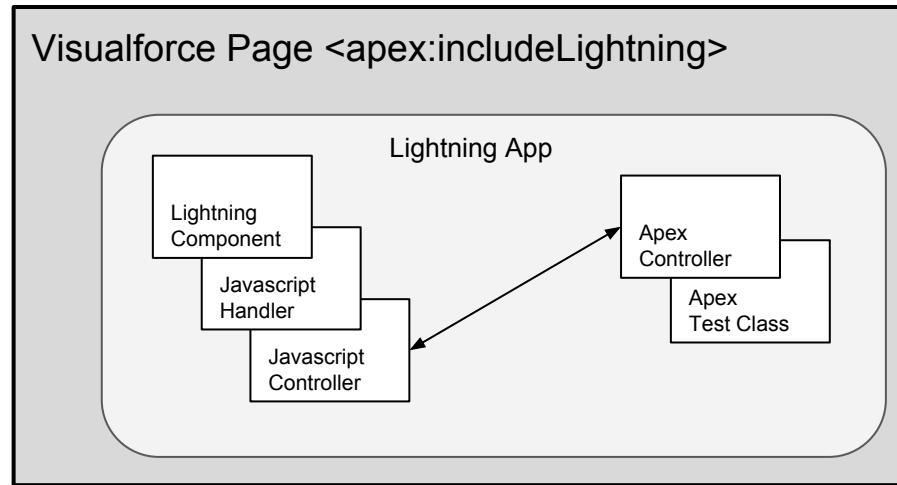




# Lightning Application



# The Visualforce Page



# The Lightning App

The screenshot displays the Salesforce CRM interface for a contact record. The top navigation bar includes tabs for Leads, Accounts, **Contacts**, Opportunities, Forecasts, Contracts, Orders, Cases, Solutions, Products, Reports, and Dashboards. The contact record is for Mr. Jack Rogers, owned by Simon Lawrence. The record details include Name, Account Name (Burlington Textiles Corp of America), Title (VP, Facilities), Department, Birthdate, Reports To (with a View Org Chart link), Lead Source (Web), Mailing Address (525 S. Lexington Ave, Burlington, NC 27215 USA), Languages, Created By (Simon Lawrence, 19/01/2017 10:04), and Description. Action buttons for Edit, Delete, Clone, and Request Update are visible at the top and bottom of the record.

**Visualforce Page <apex:includeLightning>**


The diagram illustrates the Lightning App architecture within a Visualforce page. It shows the following components and their relationships:

- Lightning Component**: The main UI component.
- Javascript Handler**: A component that interacts with the Lightning Component.
- Javascript Controller**: A component that interacts with the Javascript Handler.
- Apex Controller**: A server-side controller that interacts with the Javascript Controller.
- Apex Test Class**: A test class used for testing the Apex Controller.

```
graph TD; LC[Lightning Component] --- JH[Javascript Handler]; JH --- JC[Javascript Controller]; JC --- AC[Apex Controller]; AC --- ATC[Apex Test Class];
```



# The Lightning App



## Mr. Jack Rogers

[Twitter](#) [Facebook](#) [LinkedIn](#) [YouTube](#)

[Customize Page](#) | [Edit Layout](#) | [Printable View](#) | [Help for this Page](#)

[Show Feed](#) | [Click to add topics:](#)

[Opportunities \[0\]](#) | [Cases \[2\]](#) | [Open Activities \[0\]](#) | [Activity History \[0\]](#) | [Campaign History \[0\]](#) | [Notes & Attachments \[0\]](#) | [HTML Email Status \[0\]](#)

### Contact Detail

[Edit](#) [Delete](#) [Clone](#) [Request Update](#)

Contact Owner [Simon Lawrence \[Change\]](#)

No one has ever pair-programmed with Jack Rogers and lived to tell about it.

---

|      |                 |       |                |
|------|-----------------|-------|----------------|
| Name | Mr. Jack Rogers | Phone | (336) 222-7000 |
|------|-----------------|-------|----------------|



# Workbook time!

```
({  
  requestQuote : function(component, event, helper) {  
    helper.retrieveQuote(component,
```

```
<aura:component controller="ChuckController"  
  implements="force:appHostable,flexipage:availableForRecordHome,force:hasRecordId">
```

```
component.get("v.firstN"),  
component.get("v.secondN"));
```

```
<aura:attr  
<aura:attr  
<aura:hand  
<ui:output
```

```
({  
  retrieveQuote : function(component, firstName, secondName) {  
    var initAction = component.get("c.getChuckNorrised");  
    initAction.setParams({ firstN : firstName, secondN: v.secondName });  
    initAction.setCallback(this, function(response) {  
      if (response.getSta  
        component.set(""  
      }  
    });  
    $.enqueueAction(initAction  
  }  
})
```

```
</aura:compone
```

```
<apex:page standardController="Con  
<apex:includeLightning />  
<div id="chuckLine" />  
<script>  
  $Lightning.use("c:ContactCh  
  $Lightning.createComponent  
  ( firstN: "{!Contact.firs  
    secondN: "{!Contact.las  
    "chuckLine",  
    function(cmp) {}  
  );  
  }>  
</script>  
</apex:page>
```

```
@AuraEnabled  
public static String getChucked(String firstN, String secondN) {  
  HttpRequest req = new HttpRequest();  
  req.setEndpoint('http://api.icndb.com/jokes/random?firstName='  
    + firstN  
    + '&lastName='  
    + secondN  
    + '&exclude=explicit');  
  req.setMethod('GET');  
  Http http = new Http();  
  HTTPResponse res = http.send(req);  
  JsonJoke jsjk = (JsonJoke)JSON.deserialize(res.getBody(), JsonJoke.class);  
  return jsjk.value.joke;  
}
```



# Q & A

@srlawr

@edith\_mxuk

