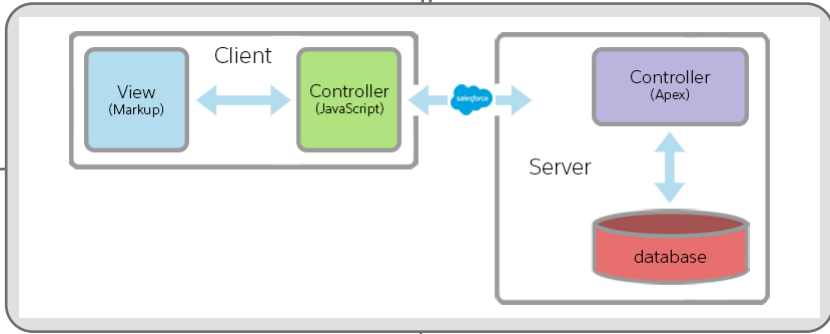


Laymen`s guide
Lightning
Component
Development.

Lightning component
framework is a UI
framework used for
developing web
apps.built on open
source Aura framework.

It uses JavaScript on
the client side and
Apex on the server
side.



Lightning Component
Development basics.
We will cover more with help
of example code below this
mindmap.

Follows Object Oriented
Programming Principles.

Object-oriented
programming consists of
Objects, Methods and
Properties.

OBJECT = black box
which stores some
information. eg.- Door

Each object has certain
properties (information
about the object). eg.
Door`s doorKnob

Methods/Functions
= Most objects have a
certain collection of
things that they
can DO. eg.- Door can
OPEN

Events are how we
trigger our functions to
run. eg. on event of
door being KNOCKED
we trigger the function
to OPEN the DOOR

JavaScript Basics

Runs on browser (open
Javascript console ctrl
+shift+J and write 1+1
and hit enter)

CaseSensitive (unlike
visualforce)

JavaScript v/s Java -
Nothing in common
except Name.

Lightning applications
are build by assembling
lightning components
together.

Lightning component
Bundle consists of
followings

Controller(sampleController.js)
Client-side controller methods to handle events in
the component.
eg. perform logic to openDoor when doorKnocked
event is fired.

Component(sample.cmp)
1. Contains Markup (HTML, other components) - What you see.
2. Also contains attribute (properties) declarations - Data you capture/
use/manipulate.
3. Event handlers - eg. perform action openDoor when doorKnocked
event is fired.
4. Event Registers - eg. which says this component can fire event
doorKnocked.

Helper(sampleHelper.js)
All the reusable javascript code goes here,
can be called from any js code in the cmp bundle
and also can be called via other components code
via inheritance.

Design(sample.design)
Used when you wish to pass in some attributes via app builder/community builder.

CSS Styles(sample.css)
All the Custom Styling
required by the
component to change
it`s look and feel goes
here.

Renderer(sampleRenderer.js)
Client-side renderer to
override default rendering
for a component.

SVG (sample.svg)
Custom icon for the
component which
appears in the
Lightning App Builder.

Find the Documentation of all the
custom and standard components in
the org at
(instance).salesforce.com/auradocs

One component can communicate to another
component,
1. via events (component/application)
2. via pass by reference(2 way binding by enclosing
them in same shell component)