

Lightning Components

Denver Technology Day

Troy Hedges
Principal Solution Engineer
thedges@sforce.com

salesforce

Forward Looking Statements

Safe harbor statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.



Agenda

Lightning Components

- ❑ Lightning Experience Overview
- ❑ Lightning Component Overview
- ❑ Extras
- ❑ Resources
- ❑ Demo



<http://bit.ly/DTDLC>

Lightning Experience Overview

Lightning Experience for End Users

A connected, intuitive user experience on any device

Work Faster

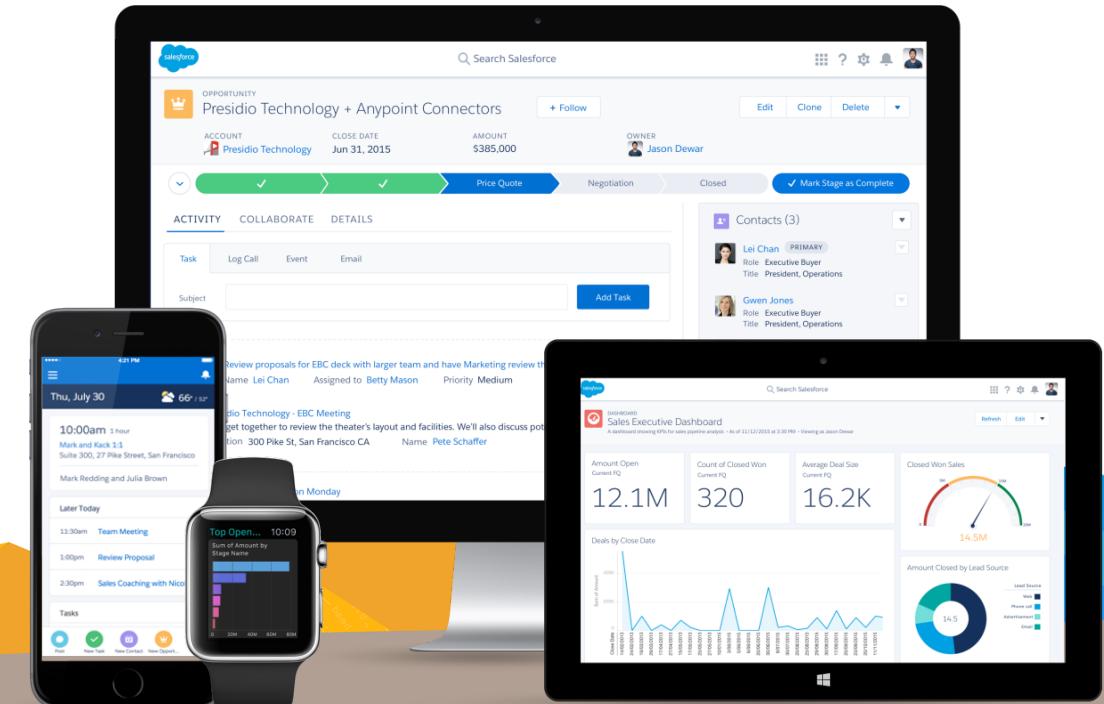
Easy-to-use app experience to maximize productivity

Work Smarter

Analytics and in-line intelligence to drive precision

Work the Way You Want

Get a customized, consistent experience across every device



Empower Everyone to Build Faster with Lightning Builder

Build, customize and deploy apps faster than ever, no code required

Easy to customize

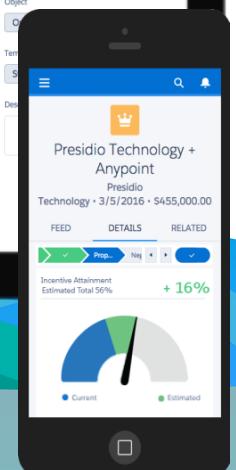
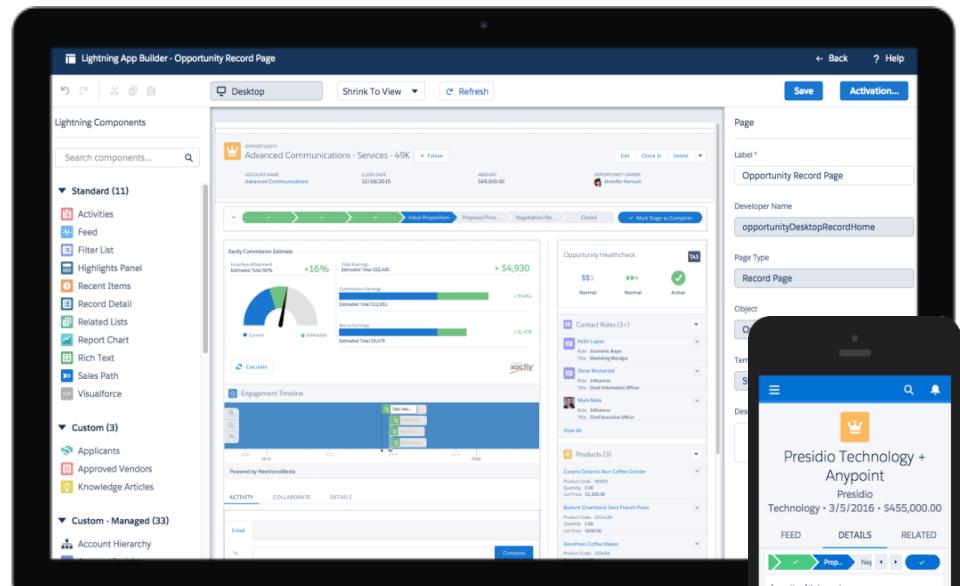
Drag and drop to make changes. Get end user feedback and iterate, instantly.

Build apps faster than ever

Customize pages in the Lightning Experience and build apps without code, for any device.

Everyone is empowered to build apps

Developers code and share components, while admins drag-and-drop components into pages.



Build and Customize Lightning Components

Democratize App Development with UI/UX component framework

Lightning Framework

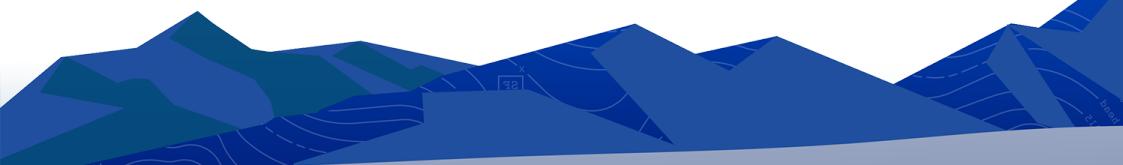
A collection of services you can use to build components and apps, faster

Lightning Components

Use pre-built building blocks from Salesforce, build your own, or connect a 3rd party app from a vibrant ecosystem.

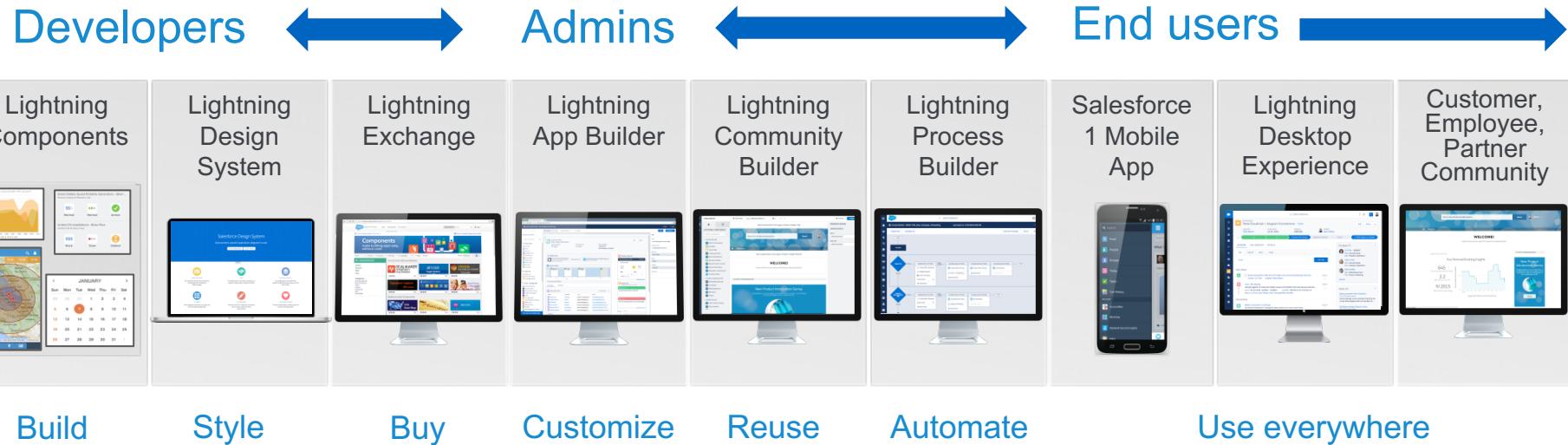
Lightning Design System

Use UI design assets and guidelines for building pixel-perfect apps.



Lightning Delivers a New Continuum of App Development

Ensuring Scalability, Performance, Availability



Lightning Components Overview



What are Lightning Components ?



- Components are self contained, re-usable units of an app
- Build complex apps using multiple simple components - including **Base Lightning Components**
- Inter-component interaction happens through events, attributes and methods
- Client side UI rendering rather than server side

What is Lightning Component Framework?

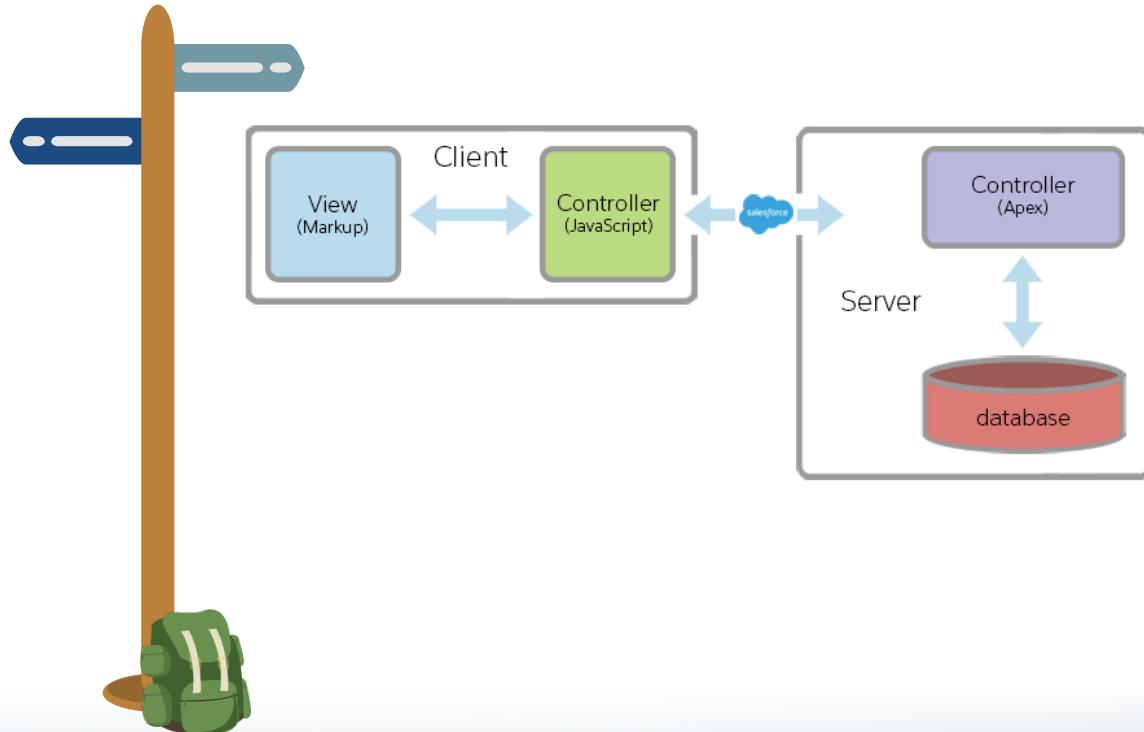
A modern UI Framework for developing dynamic single page web apps engineered for growth

Built on the Open Source
Aura Framework

Framework for both mobile
and desktop devices

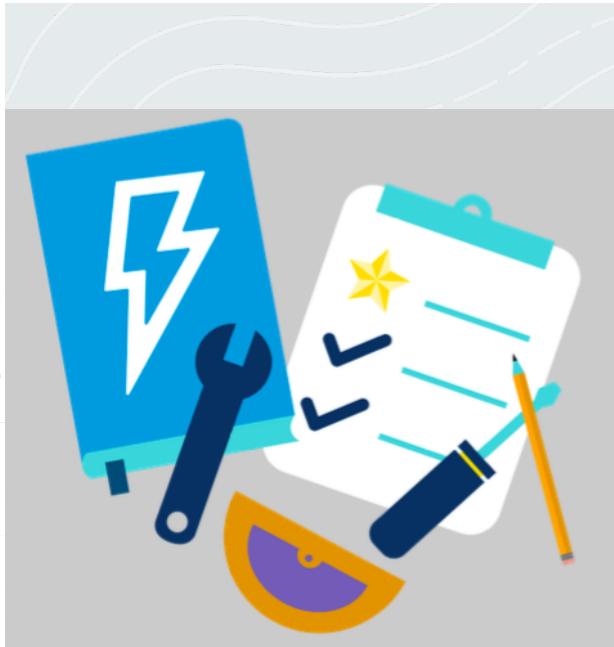
Multi-tier component
development bridges client
and server

Ecosystem supported by
AppExchange



Why Lightning Component Framework?

Features of Lightning that provides competitive advantage



- Out-of-the-box set of components, NATIVE to Salesforce
- Optimized for performance
- Loosely Coupled with publish/subscribe event driven architecture
- Extendable and Reusable across apps and orgs
- Portable via Lightning Out : Build once and use anywhere
- Responsive UI
- Built on metadata from the foundation, accelerating development

Lightning Components come from 3 places

Built by Salesforce



Tasks

- Fill out purchase order Today
- Prepare needs analysis document for Chipotle Tyler Woodhouse

Tasks

Write a comment... Share

Gordon Davis to All Sales Group 30 min ago
Here is the latest demo preso. @Betty Jones, let me know if there are any changes. I've updated slides 3-8 and slides 16-18 with new product shots. I think we made need to revisit the key benefits on slide 2 ... more

Like 2 Likes 0 Comments

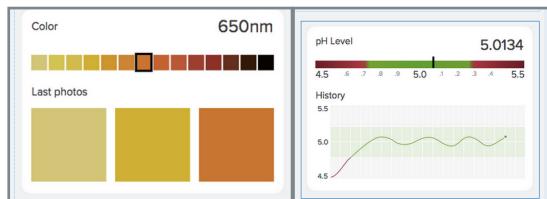
Mary Hiles posted a file 45 min ago

Feed items



Publisher bar

Built by partners,
developers, and
customers



Sliders

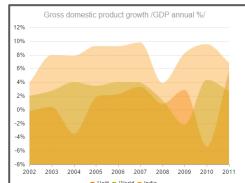


Multi-view charts

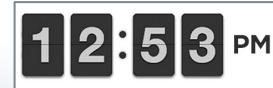
From the AppExchange

Grand Hotels Guest Portable Generators - Brian ...
Grand Hotels & Resorts Ltd

\$\$\$	▶▶	✓
Normal	Normal	Active



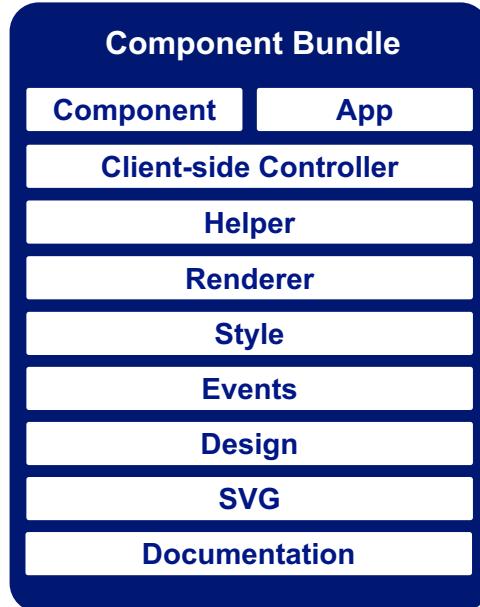
Sign Here
This is a demonstration app for the Signature Capture lightning component.
Please sign below.

A component for capturing a digital signature.

Anatomy of a Lightning Component

Self contained, reusable UI bundles

Client-side
(runs in the client
device web
browser/web-view)



Server-side
(runs in the
salesforce platform)



- Bundle is the development/deployment container
- Component & App contains the markup; one per bundle – either a component or app, not both
- Client-side controller contains the application logic in form of actions, coded in JavaScript; unique per instance of component
- Helper contains reusable JavaScript code that is shared across multiples instance of the same component at runtime
- Renderer contains any custom rendering (DOM manipulation) code for the component; not required in most cases
- Style contains any custom CSS code for the component
- Events (Component/Application) enable intra/inter component communication using publish/subscribe model
- Design contains labels and additional descriptions for attributes exposed in Lightning App Builder
- SVG contains the vector graphics for component icon that is visible in Lightning App builder design panel
- Documentation contains the component documentation
- Apex controller for server side logic; maximum of one per component

Developer Console

The screenshot shows the Force.com Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for Account@12:45 AM, TestLightningComponent, and TestLightningComponent.cmp. The main area displays the following code:

```
1 <aura:component >
2
3 </aura:component>
```

To the right of the code editor is a sidebar menu with a red border around its top section:

- Ctrl + Shift + 1 COMPONENT Create
- Ctrl + Shift + 2 CONTROLLER Create
- Ctrl + Shift + 3 HELPER Create
- Ctrl + Shift + 4 STYLE Create
- Ctrl + Shift + 5 DOCUMENTATION Create
- Ctrl + Shift + 6 RENDERER Create
- Ctrl + Shift + 7 DESIGN Create
- Ctrl + Shift + 8 SVG Create

Below this list is a blue button labeled "Bundle Version Settings".

At the bottom of the interface, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected. A table below the tabs shows columns for Name, Line, and Problem.



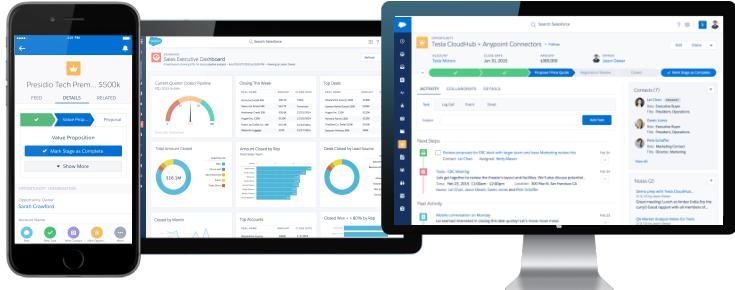
Lightning and Visualforce are fundamentally different

L I G H T N I G	visualforce™
CLIENT-SIDE UI GENERATION	SERVER-SIDE UI GENERATION
APP-CENTRIC MODEL	PAGE-CENTRIC MODEL
COMPONENT BASED FRAMEWORK	MVC FRAMEWORK
JAVASCRIPT FRAMEWORK	TAG-BASED LANGUAGE
DESIGNED FOR LIGHTNING EXPERIENCE AND SALESFORCE1	DESIGNED FOR SALESFORCE CLASSIC

Credit: Mike Topolovich, [Patterns for Migrating from Visualforce to Lightning Components: Part 2](#)

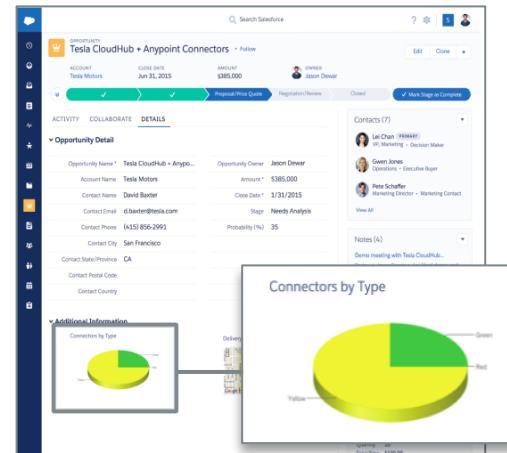
Visualforce will be supported for the long term

Lightning is the future of
Salesforce technology



Both our mobile and new desktop are built with
Lightning Components

Visualforce and your
customizations will continue to be
supported in Lightning Experience*



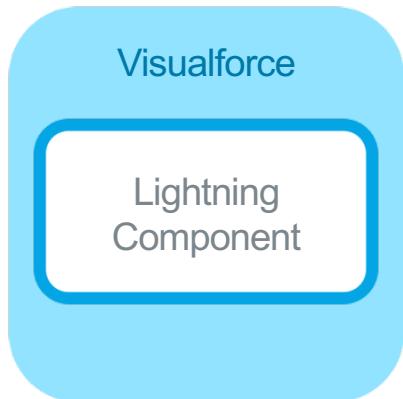
*see exceptions list

Visualforce and Lightning work together

Visualforce will always be supported. Start planning for Lightning.

1.

Use Lightning Components
in Visualforce

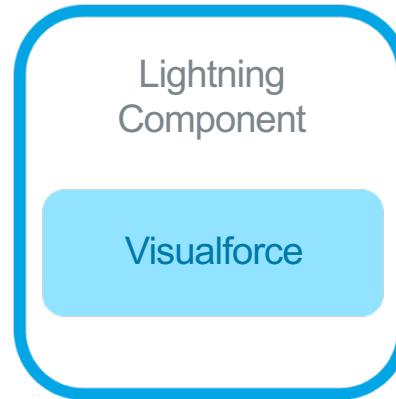


"Lightning for Visualforce"

Winter '16

2.

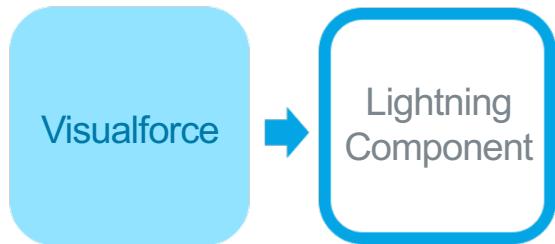
Use Visualforce in a
Lightning Component



Visualforce as an iframe

3.

Migrate Visualforce to
Components



Lightning Components will have the
same functionality as Visualforce

Coming soon

Lightning Components Usage

Lightning Pages

- Add components to Lightning App pages.
- Includes record detail pages and a home page.
- Email Application Panes (beta)

Salesforce1

Add components to Lightning Component tab.

Standalone Apps

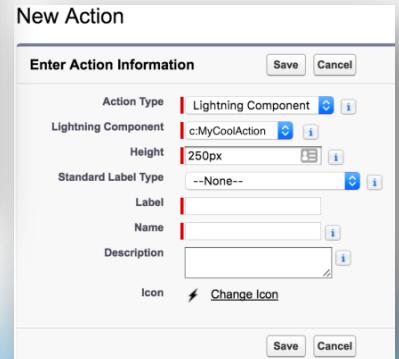
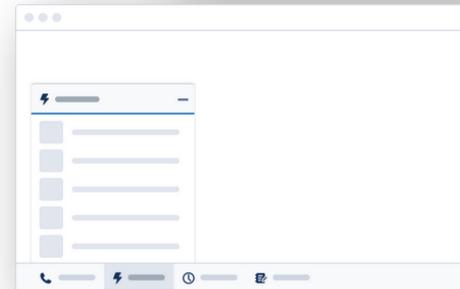
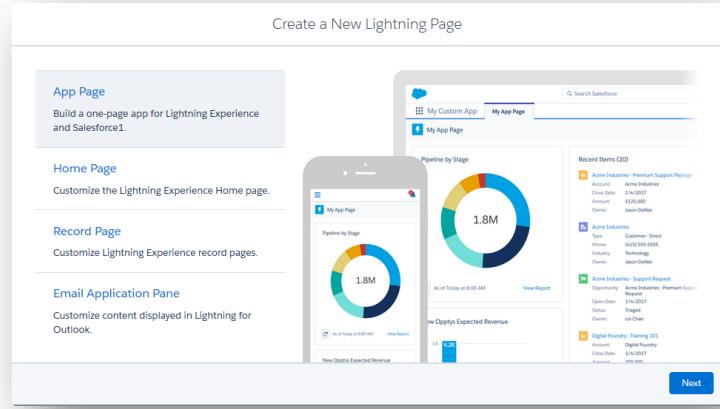
Add components to full-screen apps.

Lightning App Utility Bar

Add components to the Utility bar of Lightning Apps.

Lightning Actions

Add components as actions to Lightning pages.



Empower Everyone with the Lightning App Builder

Now Anyone Can Build Apps for Every Device

Drag & Drop

Build with standard and custom Lightning Components



The image illustrates the Lightning App Builder's drag-and-drop functionality and its cross-device compatibility. On the left, a grid of icons represents different components available for building. In the center, a desktop monitor and a smartphone both display the same complex app interface, demonstrating how users can build once and deploy to multiple devices.

Any Device

Design apps for every screen from one canvas



salesforce | platform





Lightning Components

Lightning App Builder - My Lightning App

Phone Shrink To View Refresh Save Activation...

Lightning Components

Search components...

Standard (5)

- Filter List
- Recent Items
- Report Chart
- Rich Text
- Visualforce

Custom (4) (highlighted)

- AnythingPath
- Cell
- HBCContactList
- LightningAccountMap

Custom - Managed (7)

- Account Hierarchy
- Account Plan Summary
- Opportunity Healthcheck List

Get more on the AppExchange

Phone Actions My Rich Text

All Contacts More

All Contacts
1 item, sorted by Phone

Trevor Scott
Account Na...
Title:
Department:
Email: tscott@salesforce.c...

Page

* Label: My Lightning App

* Developer Name: My_Lightning_App

Page Type: App Page

Template: One Column

Description:

Actions

New Account, Log a Call, New Case

Select...





Key Capabilities: LEX Record Page

Lightning App Builder - Time-Off Request Record Page

Desktop Shrink To View Refresh Save Activation...

Lightning Components

Search components...

Filter List
Highlights Panel
Recent Items
Record Detail
Related Lists
Report Chart
Rich Text
Visualforce
Wave Dashboard

Custom (4)

AnythingPath
Knowledge Quick Search
RSS Feed
Twitter Timeline

TIME-OFF REQUEST TO-080124-0

Check In Edit Delete

Approved > Processed > Rejected > Canceled > Requires Re-Approval

RELATED DETAILS

REQUEST INFO

Request ID: TO-080124-0
Requestor: Shane AnyPath
Created By: Shane AnyPath, 12/27/2015 1:49 PM
Policy Exception
Employee Number: 100566

Status: Approved
Date Submitted: 1/23/2008 4:31 PM
Date Approved: 1/23/2008 4:32 PM
Last Modified By: Shane AnyPath, 12/28/2015 2:44 PM

REQUEST DETAILS

Start Date: 1/25/2008
Number of Days: 5

Add Component(s) Here

Page > AnythingPath

API name for picklist that will become the path

Status__c

click to change

Component Configuration with help bubbles from developer

Lightning Supports Visualforce

The screenshot shows the Lightning App Builder interface. On the left, the 'Lightning Components' sidebar lists various components under 'Standard (13)' and 'Custom (0)'. The 'Visualforce' component is highlighted with a red box and has a red arrow pointing from it to the main canvas area. The main canvas displays an 'Account' record page for 'Beef Corporation of America'. A red box highlights the 'My Visualforce Component Section' section on the page. To the right, a configuration panel titled 'Page > Visualforce' shows settings for the component: 'Label' is set to 'My Visualforce Component Section', 'Visualforce Page Name*' is set to 'My Visualforce Page', and 'Height (in pixels)' is set to '300'. The bottom of the screen shows the Twitter integration and the 'Get more on the AppExchange' button.

When you create a custom app page in Lightning App Builder, you can add a Visualforce page to the page by using the Visualforce component.





Key Capabilities: Salesforce1 Tabs

Lightning App Builder - testpage

← Back ? Help

Activation for 'testpage'

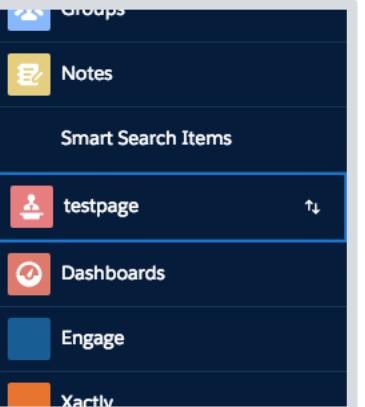
To make this page available in Salesforce1, choose a tab icon, update the tab label, set the page visibility, and set the page's position in the navigation menu.

Tab Label *
testpage

Tab Icon
 Change...

Tab Visibility
 Show for ALL Users
 System Administrators Only

Salesforce1 Navigation Position



The screenshot shows the Salesforce1 navigation menu with the following items:

- Groups
- Notes
- Smart Search Items
- testpage
- Dashboards
- Engage
- Xactly

Cancel **Activate**

Page > Filter List

Object *

Account

Filter *

My Accounts

Number of Records to Display

3

Lightning Components

Search components... 

▼ Standard (5)

-  Filter List
-  Recent Items
-  Report Chart
-  Rich Text
-  Visualforce

▼ Custom (4)

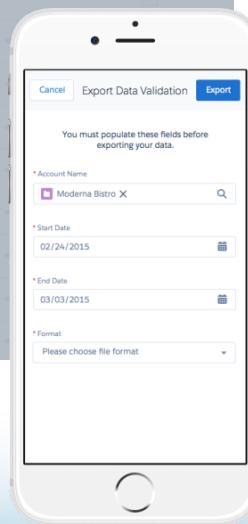
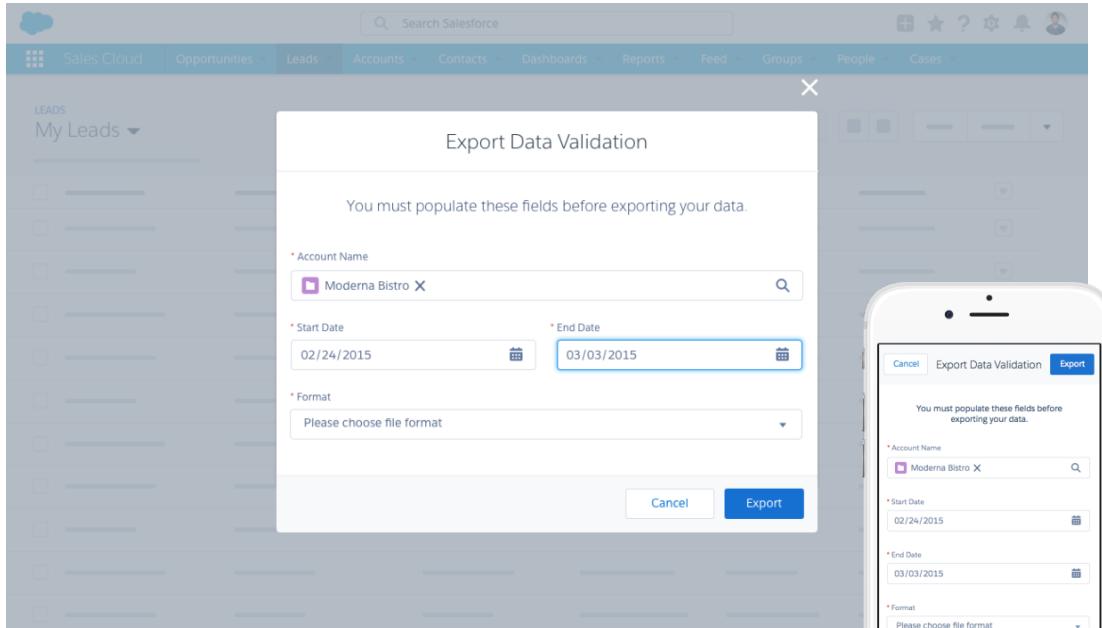
-  AnythingPath
-  Knowledge Quick Search
-  RSS Feed
-  Twitter Timeline

▼ Custom - Managed (0)

No components available.

Get more on the AppExchange

Lightning Actions



Invoke a custom Lightning Component in Salesforce1 and Lightning Desktop



Custom Global Actions (Spring '17)

Custom button, custom link, mass action support (Future)





Key Capabilities: Community Builder Extensibility

Community Templates use a similar UI and can use the same components

Community Builder Candidate

+ New Page Page: Home ▾

Help Romesh Fernan...

PAGE EDITOR

Add components to this region (Custom Header Top)

AnythingPath

Record Id 5001500000PwOYn

Object Name Case

API name for picklist that will become the path

Status

click to change

PAGE STRUCTURE

Home

Custom Header Top

Header

- Search Publisher
- Profile Header
- Navigation Menu

Custom Header Bottom

Featured

- Headline
- AnythingPath
- Record Headline

Sidebar Featured

Content

- Home Tabs

Sidebar Content

PROPERTY EDITOR

AnythingPath

Record Id 5001500000PwOYn

Object Name Case

API name for picklist that will become the path

Status

click to change

TOPICS MY CASES

WELCOME, CANDIDATE!

Your background check is Progress is 20% complete

Larry Baxter: Education: Oxford University

Medium • Waiting on School • 00001650

Waitin... Waiting... Escala... Warni... Violati... Respon... Close... Closed... Closed...

This screenshot shows the Oracle Community Builder page editor interface. The top navigation bar includes 'Community Builder', 'Candidate', 'Help', and a user profile for 'Romesh Fernan...'. The left sidebar contains a 'PAGE STRUCTURE' tree with sections like 'Home', 'Custom Header Top', 'Header' (with options for 'Search Publisher', 'Profile Header', and 'Navigation Menu'), 'Custom Header Bottom', 'Featured' (with options for 'Headline' and 'AnythingPath'), and 'Sidebar' (with 'Featured', 'Content' (including 'Home Tabs'), and 'Sidebar Content'). The main content area displays a candidate profile for 'Larry Baxter' from 'Education: Oxford University' with status 'Medium' and case ID '00001650'. It features a 'WELCOME, CANDIDATE!' message, a progress bar indicating a 20% complete background check, and a timeline at the bottom showing stages like 'Waiting...', 'Escala...', 'Warni...', 'Violati...', 'Respon...', 'Close...', and 'Closed...'. The right sidebar is titled 'PROPERTY EDITOR' and lists fields for 'Record Id' (5001500000PwOYn), 'Object Name' (Case), 'API name for picklist that will become the path', 'Status', and a checkbox labeled 'click to change'. The overall interface is designed for building custom community pages using standard UI components.

Extras

What is the Lightning Data Service? (GA Summer '17)

A centralized data and caching service for Lightning

Benefits:

Developer delight

No APEX. No SOQL. Auto handling of FLS and sharing. Declarative CRUD operations.

Performance

Cache hits and shared data requests improve performance.

Single data source

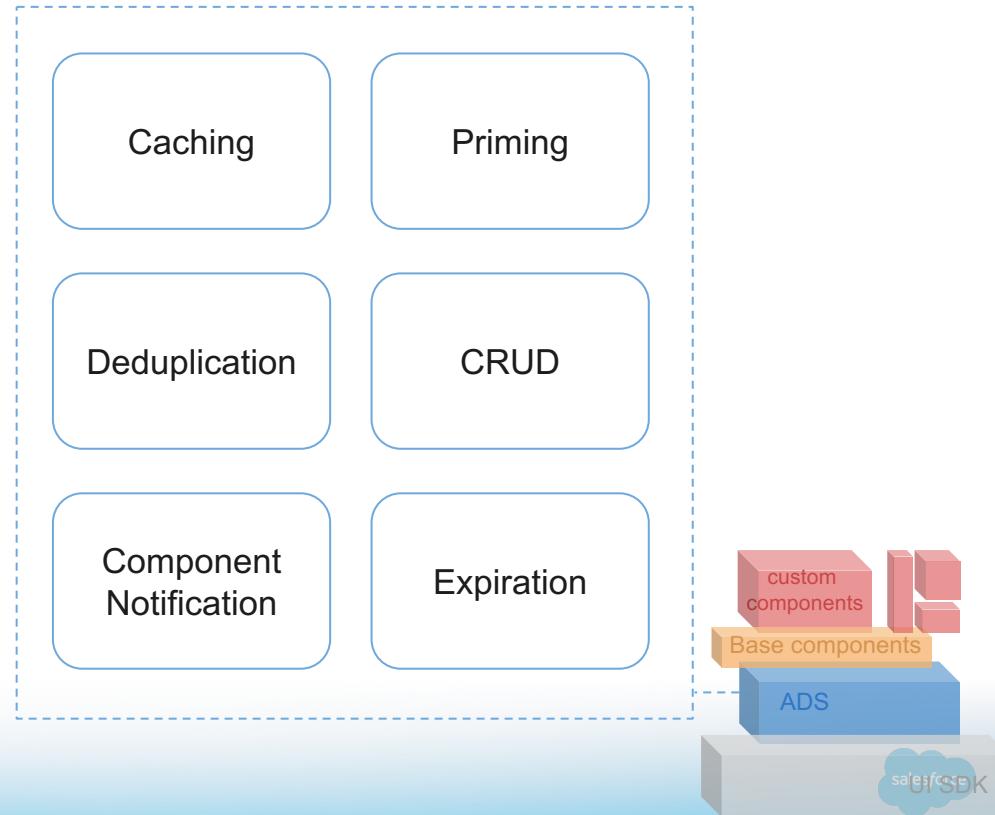
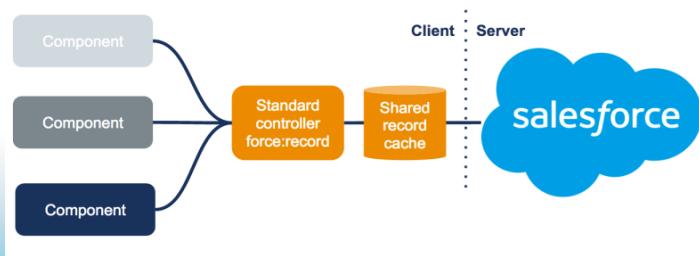
Different components using the same records always display the same data.

Notifications when data changes

recordUpdated event fired on force:record when data changes.

Offline read and write

Local drafts of saves and deletes persist on device until connection is restored.



Locker Service In Lightning Components

- LockerService is a powerful **security architecture** for Lightning components. LockerService enhances security by **isolating individual Lightning components in their own containers**. LockerService also promotes best practices that improve the supportability of your code by only allowing access to supported APIs and eliminating access to non-published framework internals.
- LockerService Requirements:
 - JavaScript ES5 Strict Mode Enforcement
 - DOM Access Containment
 - Restrictions to Global References
 - Access to Supported JavaScript API Framework Methods Only
 - Stricter Content Security Policy (CSP)

Lightning LockerService



Rollout:



Winter '17: Critical Update for ALL Orgs



Spring '17: Critical Update for All Orgs + CSP Checks in Sandbox and DE Orgs

Summer '17: Auto-Enabled in ALL Orgs

* For components of API version 40 and above

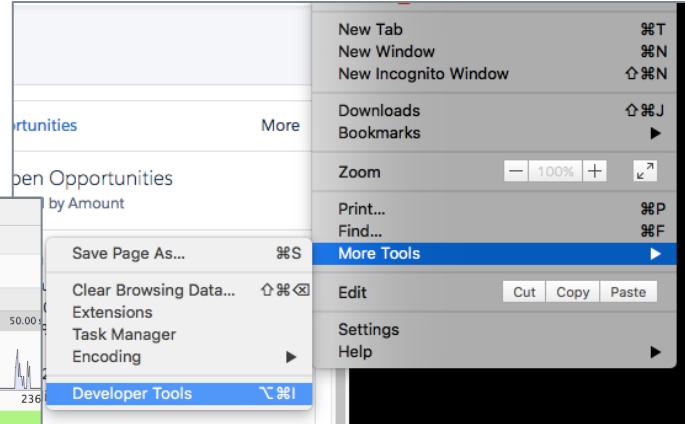
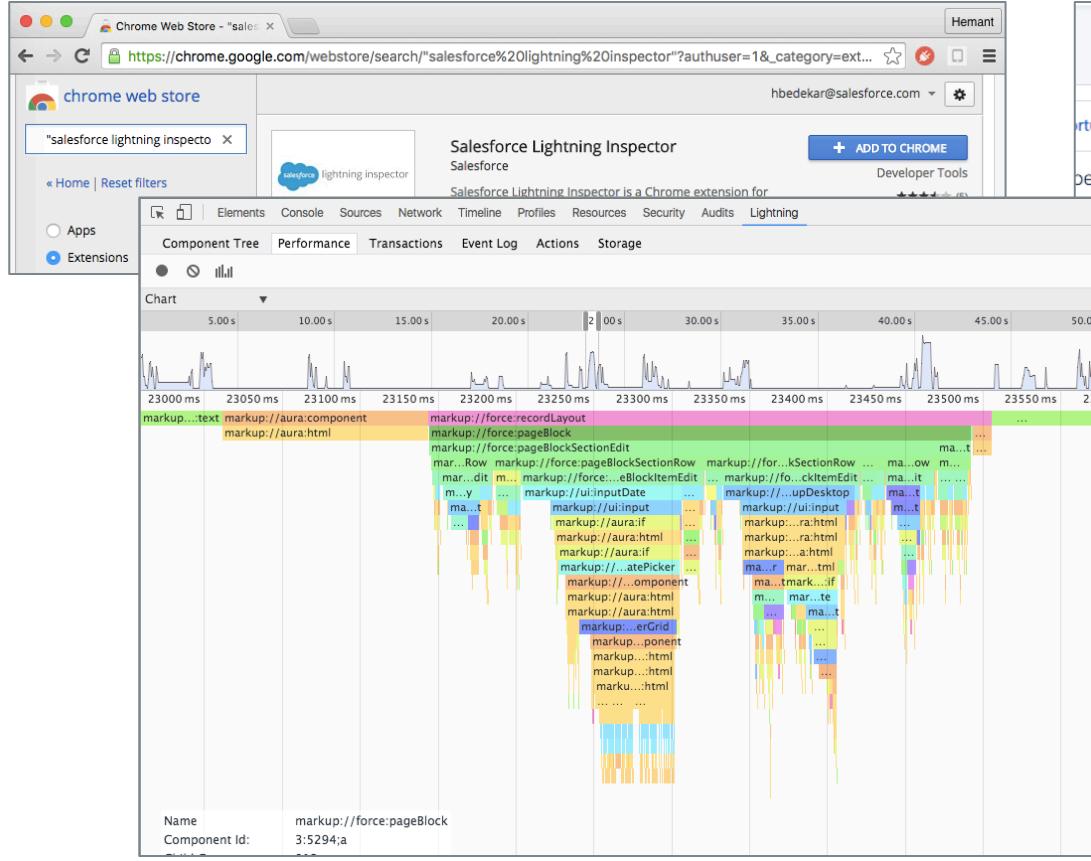


How to Debug Lightning Components

- Use JS debugger, console.log statements
- Use Lightning inspector plugin in chrome
- Use Lightning CLI to debug code (LINT)



Lightning Inspector



Heroku Toolbelt: Lightning Lint

The screenshot shows two windows side-by-side. On the left is the 'Salesforce Summer '16 Release Notes' page, specifically the 'Lightning Components' section. It includes a brief introduction to the Lightning CLI, a note about LockerService enablement, and links to the Lightning Components Developer Guide and ESLint Project. On the right is a terminal window titled 'Search Summer '16' with the command 'heroku lightning:lint' run. The output shows five ESLint errors related to ECMAScript syntax and intrinsic API usage.

```
theedges-ltm2:PSObjectSearch theedges$ heroku lightning:lint .

FILE: /Users/theedges/Google Drive/TroyAntUtils/Pkg-PSObjectSearch/aura/PSObjectSearch/PSObjectSearchHelper.js:

error  ecma-intrinsics  Invalid Intrinsic API
Line:134:24
    sql = "SELECT " + Array.from(fieldSet).join(', ') + "\n";
           ^
           ^

error  no-redeclare  'i' is already defined
Line:143:16
    for (var i = 0; i < filterFieldComps.length; i++) {
           ^
           ^

error  eqeqeq  Expected '!==' and instead saw '!='
Line:145:77
    if (filterFieldComps[i].value !== null && filterFieldComps[i].value != "") {
           ^
           ^

error  eqeqeq  Expected '===' and instead saw '=='
Line:149:48
    } else if (filterFieldComps[i].ftype == 'string') {
           ^
           ^

error  no-redeclare  'cls' is already defined
Line:202:11
    var cls = mapLatField + " != null AND " + mapLngField + " != null";
```

* 5 problems (5 errors, 0 warnings)

Resources

Create Modern App UI/UX with Lightning Design System

Developers can focus automating business processes rather than creating pixels



Consistent User Experience

Salesforce style guides and modern enterprise design patterns give a unified experience across every app

Deliver Apps Fast

Open source framework and reusable code base to build apps in record time

Seamless upgrades across releases

Easy migration path with Design Tokens versioned for each release

<https://www.lightningdesignsystem.com/>



Trailhead: Lightning



7 hrs 15 mins
7 Modules

Get Started with Lightning Experience

Start your journey to becoming a new Salesforce Admin with Lightning Experience.

Beginner | Administrator



1 hr 55 mins
2 Modules

Sell Lightning Fast with Salesforce

Learn how to use the new Salesforce to sell lightning fast in as few clicks as possible.

Beginner | Business User



3 hrs 45 mins
4 Modules

Migrate to Lightning Experience

Increase the productivity of your sales team by migrating your existing org to the new Lightning Experience.

Intermediate | Administrator



10 hrs 35 mins
5 Modules

Develop for Lightning Experience

Build apps fast for Lightning Experience with Visualforce, components, and new design resources.

Intermediate | Developer



Unlocked

Lightning Experience Specialist

Power up your sales reps with a super-charged interface and process automation.



Learn How to Build Lightning Components

Search (359 results for "lightning component")

Build an Account Geolocation App
Intermediate | Developer
Project

Create the AccountList Component
Nest Lightning components Use **component** attributes Use... > New > **Lightning Component**.
Lightning Component: Specify AccountList as the bundle

Create the AccountLocator Component
Create a Lightning component Use a Lightning component Create a Lightning Application Use a Lightning Component in a Lightning Application Use a Lightning Component

Create the AccountMap Component
> New > **Lightning Component**. Specify AccountMap... Create the AccountMap Component What You'll Do Create the AccountMap component responsible for displaying

Build a Restaurant-Locator Lightning Component
Advanced | Developer
Project

Make the Component Available in Salesforce1
a context-aware, Yelp-style Lightning Component! ... Make the Component Available in Salesforce1
What You'll Do Create a Lightning App for Salesforce1 Activate

Create Initial Component Layout
Create a Lightning Component Add the component... Lightning Design System Although we're adding our component... page Create a Lightning Component Click the Setup

Making the Component Context Aware
the InTheArea component from the Lightning Components... Make the Component Context Aware
What You'll Do Modify the component to work with Account and Contact

Quick Start: Lightning Components
Beginner | Developer
Project

Create a Lightning Component
Create a Lightning Component A lightning component... create a component bundle. In the Developer Console, select File | New | Lightning Component

Enable Lightning Components
Introduction The Lightning Component framework... a simple Lightning Component that renders a list... controller class, then create a Lightning Component

Lightning Components Developer Guide

Version 40.0, Summer '17

@salesforcedocs
Last updated: May 25, 2017

<https://trailhead.salesforce.com/>



Lightning Component Development Guide

The screenshot shows the "Lightning Components Developer Guide" page. At the top, there's a navigation bar with links for "PRODUCTS", "RESOURCES", "COMMUNITY", "BLOG", and "TRAILHEAD". On the right side of the header are "Search", "Login", and "Sign Up" buttons. Below the header, the page title is "Lightning Components Developer Guide". On the left, there's a sidebar with dropdown menus for "v40.0", "EN", and "PDF", and a search bar. The main content area has a section titled "Component Reference" which says: "Use out-of-the-box components for Lightning Experience, Salesforce1, or for your Lightning apps. These components belong to different namespaces, including:" followed by a list of namespaces with their descriptions. At the bottom of the page are "Previous" and "Next" navigation buttons.

v40.0 EN PDF

Search in document

Contents

- What is the Lightning Component Framework?
- Quick Start
- Creating Components
- Using Components
- Communicating with Events
- Creating Apps
- Debugging
- Fixing Performance Warnings
- Reference
 - Reference Doc App
 - Supported aura:attribute Types
 - aura:application
 - aura:component
 - aura:dependency
 - aura:event
 - aura:interface
 - aura:method
 - aura:set
- Component Reference
 - aura:expression
 - aura:html
 - aura:if
 - aura:iteration

Component Reference

Use out-of-the-box components for Lightning Experience, Salesforce1, or for your Lightning apps. These components belong to different namespaces, including:

- aura
 - Provides components that are part of the framework's building blocks.
- force
 - Provides components for field- and record-specific implementations.
- forceChatter
 - Provides components for the Chatter feed.
- forceCommunity
 - Provides components for Communities.
- lightning
 - Provides components with Lightning Design System styling. For components in this namespace that are used in standalone Lightning apps, extend `force:slds` to implement Lightning Design System styling. In instances where there are matching `ui` and `lightning` namespace components, we recommend that you use the `lightning` namespace component. The `lightning` namespace components are optimized for common use cases. Event handling for `lightning` namespace components follows standard HTML practices and are simpler than that for the `ui` namespace components. For more information, see [Event Handling in Base Lightning Components](#).
- ui
 - Provides an older implementation of user interface components that don't match the look and feel of Lightning Experience and Salesforce1. Components in this namespace support multiple styling mechanism, and are usually more complex.

◀ Previous Next ▶

https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/aura_compref.htm



Introducing AppExchange for Components

The Enterprise Marketplace for Lightning Components

Build apps easily

With pre-built Components from Salesforce partners that work for any device

Build More Engaging Apps

Use the best pre-built components in your apps

Deliver apps faster

Use Components from AppExchange with your own custom-built Components

The screenshot shows the Salesforce AppExchange Components page. At the top, there's a banner with the text "Build mobile apps at lightning speed." and several app icons. Below the banner are filters for "Free", "All Editions", "All Ratings", and "All Languages", along with "Apply" and "Clear" buttons. A "Sort By" dropdown is set to "Popularity". On the left, a sidebar has sections for "Components" (selected), "Home", "Popular Components", "New Components", and "Free". It also lists "Categories" like Data Management, Data Visualization, Feeds, Finance, Productivity, Sales, and Time & Date. A "Most Popular" section displays 12 components in a grid, each with a thumbnail, name, rating, and download count. Examples include "Lightning DataTable Dev" (4.5 stars, 15 reviews), "Lightning Carousel and Banner" (4.5 stars, 4 reviews), and "My Tasks LTC" (4.5 stars, 12 reviews). Some components have "FREE" status, while others like "Hierarchy" and "Enhanced Lightning Grid" are marked as "PAID". A "View More" link is at the bottom right of the grid.

<https://appexchange.salesforce.com/components/free>



Strike Components by Appiphony



Spinner
Tabset & Tab (Beta)
Textarea

LIGHTNING STRIKE COMPONENTS

Strike Badge
Strike Chart
Strike Data Grid
Strike Datepicker
Strike Input
Strike Lookup
Strike Modal
Strike Multi Lookup
Strike Multi Select Picklist
Strike Pill
Strike Popover
Strike Select
Strike Textarea
Strike Tile
Strike Tooltip

Strike Lookup

A form element for searching and selecting an existing record

LIVE EXAMPLE

Account

a

COPY

LIVE CODE

```
<c:strike_lookup
    label="Account"
    object="Account"
    searchField="Name"
    placeholder="Select an option"
    iconName="standard:account"
    subtitleField="Industry"
    order="Name"
    limit="5"
    loadingMessage="Loading..."
    errorMessage="Invalid input"/>
```

Attributes

* label [?](#)
Account

* object [?](#)
Account

* searchField [?](#)
Name

helpText [?](#)

placeholder [?](#)
Select an option

iconName [?](#)
standard:account

subtitleField [?](#)
Industry

filter [?](#)

order [?](#)
Name

<http://www.lightningstrike.io/>



Lightning Component Reference

The screenshot shows the 'Reference' tab selected in the top right corner of the Aura interface. The left sidebar contains a search bar and a navigation menu with sections like 'Overview', 'Applications', 'Components' (expanded to show sub-components: aura, c, force, forceChatter, forceCommunity, lightning, and various UI components), 'Interfaces', 'Events', 'Libraries', and 'JavaScript API'. The main content area is titled 'Reference Overview' and contains the following text:

The Reference tab enables you to browse API information as well as source code for all the definitions, such as components, in Aura.

APIs

Click **JavaScript API** to see the publicly accessible methods for each JavaScript object. The \$A namespace is the entry point for using the framework in JavaScript code.

Definitions

You can browse the source for all the applications, components, interfaces, events, and tests in Aura by expanding the sidebar folders. The definitions are grouped in sub-folders for each namespace. For example, click **Components > Aura > Component** to find information about `<aura:component>`, which is the root of the component hierarchy.

There are multiple tabs available for each of the definitions. For components, the **overview** tab lists the component's attributes. For `<aura:component>`, you can see that there's only one attribute called `body`. Click the **source** tab to see the attribute defined in a `<aura:attribute>` tag. The source shows all the markup, including the system attributes, such as `abstract`, which are properties of the top-level `<aura:component>` tag.

The other tabs vary depending on the type of definition. For components, you might see tabs for the client-side renderer and client-side controller if they're included in the component's bundle.

Support Level for Definitions

Each definition has a support level indicating whether it is fully supported or not. The support level is defined in the `support` system attribute in the root tag of the definition. The support level is shown in the top-right corner for each definition in the Reference tab.

We will explain the support levels in terms of components, which are the most common definitions to have different support levels. The support levels in ascending order are:

1. **PROTO** — The component is still in development and is unsupported. This is the default support level.
2. **DEPRECATED** — The component is deprecated and is no longer supported.
3. **BETA** — The component is ready for beta use but we don't recommend using it in a GA app.
4. **GA** — The component is generally available for use in apps and is supported.

If you extend a component, you can't give the child component a higher support level than the parent component. For example, if cmpA is **BETA** and cmpB extends cmpA, cmpB can't have a support level of **GA**. It can have a support level of **BETA** or **PROTO**.

<https://<org-url>/auradocs#reference>

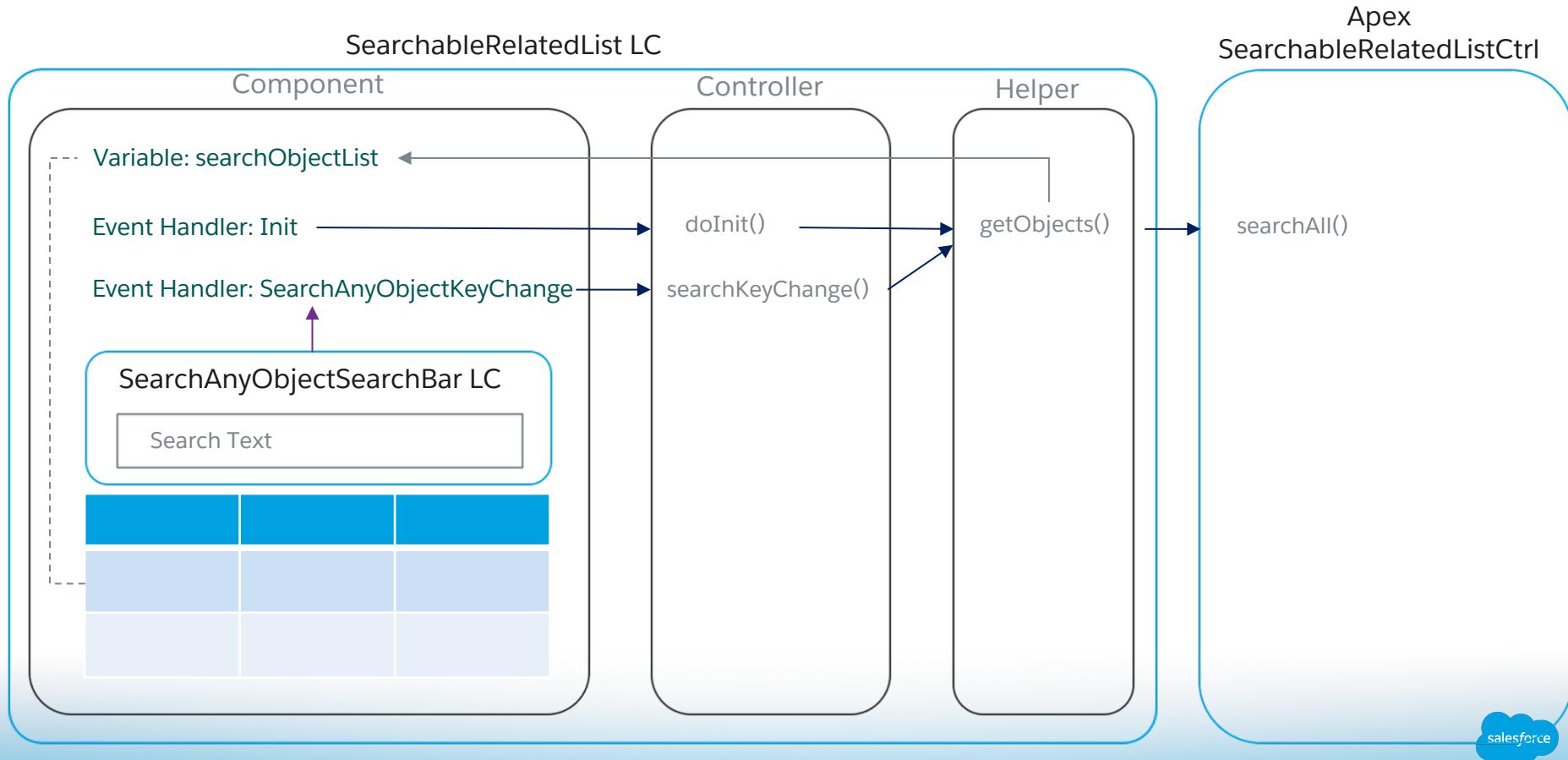


GitHub

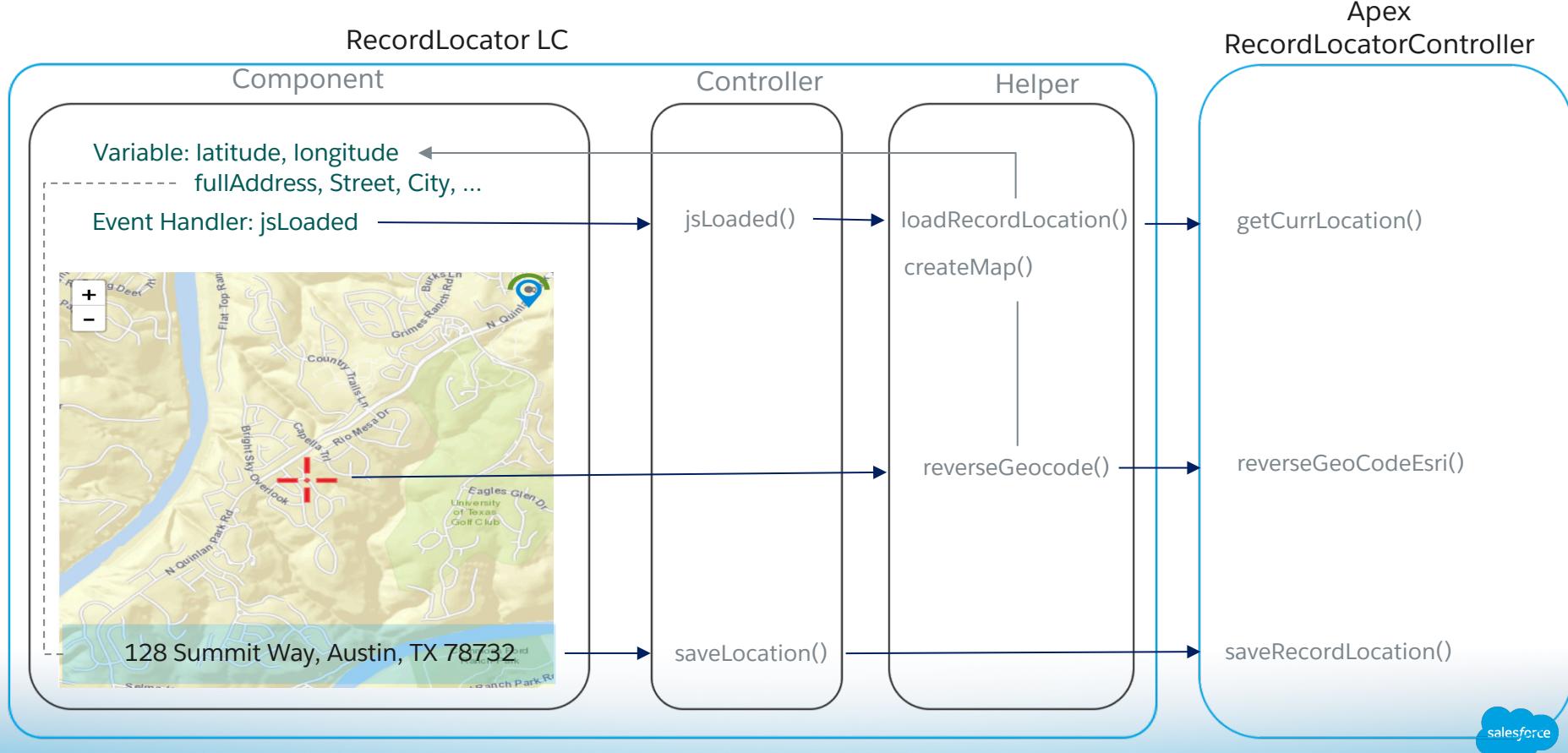
- Christophe Coenraets - <https://github.com/ccoenaerts>
 - DreamHouse Sample App - <https://developer.salesforce.com/dreamhouse/>
- Shane McLaughlin - <https://github.com/mshanemc/>
- Troy Hedges - <https://github.com/thedges>

Demo

Searchable Related List



Record Locator



The Salesforce logo, which consists of the word "salesforce" in a white, lowercase, sans-serif font, centered within a white, rounded, three-dimensional cloud shape.

salesforce

Thank you