Angular 2 Fundamentals

GETTING STARTED

Jim Cooper
SOFTWARE CRAFTSMAN

@jimthecoop

Required Prerequisites

Basic JavaScript

app.pluralsight.com/paths/skills/javascript

Basic HTML

app.pluralsight.com/paths/skills/html5

Helpful Prerequisites

Basic Node and Npm

app.pluralsight.com/courses/npm-playbook

Modules and Module Loaders

app.pluralsight.com/courses/javascript-module-fundamentals

ES2015

app.pluralsight.com/courses/javascript-fundamentals-es6

TypeScript

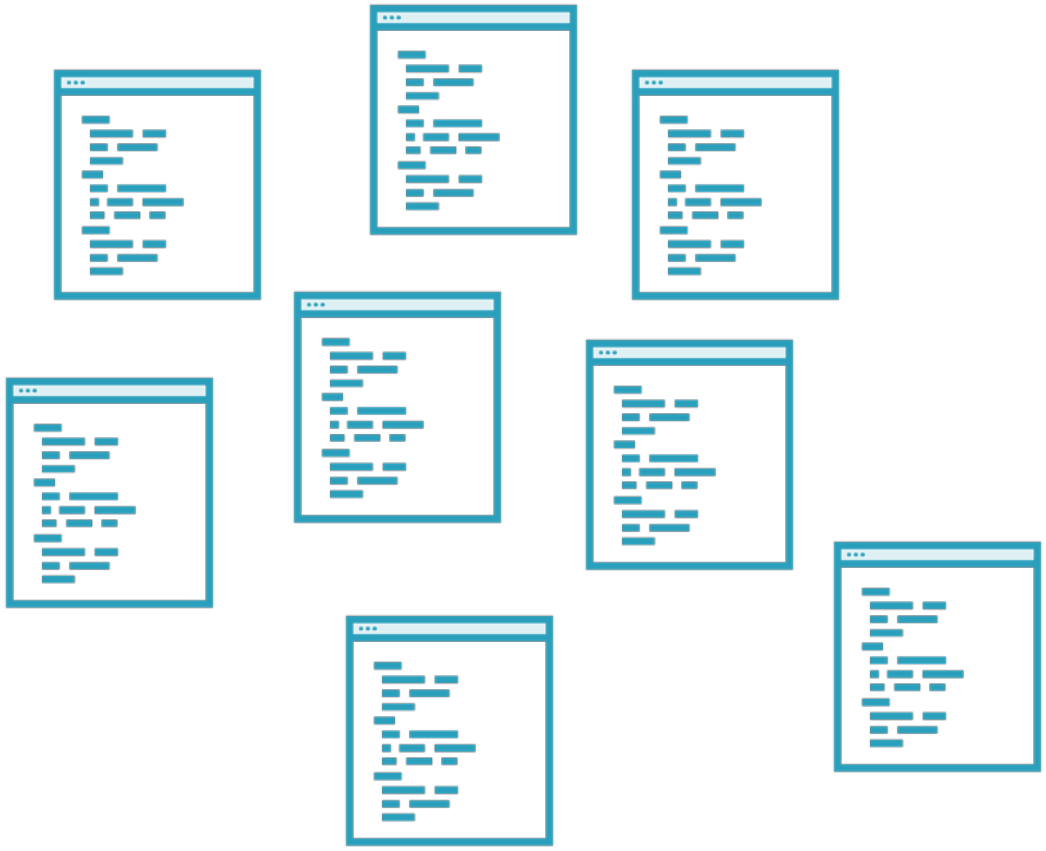
app.pluralsight.com/courses/typescript

Angular

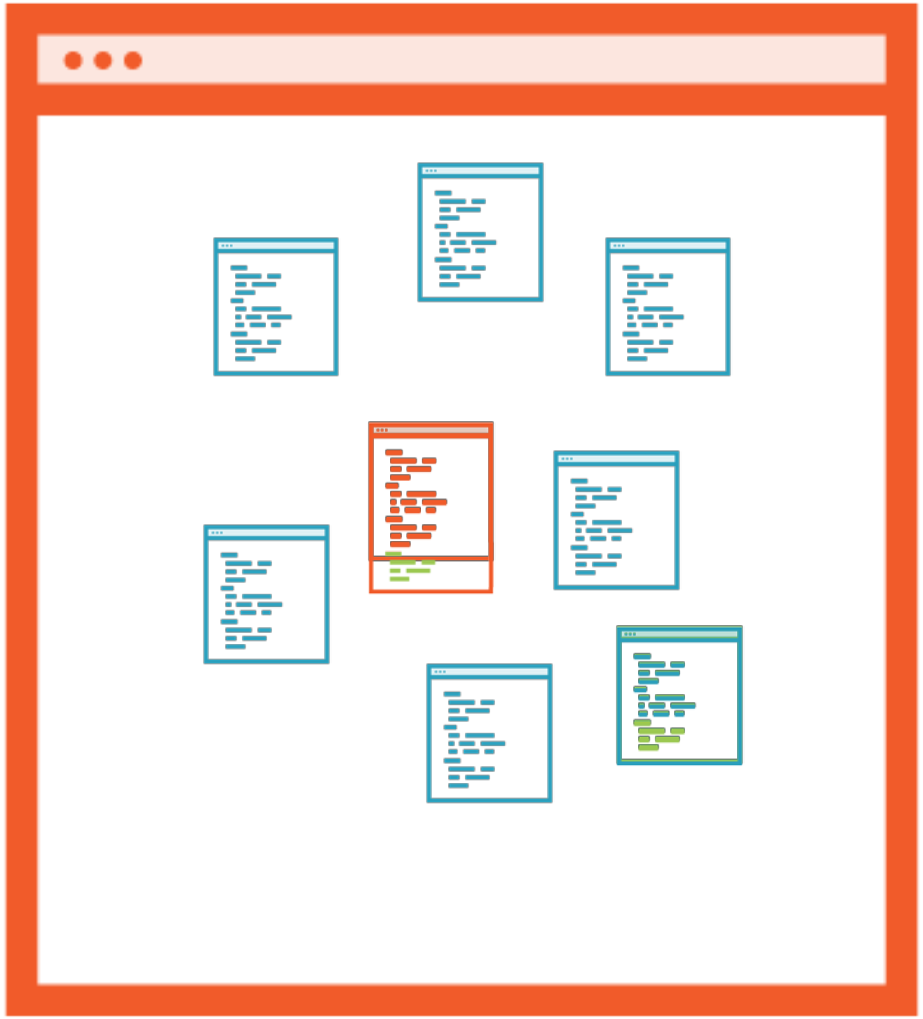
app.pluralsight.com/courses/angular-2-getting-started-update

Why JavaScript Modules are Important

Why Modules are Important



Why Modules are Important

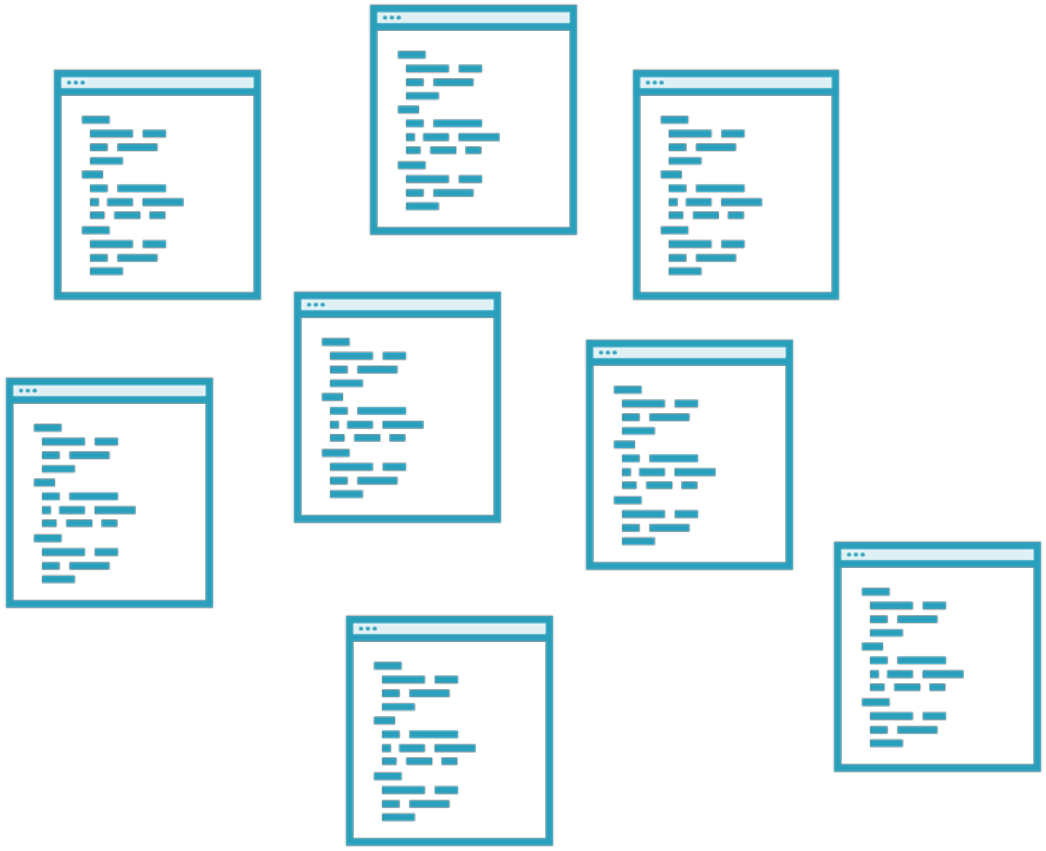


```
import { foo } from
'../folder/some-file.js'

export {
  foo: someFunction()
}
```

What is SystemJs?

index.html

```
<script src="file1.js">…
<script src="file2.js">…
<script src="file3.js">…
<script src="file4.js">…
<script src="file5.js">…
<script src="file6.js">…
.
.
.
```



What is SystemJs?

index.html

```
<script src="system.js">
<script src="config.js">
```

system.config.js

```
var config = {
 map: {
  'app': '/folder/app'
 },
 packages: {
  'app': {main: 'main.js'}
 }
}
```

TypeScript Features

Static Typing

Interfaces

Class Properties

Public/Private Accessibility

```
let name : string
let age : number
let birthDate : date
```

Static Typing

```
interface ICat {
 name:string
 age:number
}
```

TypeScript Interfaces

```
interface ICat {
 name:string
 age:number
}

let fluffy:ICat
```

```
fluffy =
  {
    name: 'Fluffy',
    age: 'seven'
  }
```

TypeScript Interfaces

```
interface ICat {
 name:string
 age:number
}

let fluffy:ICat
```

```
fluffy =
  {
    name: 'Fluffy'
  }
```

TypeScript Interfaces

```
interface ICat {
 name:string
 age?:number
}

let fluffy:ICat
```

```
fluffy =
  {
    name: 'Fluffy'
  }
```

TypeScript Interfaces

```
class Cat {
  constructor (name) {
    this.name = name
  }
}
```

TypeScript Class Properties

```
class Cat {
  name:string
  constructor (name) {
    this.name = name;
  }
}
```

TypeScript Class Properties

```
class Cat {
  name:string
  color:string
  constructor (name) {
    this.name = name;
  }
}
```

TypeScript Class Properties

```
class Cat {
  name
  color
  constructor (name) {
    this.name = name;
  }
}
```

TypeScript Class Properties

```
class Cat {
  name:string
  color:string
  constructor (name) {
    this.name = name;
  }
  speak() { console.log('meow') }
}
```

Public and Private Accessibility

```
class Cat {
  name:string
  color:string
  constructor (name) {
    this.name = name;
  }
  speak() { console.log('My name is: ' + this.name) }
}
```

Public and Private Accessibility

```
class Cat {
  name:string
  speak() { console.log('My name is: ' + this.name) }
}

let fluffy = new Cat()
console.log(fluffy.name)
fluffy.speak()
```

Public and Private Accessibility

```
class Cat {
  private name:string
  private speak() { console.log('My name is: ' + this.name) }
}

let fluffy = new Cat()
console.log(fluffy.name)
fluffy.speak()
```

Public and Private Accessibility

```
class Cat {
  private name:string
  private speak() { console.log('My name is: ' + this.name) }
}

let fluffy = new Cat()
console.log(fluffy.name) //compile-time error
fluffy.speak() // compile-time error
```

Public and Private Accessibility

```
class Cat {
  private name:string
  private color:string
  constructor(name, color) {
    this.name = name
    this.color = color
  }
}
```

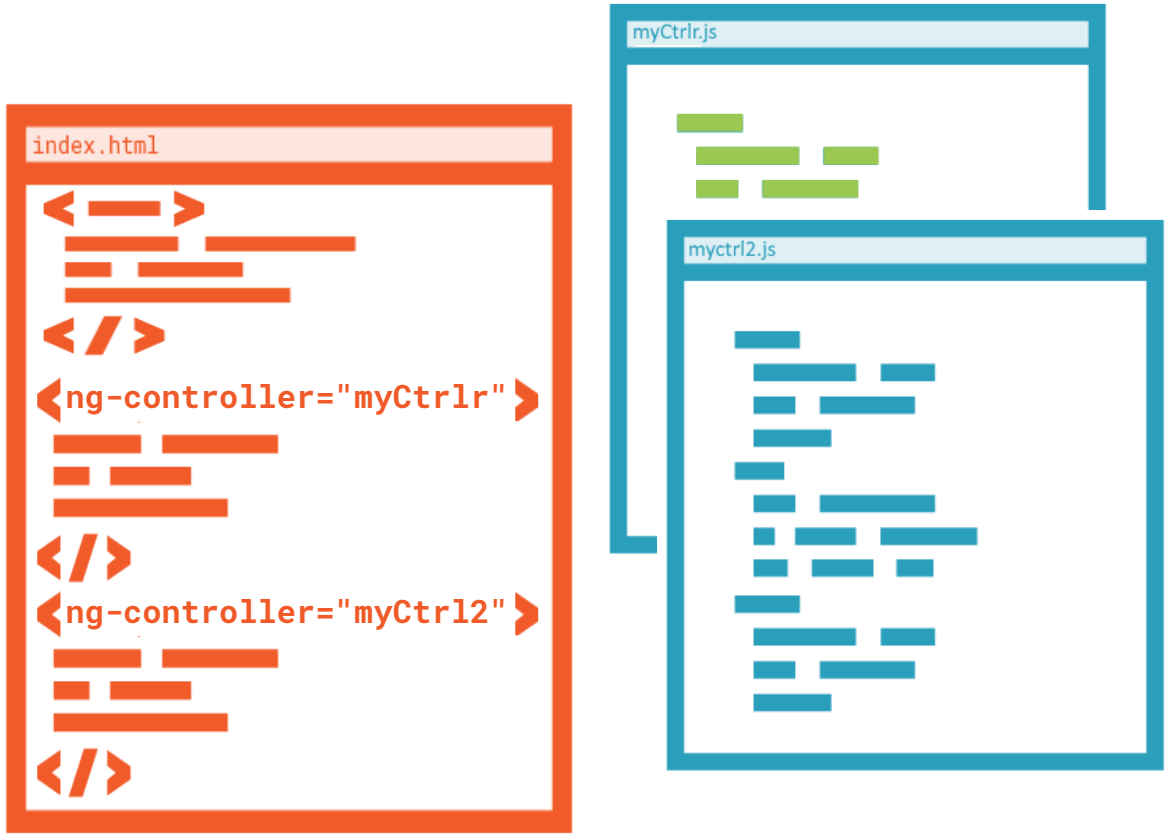
```
class Cat {
  constructor(private name, private color) {

  }
}

let fluffy = new Cat('Fluffy', 'White')
```

Public and Private Accessibility

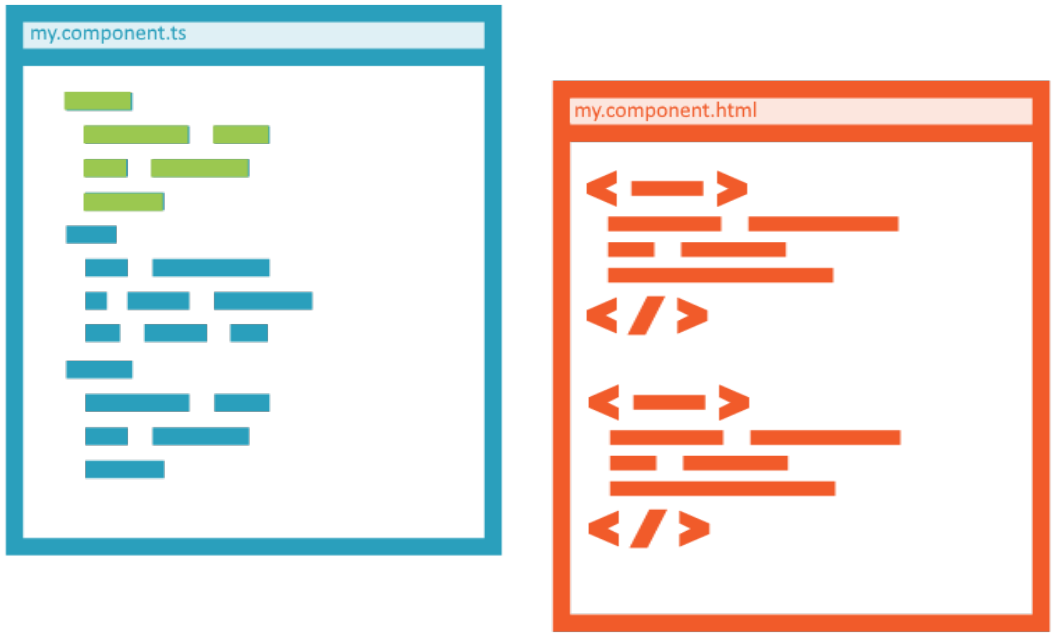
Angular 2 Conceptual Overview

MVC vs Components

Angular 1

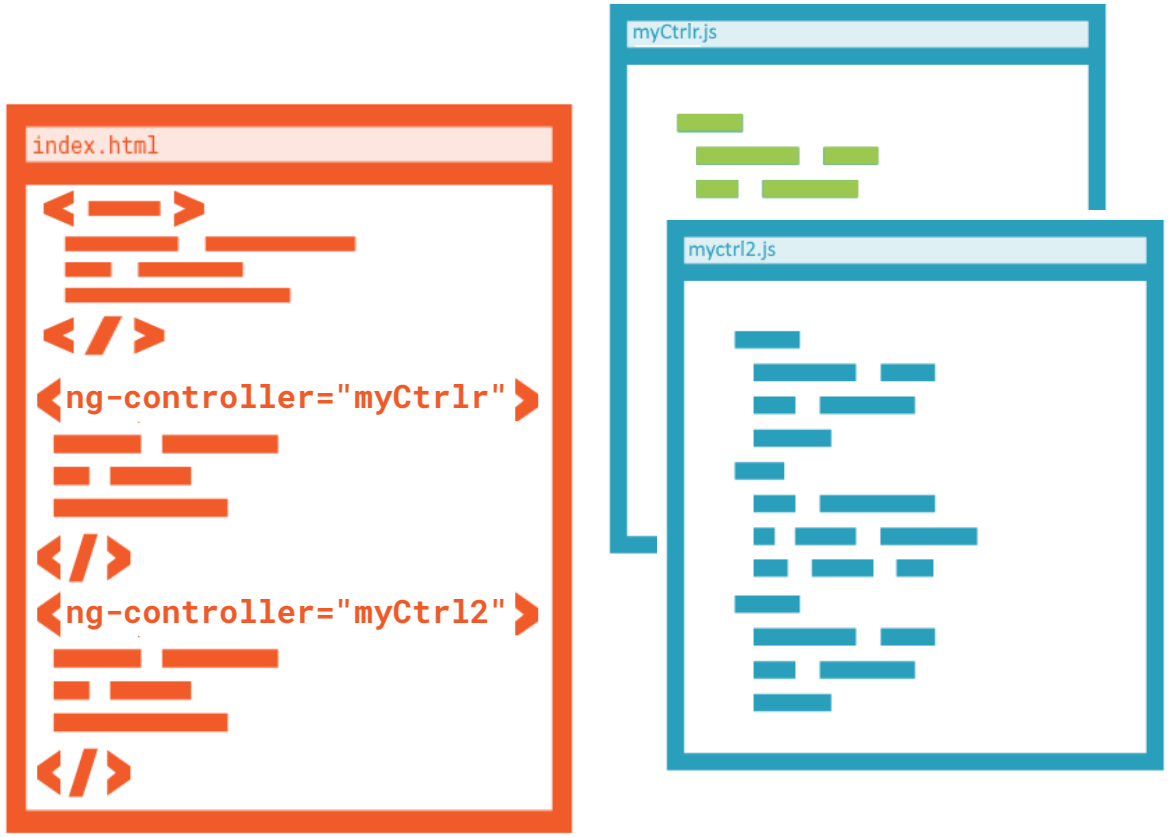


Angular 2



MVC vs Components

Angular 1



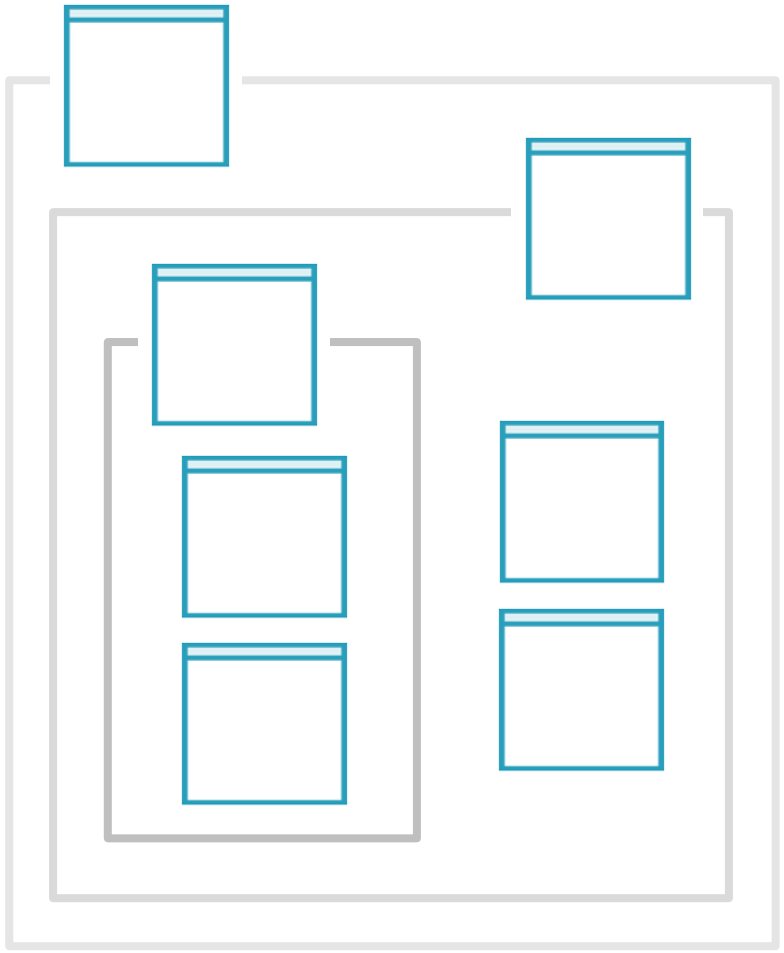
Angular 2

MVC vs Components

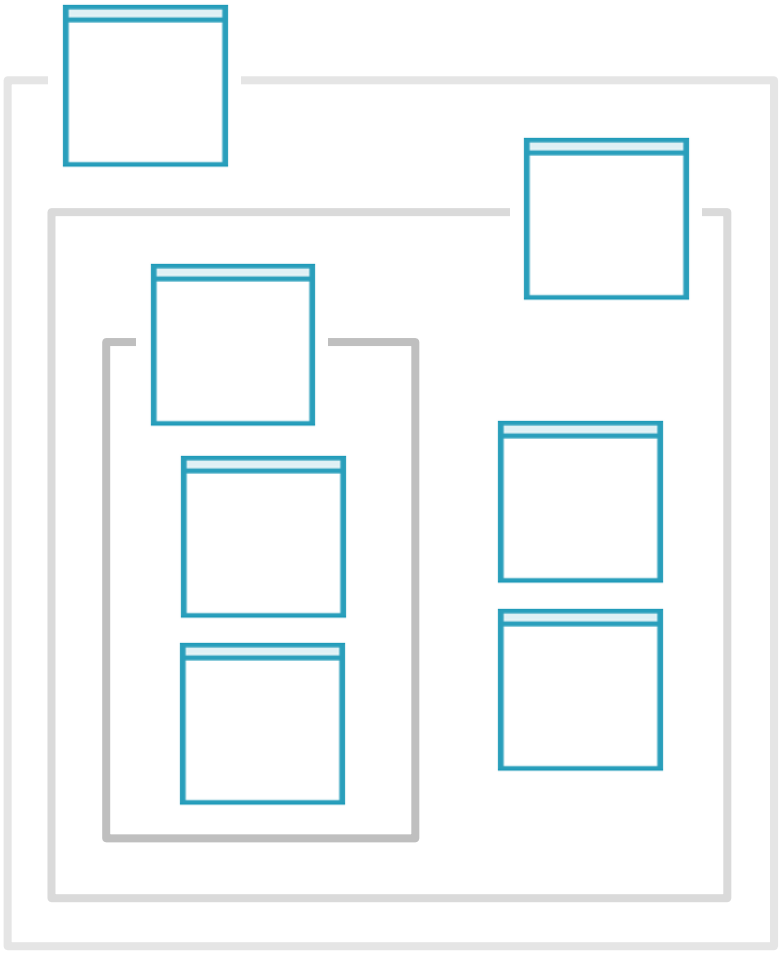
Angular 1

Angular 2

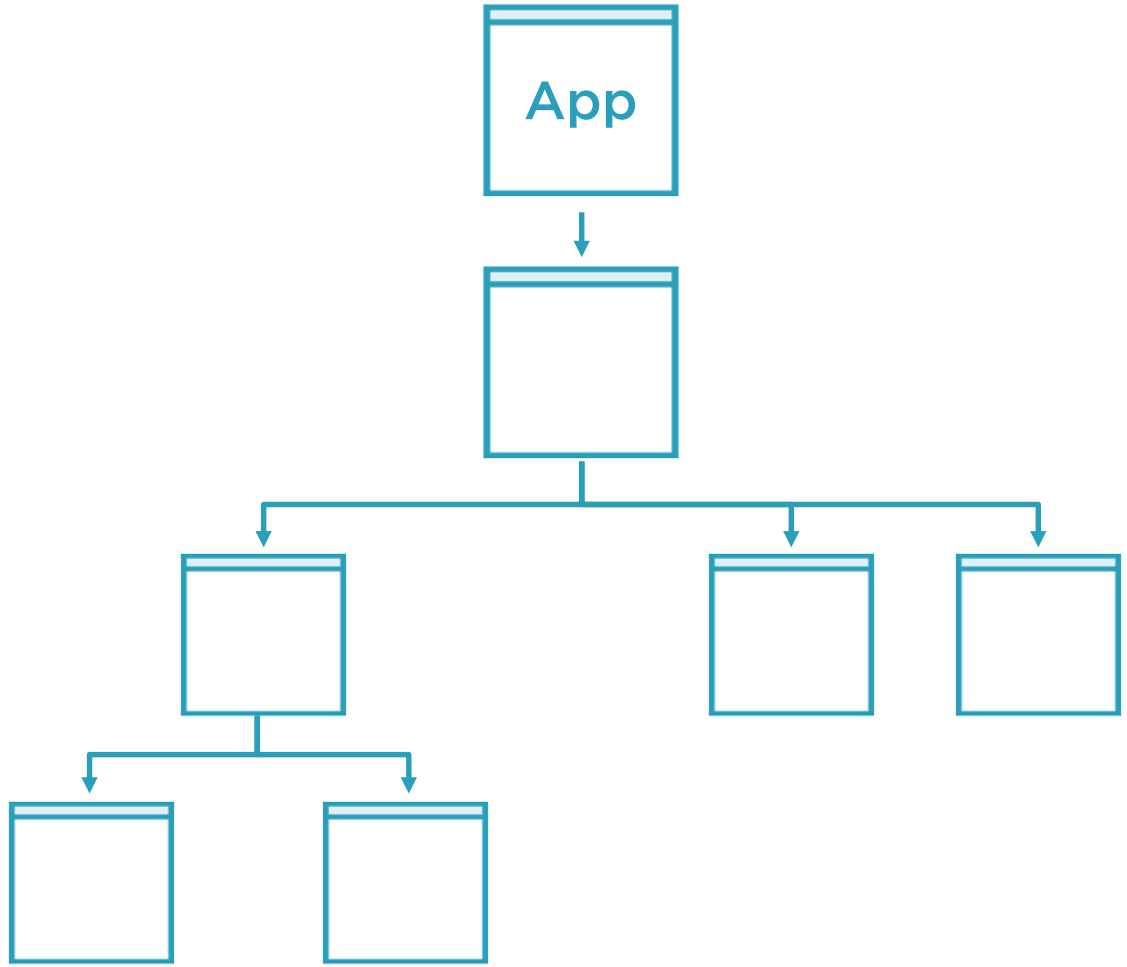
Angular 2 Component Hierarchy



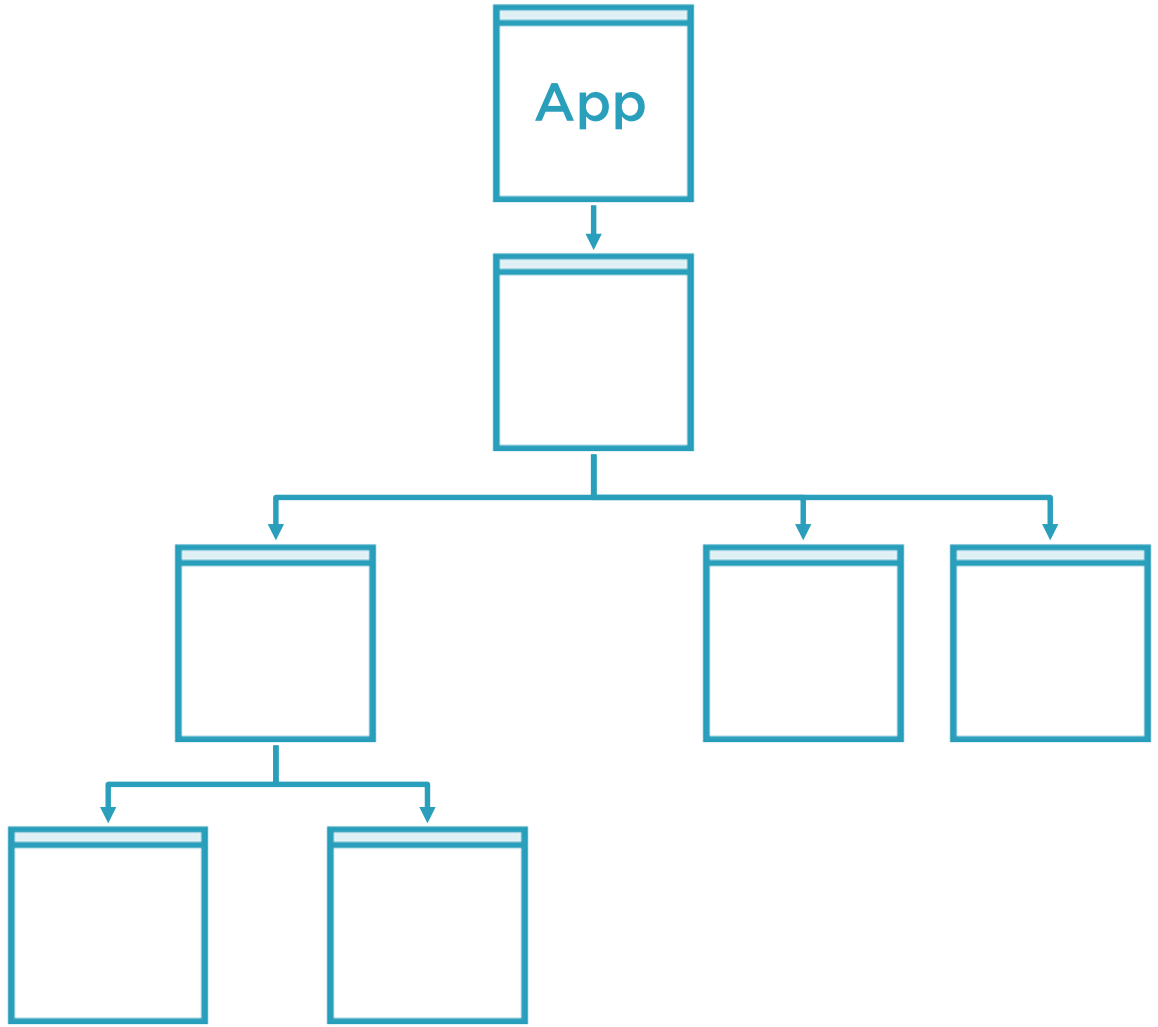
Angular 2 Component Hierarchy



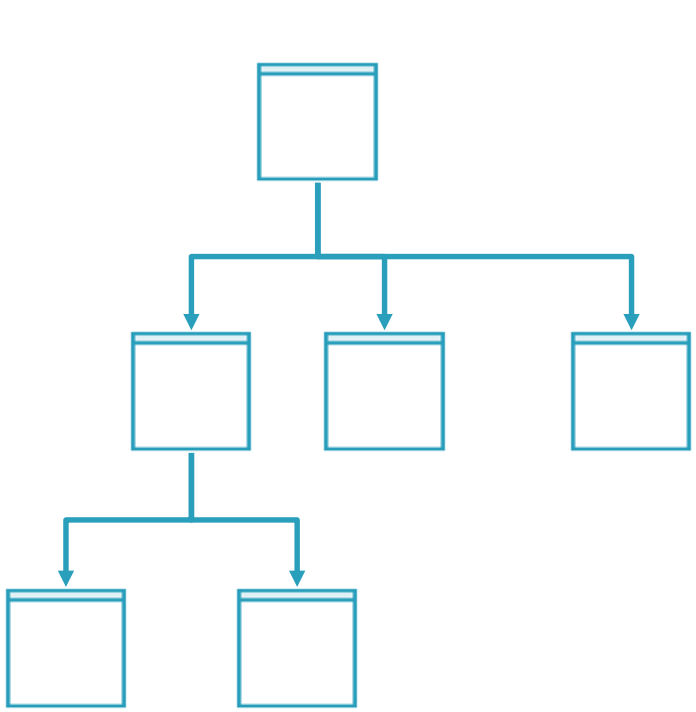
Angular 2 Component Hierarchy

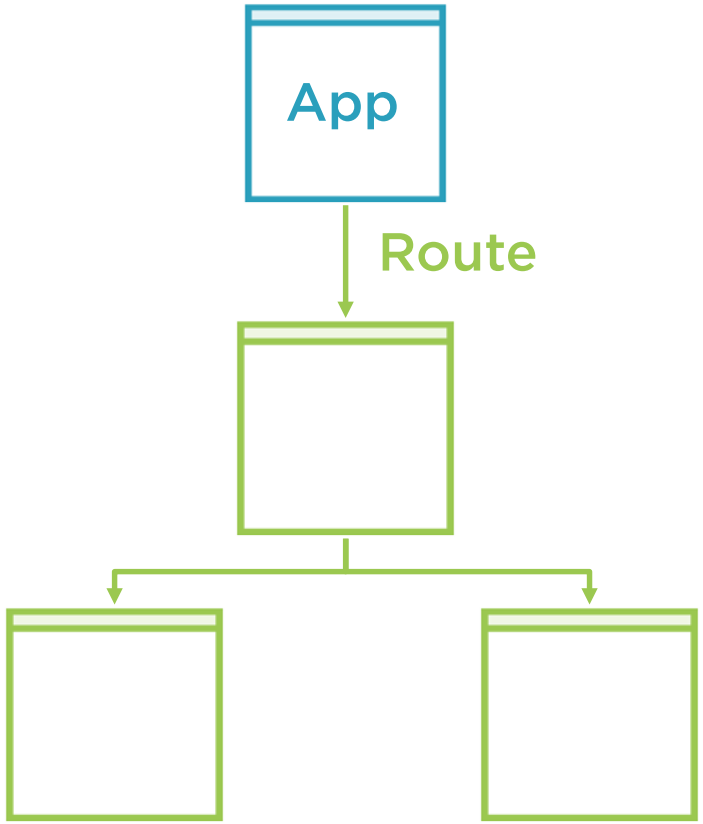


Angular 2 Component Hierarchy

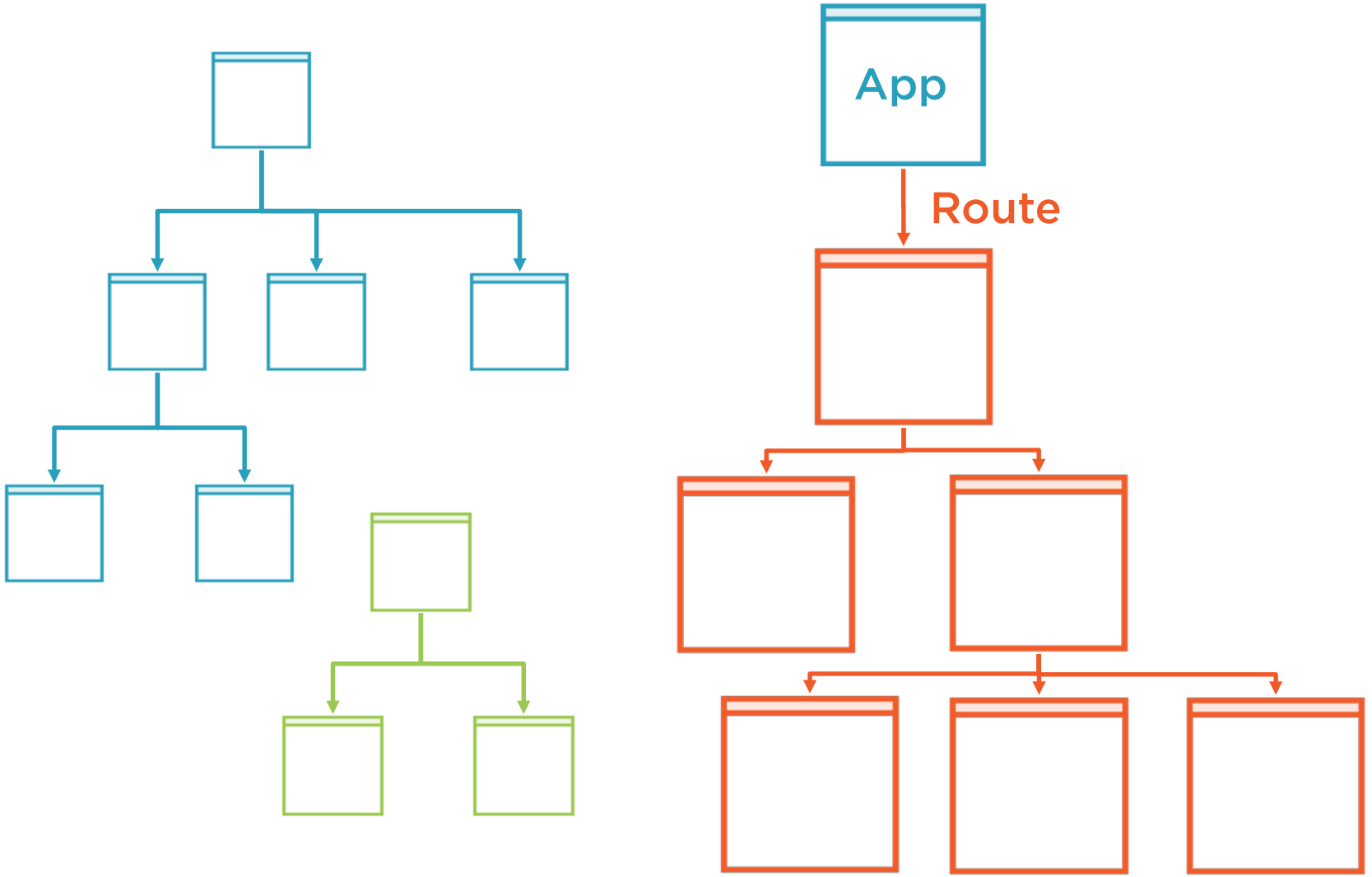


Angular 2 Component Hierarchy

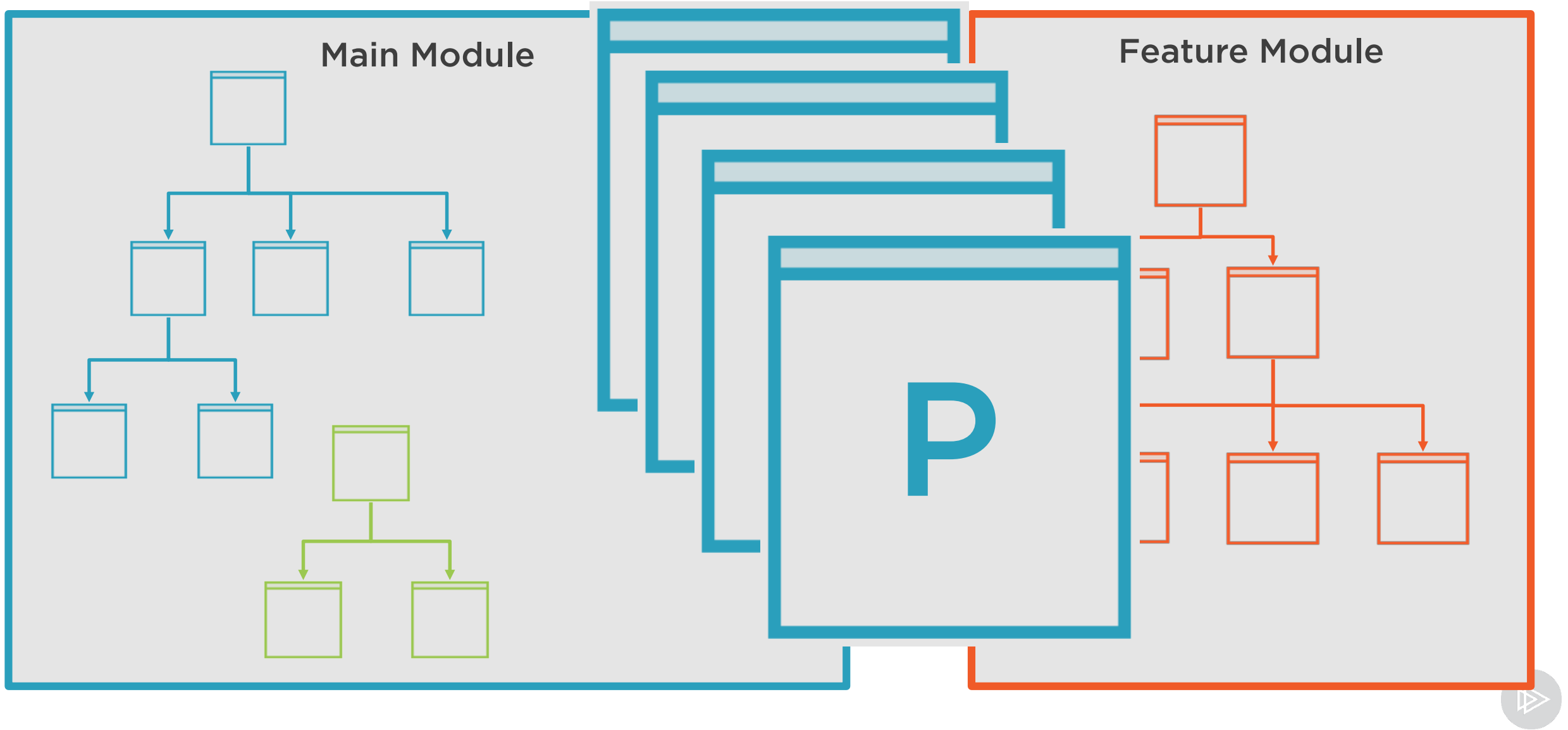




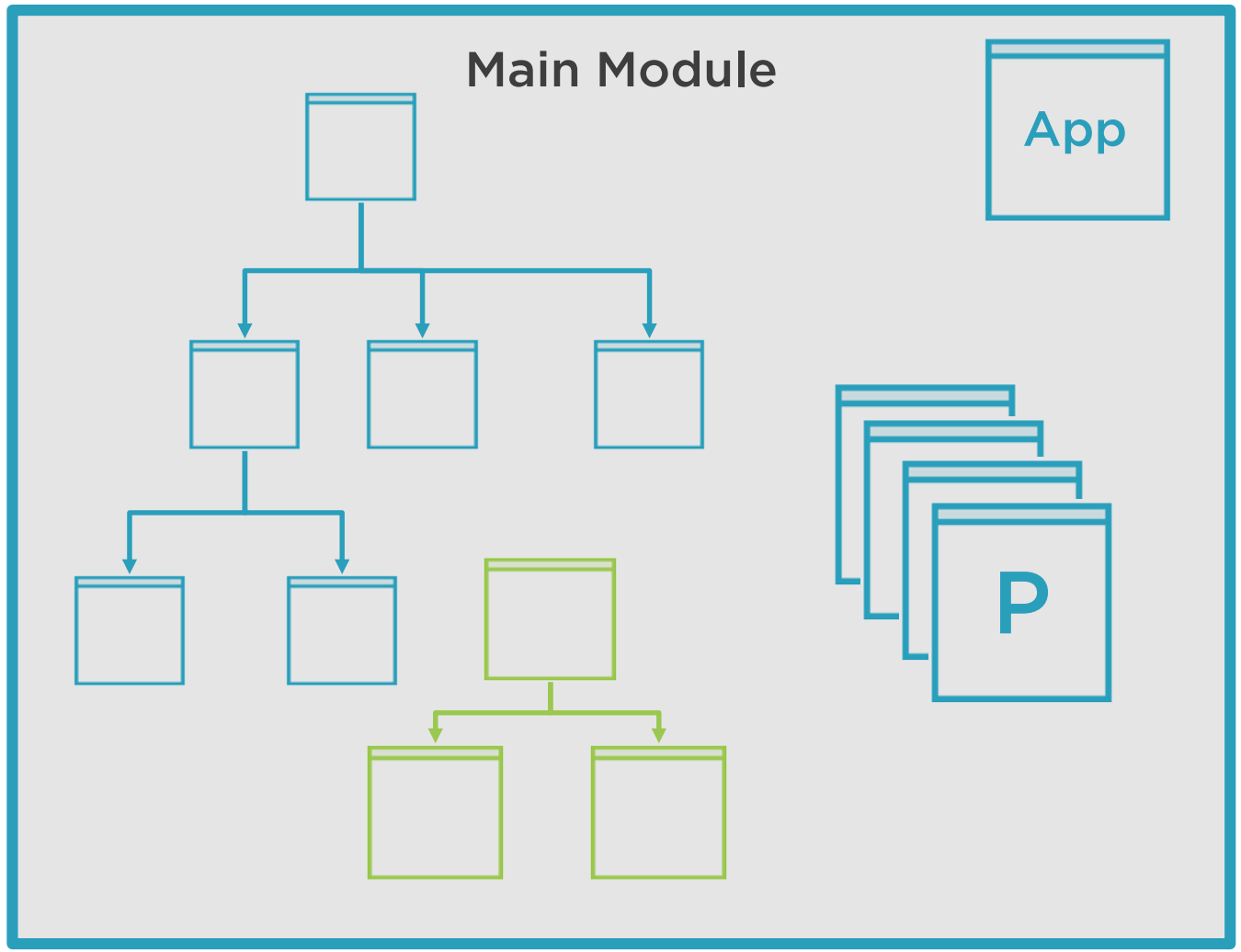
Angular 2 Component Hierarchy

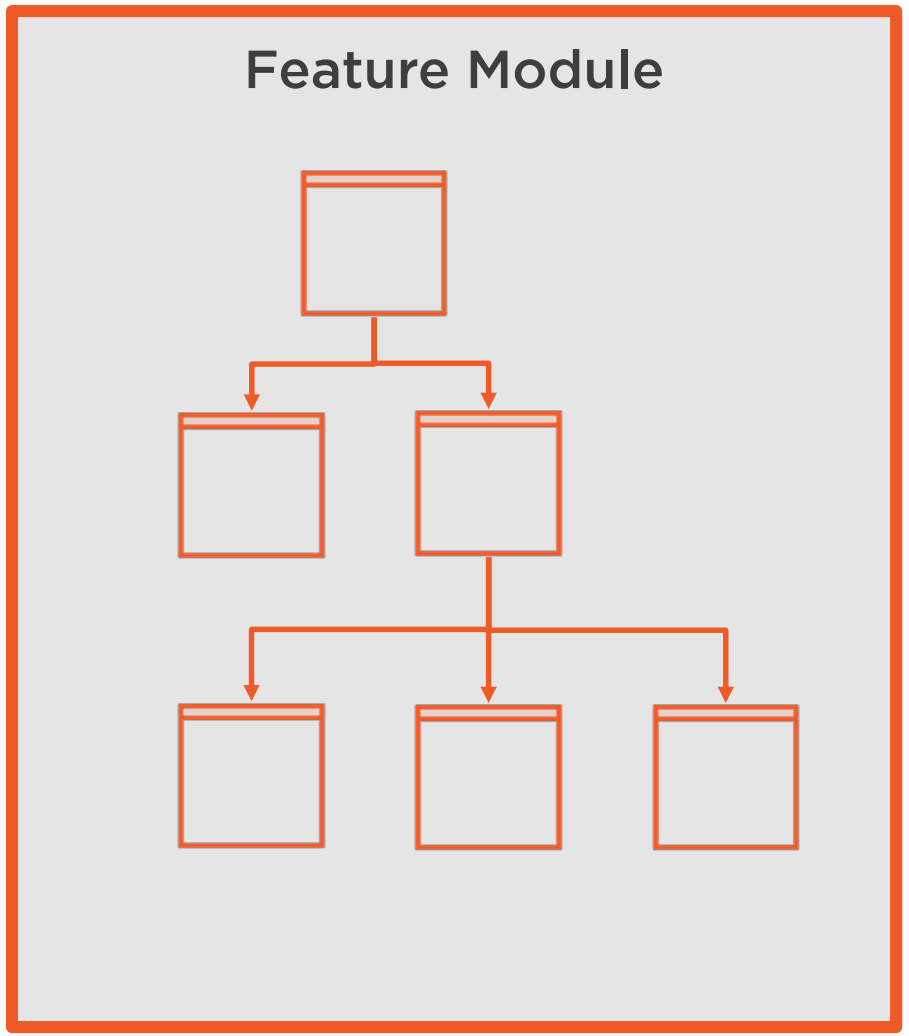


Angular 2 Component Hierarchy

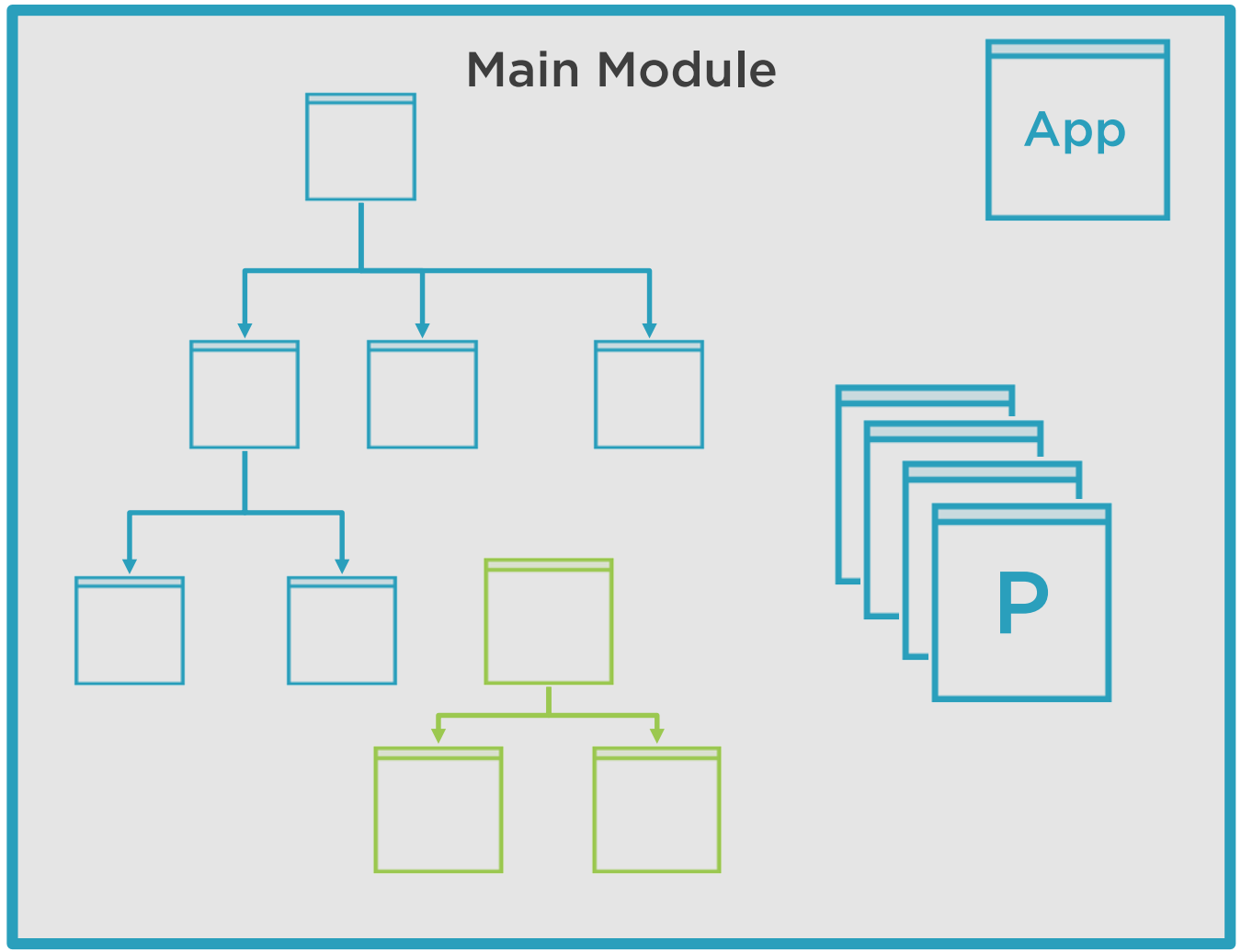


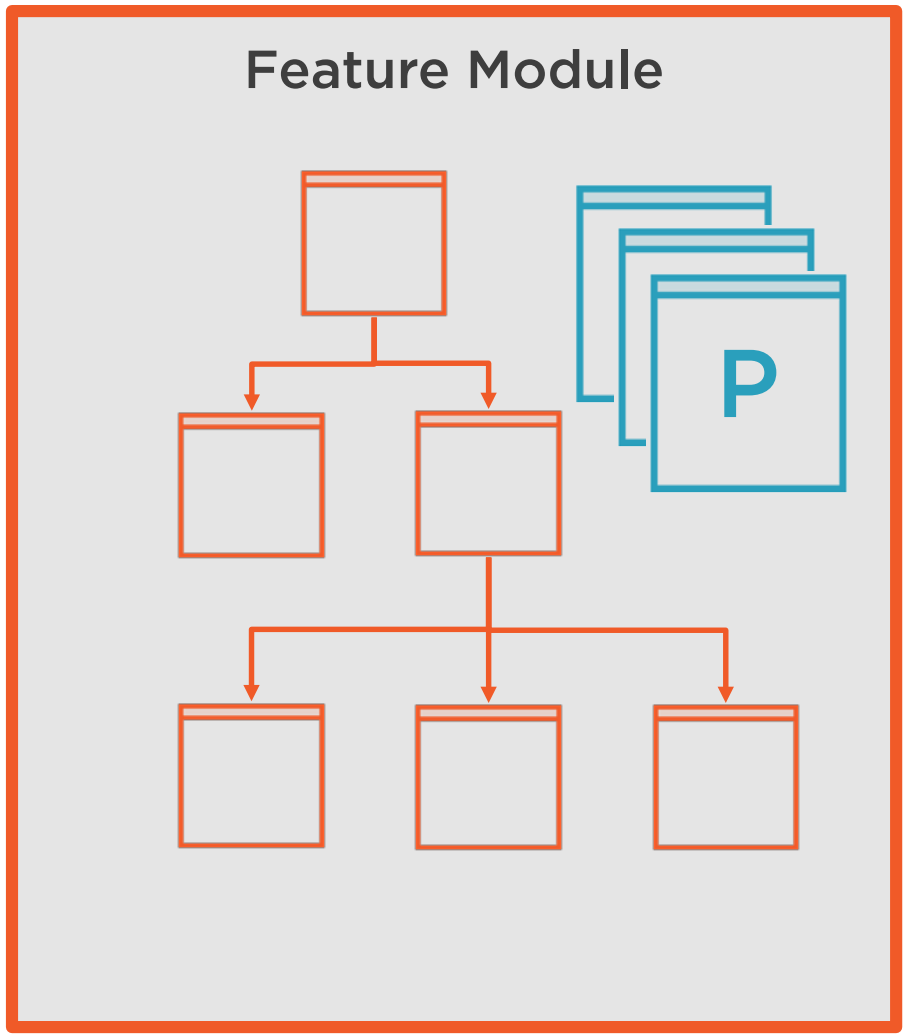
Angular 2 Component Hierarchy



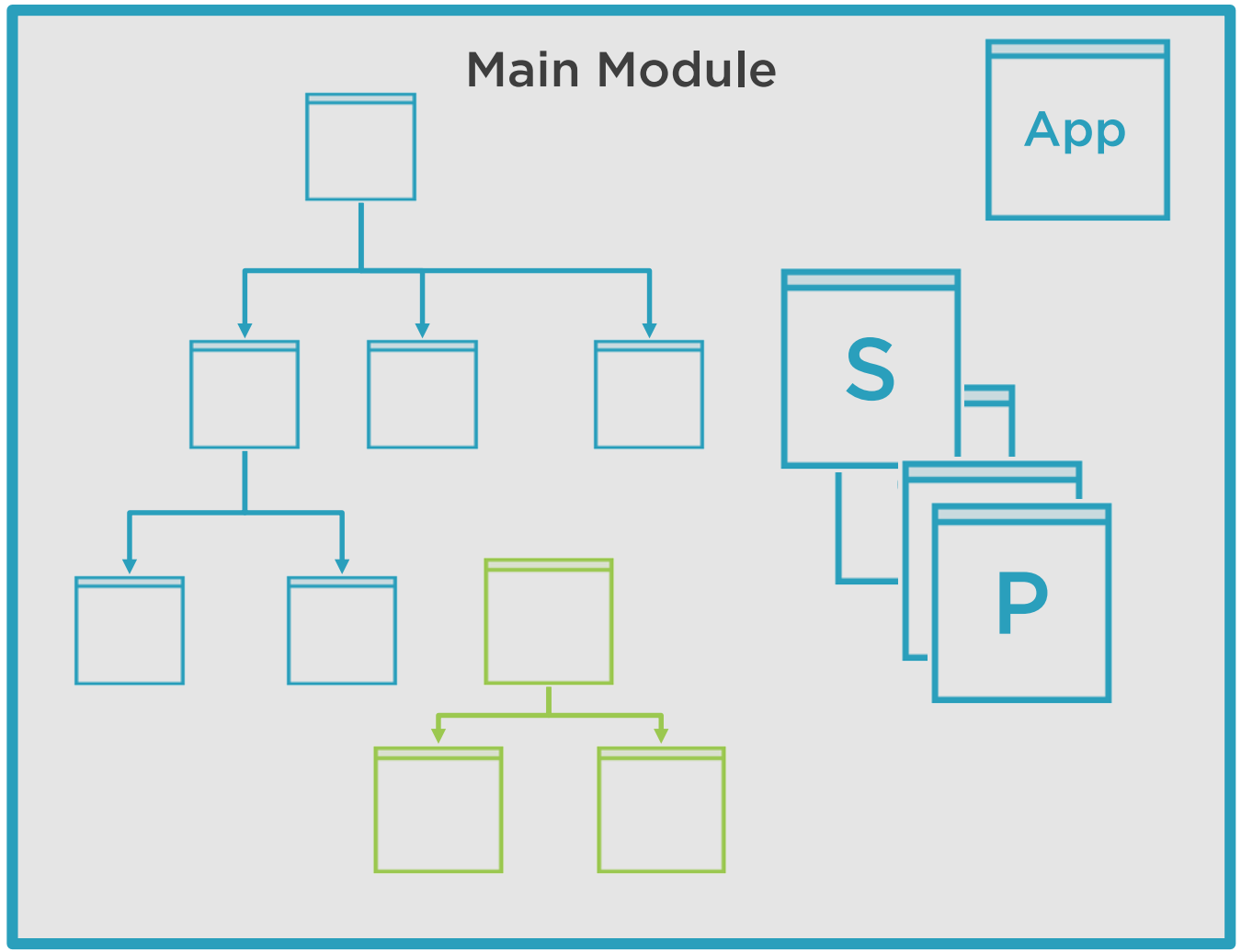


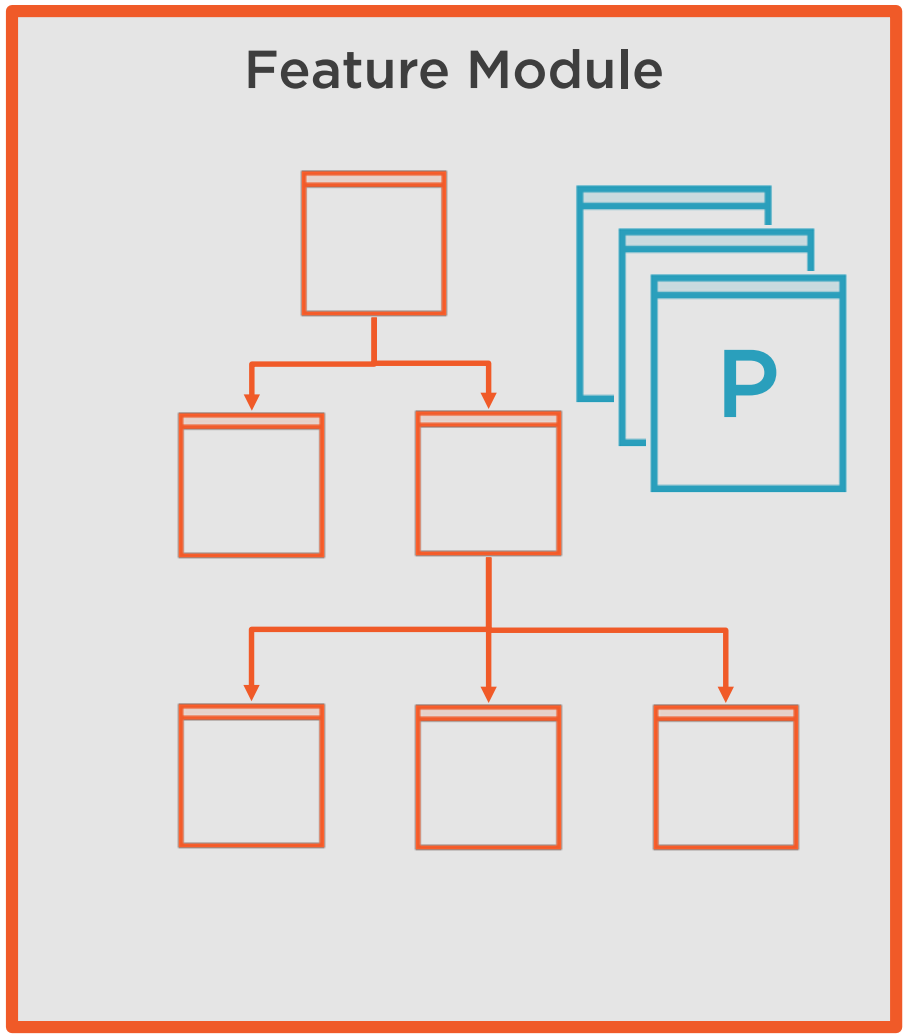
Angular 2 Component Hierarchy





Angular 2 Component Hierarchy





Summary

- Prerequisites
- Conceptual Overview
- Bootstrapping a New App
- Creating our First Component
- Angular CLI