Angular Course

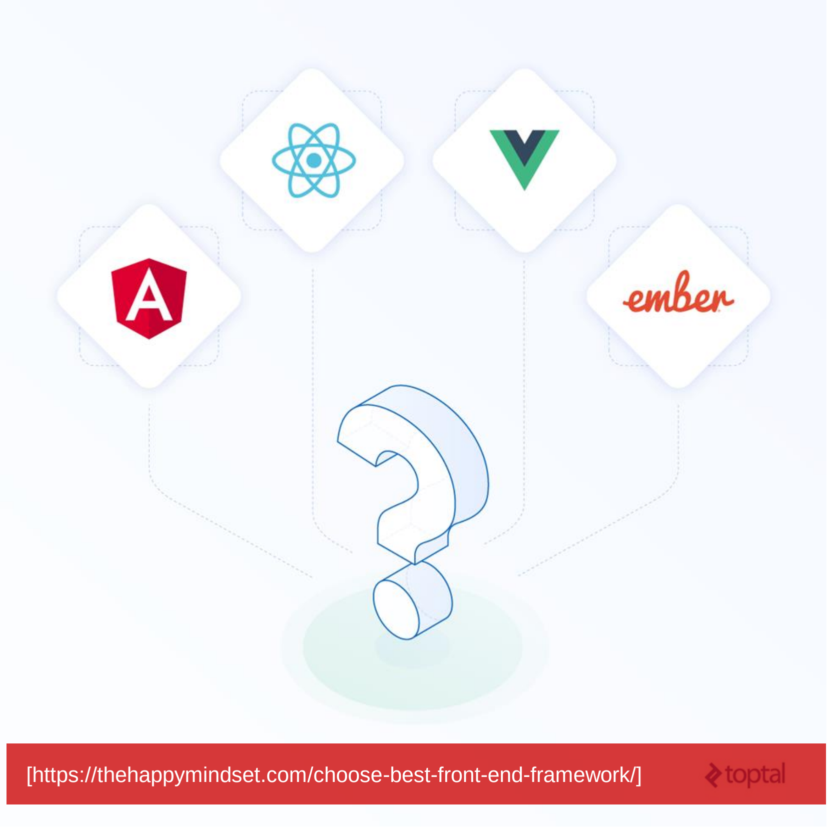
In this course you will get a hands-on of the major features of Angular

David Oliveira
Senior Consultant

23th January, 2018

Why Angular?

- Availability of Learning Resources
- Popularity
- Core Features
- Usability
- Ease of Integration (with Other Libraries)
- Outputs: Web, Mobile, Desktop



[https://thehappymindset.com/choose-best-front-end-framework/]

Angular Basics

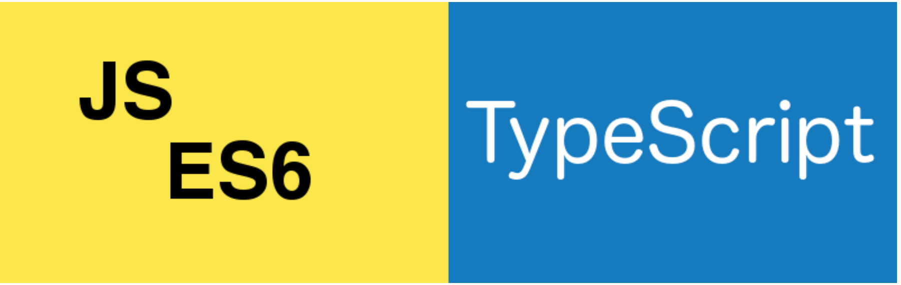
- Components
- Forms
- Directives
- Pipes
- Services
- Routing
- Observables

Angular Non-Basics

- App_Initializer
- Redux in Angular
- Unit Testing
- The mechanics of DOM updates in Angular (View encapsulation)
- Analyze an Angular App’s Bundle Size
- Etc.

TypeScript

```
function doSomething(someData)
{
    // do something
}
```

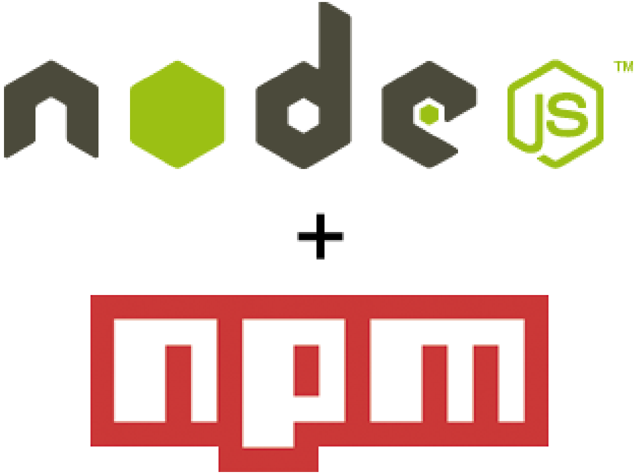


Type and Properties of someData?

Node JS + NPM

Node JS

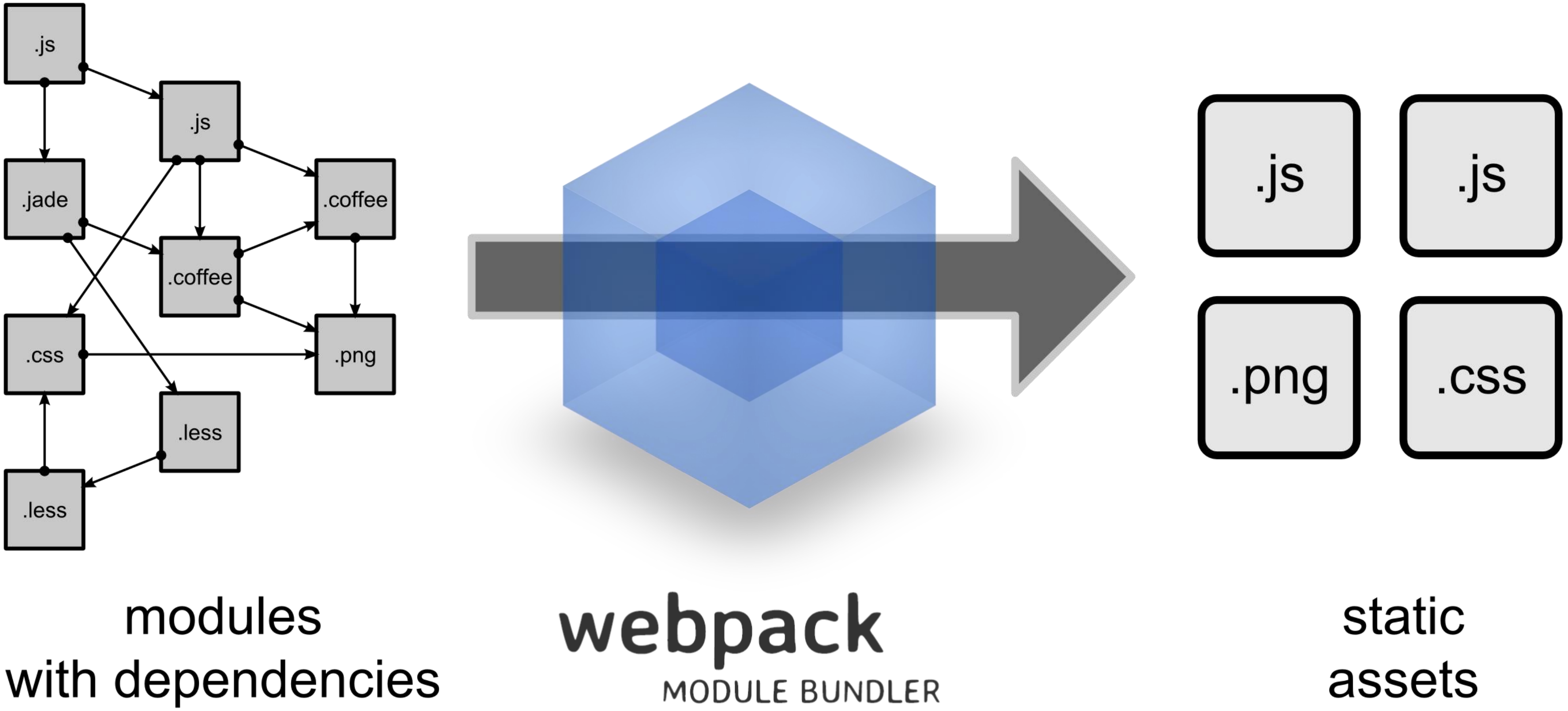
Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.



NPM

Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Webpack



modules
with dependencies

static
assets

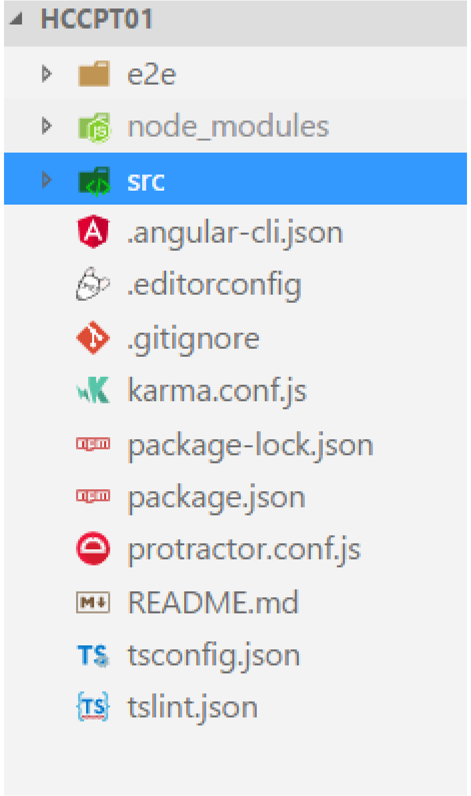
Angular CLI

The Angular CLI is a tool to initialize, develop, scaffold and maintain Angular applications

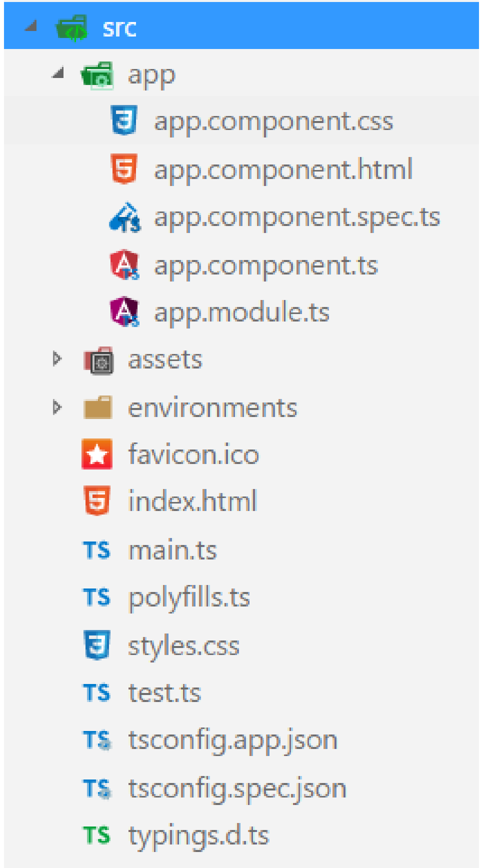
```
npm install -g @angular/cli

ng new hccpt
cd hccpt
ng serve
```

Navigate to http://localhost:4200/.
The app will automatically reload if you change any of the source files.

Angular Solution File Structure

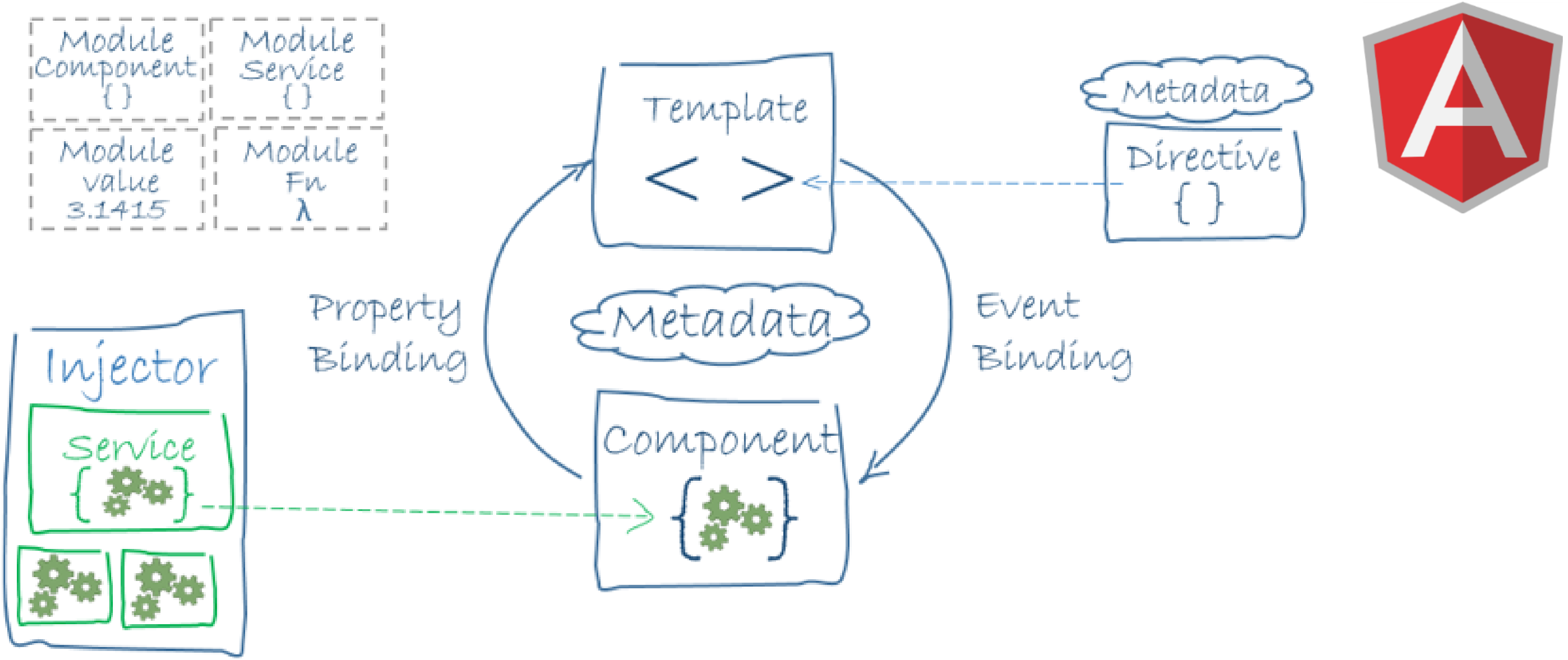


```
◢ HCCPT01
  ▷ e2e
  ▷ node_modules
  ▷ src
    .angular-cli.json
    .editorconfig
    .gitignore
    karma.conf.js
    package-lock.json
    package.json
    protractor.conf.js
    README.md
    tsconfig.json
    tslint.json
```



```
◢ src
  ◢ app
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
      app.module.ts
  ▷ assets
  ▷ environments
    favicon.ico
    index.html
    main.ts
    polyfills.ts
    styles.css
    test.ts
    tsconfig.app.json
    tsconfig.spec.json
    typings.d.ts
```

Angular Architecture Overview



Angular Course

In this course you will get a hands-on of the major features of Angular

David Oliveira
Senior Consultant

30th January, 2018

Our App to build with Angular

We will build an application to manage a Condominium Administration with the following features:

- Add/Edit Contacts (Owners and Suppliers)
- Add/Edit Payments
- Dashboard with Charts about Payments

Create Application

https://github.com/davidoliveira/AngularCourse

```
ng new KondominioApp --routing --style=scss

cd KondominioApp

npm install bootstrap@next --save

npm install --save @ng-bootstrap/ng-bootstrap

ng serve
```

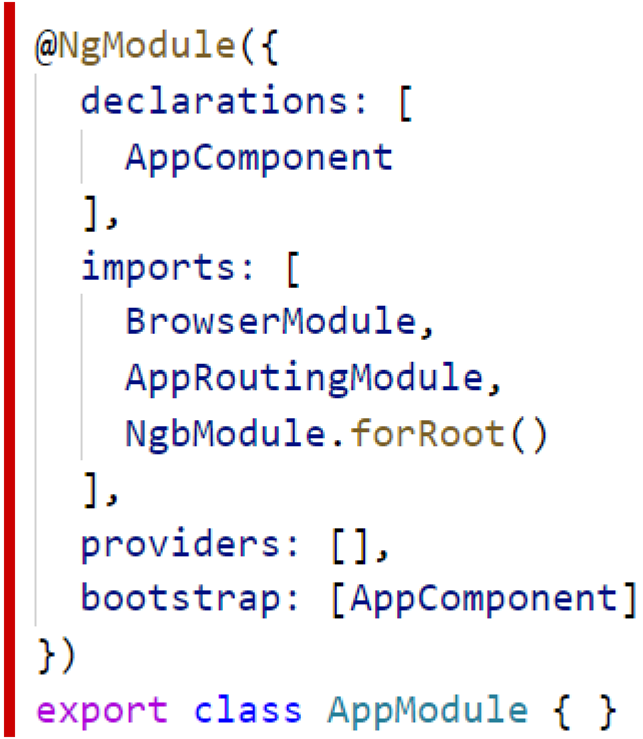
https://github.com/angular/angular-cli/wiki/generate

https://github.com/angular/angular-cli/wiki/stories

Angular Modules

https://angular.io/guide/ngmodules

https://angular.io/guide/ngmodule-vs-jsmodule



```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    NgbModule.forRoot()
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

declarations—this application's lone component.

imports—import BrowserModule to have browser specific services such as DOM rendering, sanitization, and location.

providers—the service providers.

bootstrap—the *root* component that Angular creates and inserts into the index.html host web page.

A declarable can only belong to one module, so only declare it in one @NgModule. When you need it elsewhere, import the module that has the declarable you need in it.

Angular Modules

ng g component Login

ng g module Contacts --routing

- ng g component Index
- ng g component Detail

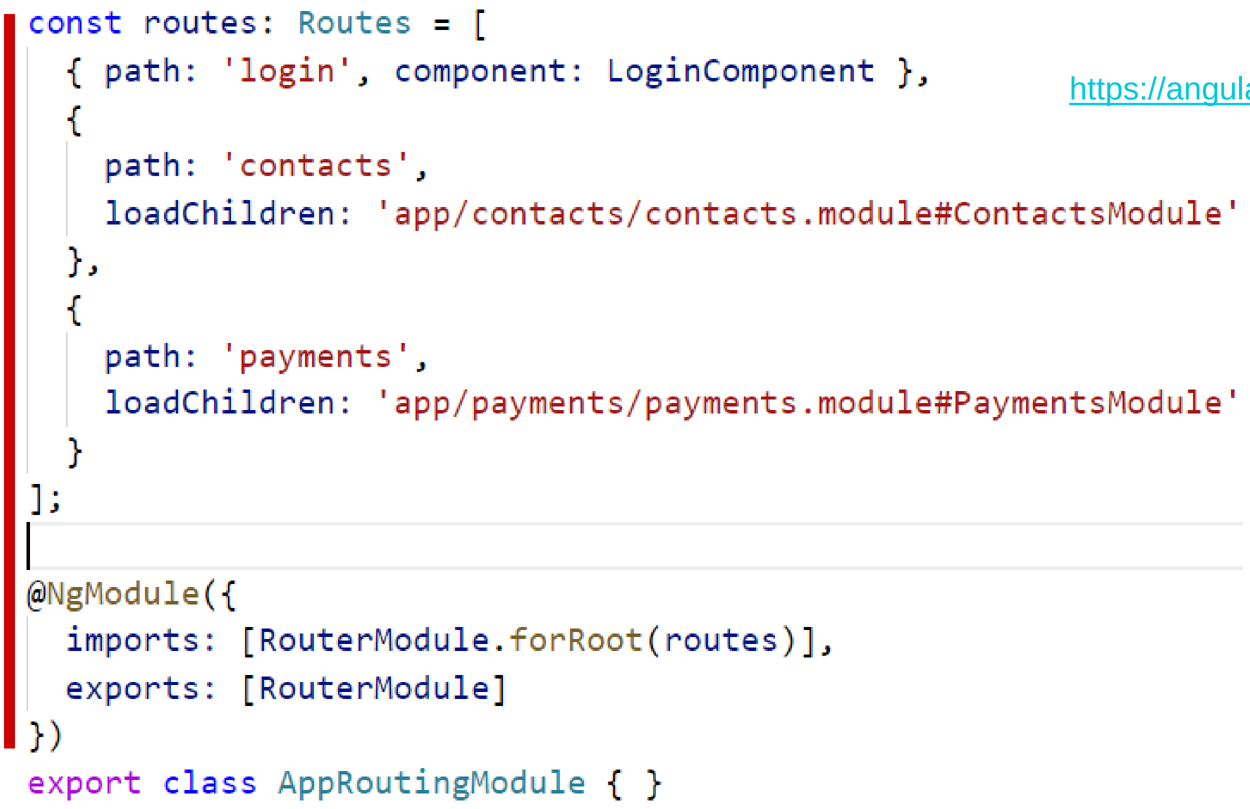
ng g module Payments --routing

- ng g component Index
- ng g component Detail

Angular Routing

https://angular.io/guide/router

https://angular.io/guide/lazy-loading-ngmodules



```
const routes: Routes = [
  { path: 'login', component: LoginComponent },
  {
    path: 'contacts',
    loadChildren: 'app/contacts/contacts.module#ContactsModule'
  },
  {
    path: 'payments',
    loadChildren: 'app/payments/payments.module#PaymentsModule'
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Angular Course

In this course you will get a hands-on of the major features of Angular

David Oliveira
Senior Consultant

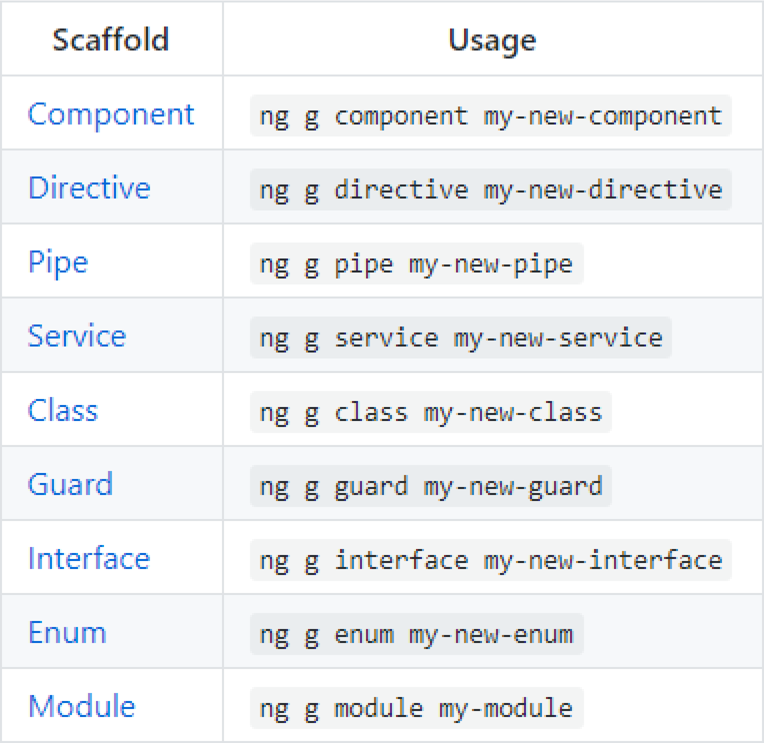
5th February, 2018

Resume

Last Week

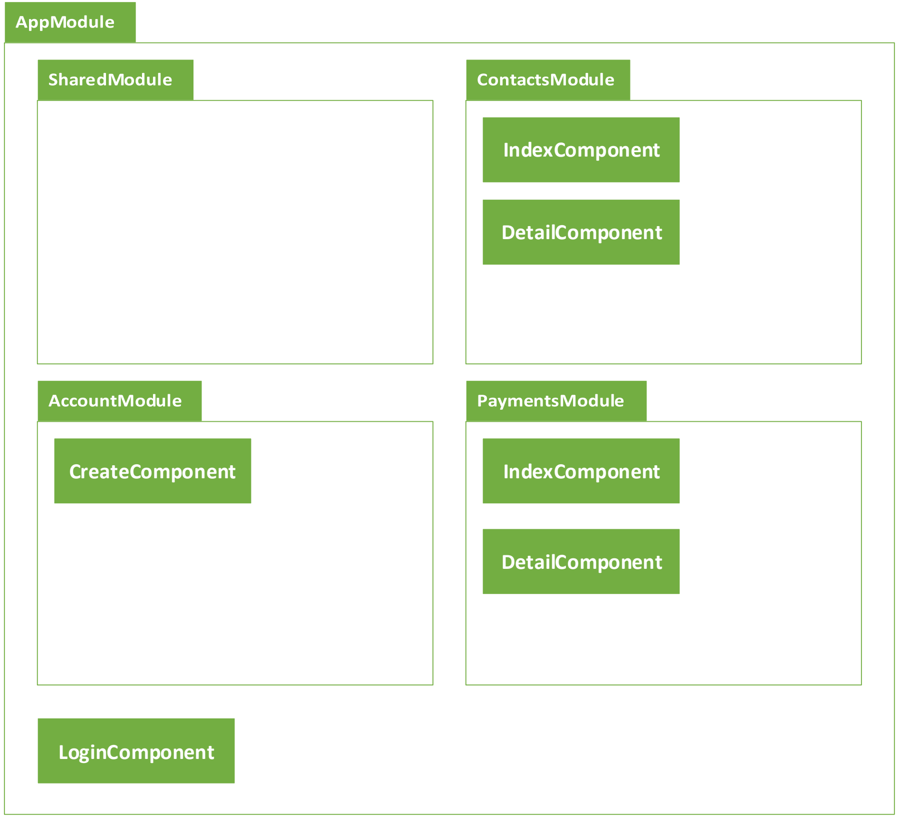
- Setup our App to manage a Condominium
- Add Modules and Components for each Module
- Setup Routing

Angular CLI



Scaffold	Usage
Component	`ng g component my-new-component`
Directive	`ng g directive my-new-directive`
Pipe	`ng g pipe my-new-pipe`
Service	`ng g service my-new-service`
Class	`ng g class my-new-class`
Guard	`ng g guard my-new-guard`
Interface	`ng g interface my-new-interface`
Enum	`ng g enum my-new-enum`
Module	`ng g module my-module`

Our App Structure



AppModule

SharedModule

ContactsModule

IndexComponent

DetailComponent

AccountModule

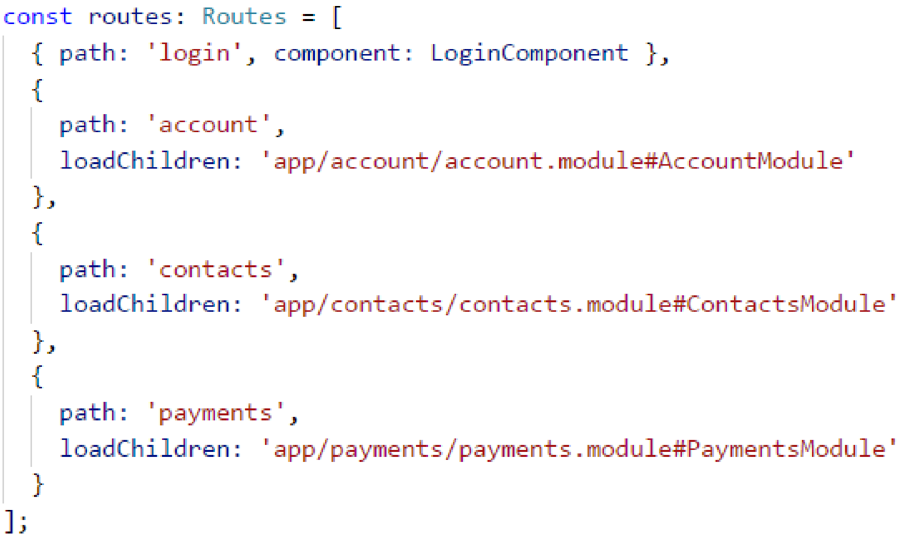
CreateComponent

PaymentsModule

IndexComponent

DetailComponent

LoginComponent



```
const routes: Routes = [
  { path: 'login', component: LoginComponent },
  {
    path: 'account',
    loadChildren: 'app/account/account.module#AccountModule'
  },
  {
    path: 'contacts',
    loadChildren: 'app/contacts/contacts.module#ContactsModule'
  },
  {
    path: 'payments',
    loadChildren: 'app/payments/payments.module#PaymentsModule'
  }
];
```

Routing configuration to navigate through our application. Check Angular documentation about it:

https://angular.io/guide/feature-modules
https://angular.io/guide/lazy-loading-ngmodules

Questions and Discussion

HITACHI

Inspire the Next