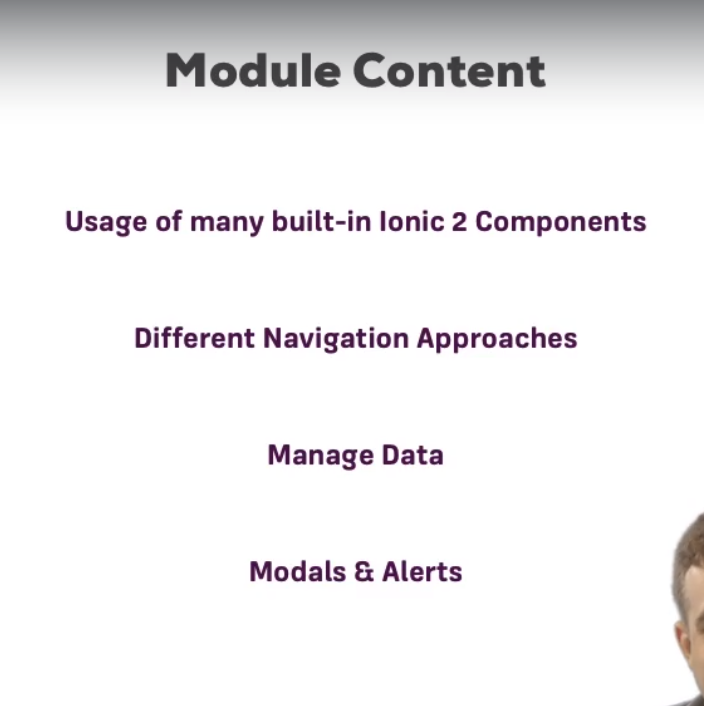
38)Module introduction

This is what we are going to cover in this module.



We will have extensive look at components ionic 2 ships with modals, alerts, cards and lot more. You will learn how to use them. Where to find them and how they actually work.

We will also have look at navigation, but not just basics navigation but how tabs work, how we can implemt side menu and how they differs to normal navigation

We will also have a look at how we can manage data. How we can use angular service sin ionic app and how that will work.

Then we will have a look at modls and alerts. I have already mentioned them but it’s important. Because these are some key faetues how we can not only show them but also how we can really use them by passing data to them and fetching data we get from them there.

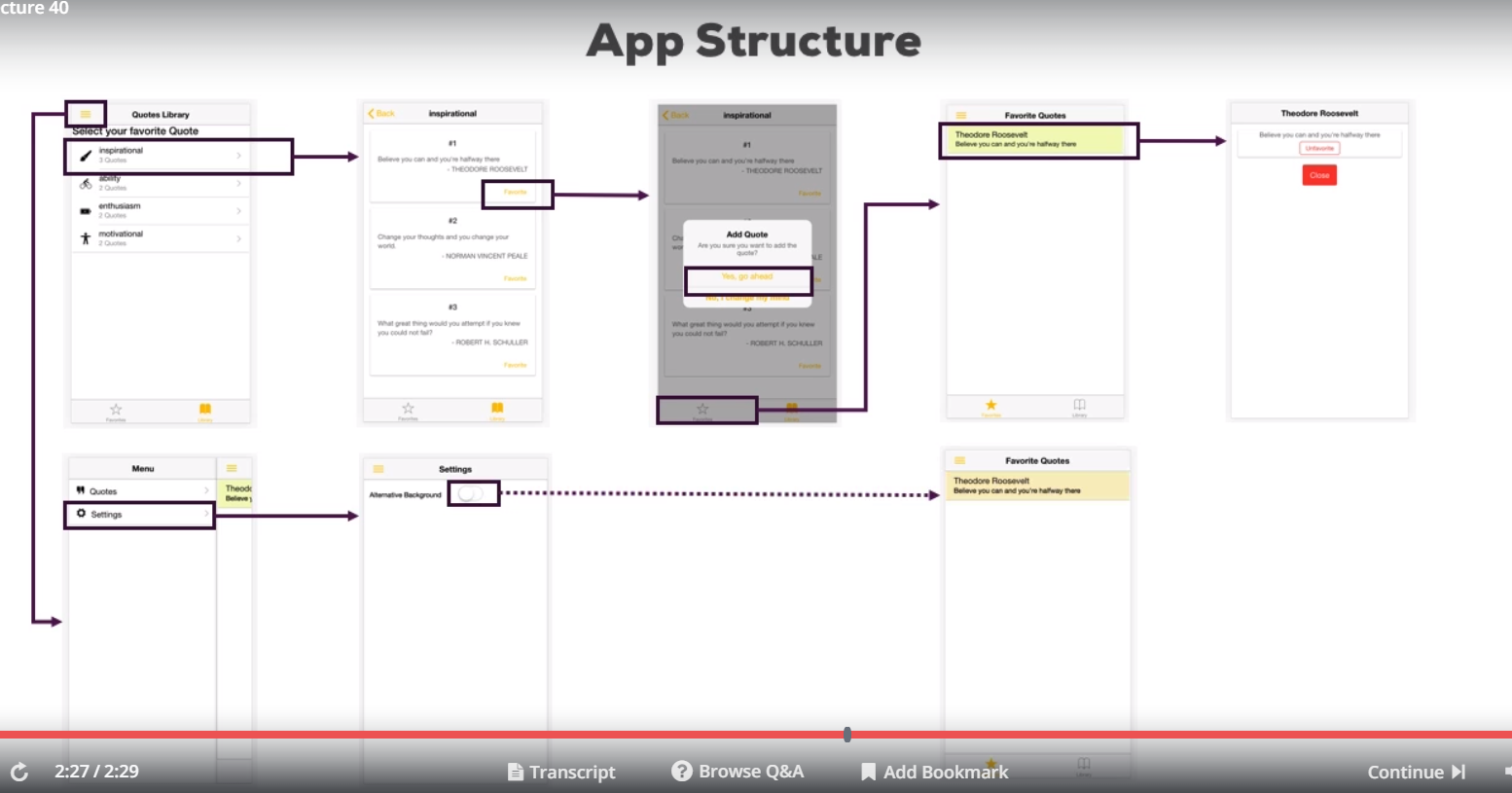
39)what we are going to build

See video

40)Breaking the app into pieces(Defining the app structure)

Here we saw which screens we need and which pages are required. So as a first step we will start creating pages that are required.

For details see lecture, but it was just about which pages we need.



41)Creating the required pages

We will start by creating a blank app. So run this-

**ionic start quote-app blank**

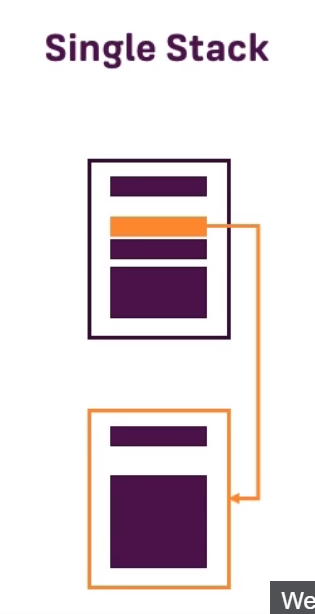
then we create page for library, which displayes the course categories.then we create quotes pages which displayes the quotes of selected categories. Then we create quote page which will be displayed when click on our quote on favorites page, displaying a modal. Then we create favorites page, which displays all our fav quotes. The last but not least we need setting page. Actually soon,we need a additional but I will come back to this. Thenw e delete the home page of our app. This will break our app but we will soon fix it. Now we get all pages we need , we will use tabs navigation to hook them up. First lets make fav page as our root page . this needs to be done in app.compoent.ts. then we remove HomePage from app module. Note if you are using ionic 2 you need to add import of newly created pages in app.module in declarations and entryCompoents. But in version 3 its not required as we explained earlier.

The next step is to setup tabs navigation. We nt know about it. We will see it next lecture.

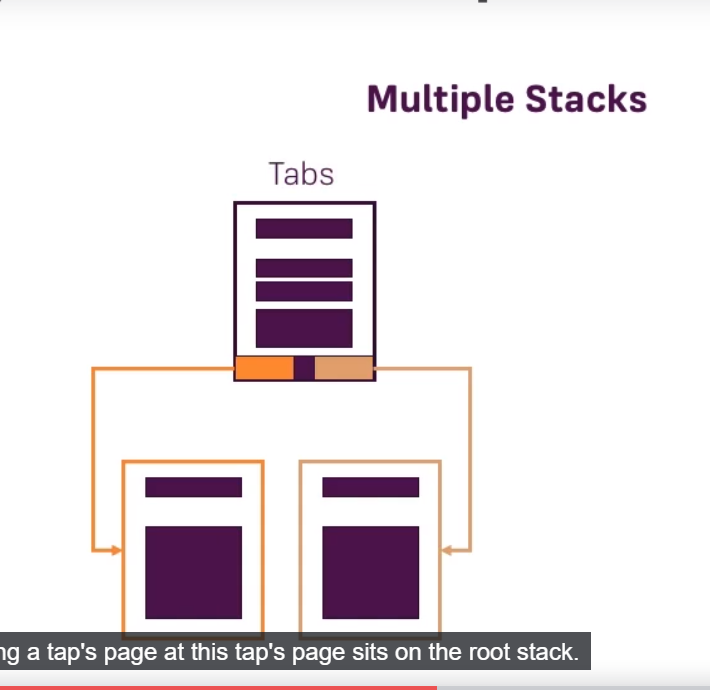
42)Multiple Stacks of pages vs One Single Stack

We created all pages that we needed in ur app. They are still rw we have not added any thing in them. As of now our app is not working great. but right now we can see only fav page. We do not have any tabs to navigate around and we got no side menu. I will come back to side menu , here I want to look at tabs. Tabs navigation is something we have not looked at yet. lets look t it now.

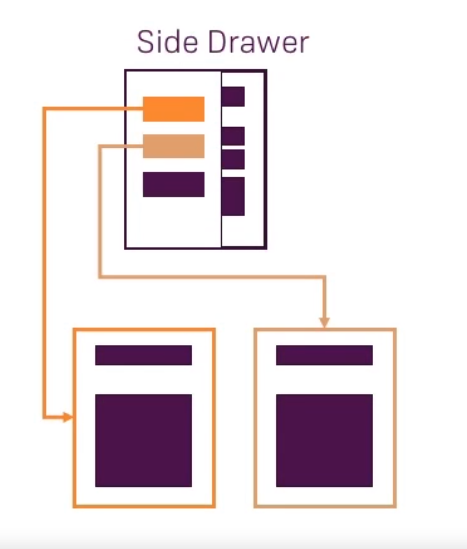
To understand how we can navigate around with tabs or side menu we have to go back to navigation stack, the stack of pages the ionic 2 uses. Until now we saw the single stack, that means if we click on button, we push a new page on that stack and we view that page. The page in this case.



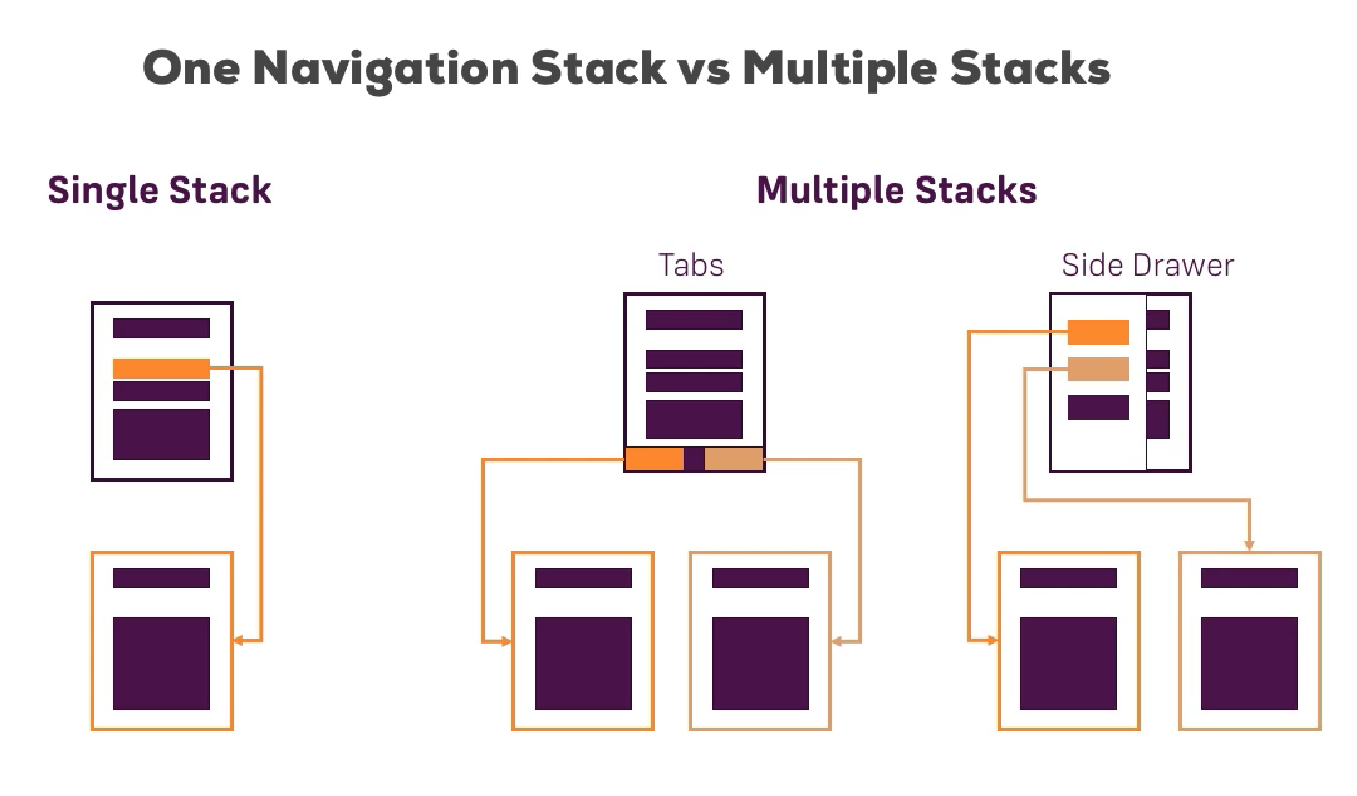
Now ionic 2 is also bale to handle multiple stacks , kind of. It always have one root stack, but inside of stack you can split it up lets see example.



Here we are using the tabs page. This Tabs page sits on root stack. However when we click a page, click a tab here I should say, it will load 2 different sub stacks of pages. So if the page loaded through tab we navigated around and push pages on stack, this would be on stack for this tab and if we then switch the tab we will be on totally new stack of pages and you will see this in action soon . as a side note you will also later see this later when we implement the side menu because there we will also have 2 different stack of pages.



This is theory. Lets actually implement this by actually adding tabs to our application.



43)Implementing tabs navigation with multiple stacks of page

now we did learned that we can use tabs to split up the navigations stack. To stack of pages , to manage different stacks of pages. Lets add tabs. Actually tabs are added as epaarte pages. So in pages folder we craete tabs folder. Or just generate a page called tabs from cli. In this page we have ts, file which is normal component.

Now we also have template of page. Now how does this template look like? We saw other pages always have <ion-header> and then <ion-content> and so on. Now tabs page is different because tabs whilest placed in their own page in ionic 2, simply give us a frame to load other pages inside them. Keep in mind that tabs are at bottom of page so they don;t actually make up full page, only fraction of it. But then they will enclose the page which gets loaded through the tabs. This setup is necessary because if the tabs page would be a totally separate page and would take up the full page then click on the tab will remove that page and we would see the loaded page instead. Which we do not want to do.

So we have ion-tabs components in template. **Ion-tabs**  is special component ionic 2 shipd with which allows us to , well set up some tabs. Inside of ion-tabs we will cretae individual **ion-tab .** So ion-tabs is the warpper , and inside of it we define which tabs we actually want to have. Now ion-tab also needs some configuration. We set up root proeprty using property binding, now as a value we give it a page. So that page will be loaded when we click on that tab. That page will be loaded in empty space that tabs page gives us.

So you can think of tabs compoent not only giving us tabs at bottom but also a placeholder where actual loaded page , depending upon which tab you clicked on, will be loaded in. And this is important if you keep the navigation concept in mind, keep in mind that top most pages is the page you view. So gain if clicking a tab would mean that we simply push a new page on stack, it would also mean that we make tabs disappear, because newly pushed page wnt have tabs at bottom. Therefore w ehave this different setup where we instead split up the navigation stack and have separate navigation stack rendered inside of this tabs page, kind of in its empty placeholder. Althorugh tab without title and iocn is boring , so we add title with tabTitle property. W ecan also add TabIcon property to add icon. We can use either of these or both of these. Now what do we assign as a value to tabIcon property? We can find it out in official ionic docs. Go to docs , click on icon search favorite. Now there we have this start icon, now left side of start, we can see name of icon, we need to assign that as value to tab-icon proeprty. Now we will have tab with start icon and favorite text below it or depending on which platform you are at, this will be placed automatically. Thenw e add a new tab, the library tab. Code-

tabs.ts-

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

@IonicPage()

@Component({

selector: 'page-tabs',

templateUrl: 'tabs.html',

})

export class TabsPage {

favoritesPage = 'FavoritesPage';

libraryPage = 'LibraryPage';

}

tabs.html-

<ion-tabs>

<ion-tab [root]="favoritesPage" tabTitle="Favorites" tabIcon="star"></ion-tab>

<ion-tab [root]="libraryPage" tabTitle="Library" tabIcon="book"></ion-tab>

</ion-tabs>

now one last step is we want to make this tabs page as our root page instead of fav page. You know how to do this app.component.ts. Lets run app-

now how does this ionic which tab to load? It automatically loads the first tab. Lets say we want to have second tab loaded by default.

Then use selctedIndex property on ion-tabs.

<ion-tabs selectedIndex="1">

<ion-tab [root]="favoritesPage" tabTitle="Favorites" tabIcon="star"></ion-tab>

<ion-tab [root]="libraryPage" tabTitle="Library" tabIcon="book"></ion-tab>

</ion-tabs>

default index starts at 0. if you want to dig deeper into tabs. Then go to docs, the click on api, then go to tabs. Here you can learn how to place it, how to configure it, how to style it . To really fine tune it according to your needs. General functionality is shown here though. However general funcionality is shown here though.

44)Adding quote data to app

here we added a new folder data in src. This folder has quotes.ts file which contains array of quotes. We want to us these quotes in fav tab.

Quotes.ts-

export default [

{

category: 'inspirational',

quotes: [

{

id: '1',

person: 'Theodore Roosevelt',

text: 'Believe you can and you\'re halfway there'

},

{

id: '2',

person: 'Norman Vincent Peale',

text: 'Change your thoughts and you change your world.'

},

{

id: '3',

person: 'Robert H. Schuller',

text: 'What great thing would you attempt if you knew you could not fail?'

}

],

icon: 'brush'

},

{

category: 'ability',

quotes: [

{

id: '4',

person: 'John Wooden',

text: 'Ability may get you to the top, but it takes character to keep you there.'

},

{

id: '5',

person: 'Robert Frost',

text: 'Education is the ability to listen to almost anything without losing your temper.'

}

],

icon: 'bicycle'

},

{

category: 'enthusiasm',

quotes: [

{

id: '6',

person: 'Benjamin Disraeli',

text: 'Every product of genius must be the product of enthusiasm.'

},

{

id: '7',

person: 'Norman Vincent Peale',

text: 'Enthusiasm releases the drive to carry you over obstacles and adds significance to all you do.'

}

],

icon: 'battery-charging'

},

{

category: 'motivational',

quotes: [

{

id: '8',

person: 'Jim Rohn',

text: 'Either you run the day or the day runs you.'

},

{

id: '9',

person: 'Donna Brazile',

text: 'I was motivated to be different in part because I was different.'

}

],

icon: 'body'

}

];

45)Using the quotes data

now we want quotes in our library.ts. We can have variable which will store all quotes, it can be of type any you could say, butactually we do know the structure of this. W edo know how quotes object like. So we can create fittting interfaces for this, so that we can define the types explicitly. So we define new interface in quote.interface.ts in data folder.

Quote.interface.ts-

export interface Quote {

id: string;

person: string;

text: string;

}

Actually we can also create interface for category but actually i am only going to use category in library page, so i can simply define a type in library.ts.

Library.ts-

import { Component, OnInit } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { Quote } from '../../data/quote.interface';

import quotes from '../../data/quotes';

@IonicPage()

@Component({

selector: 'page-library',

templateUrl: 'library.html',

})

export class LibraryPage implements OnInit {

quoteCollection: {category: string, quotes: Quote[], icon: string}[];

ngOnInit() {

this.quoteCollection = quotes;

}

}

in library .ts we assign quotes to quoteCollection. Now quotes is array having all quotes in quotes.ts file. W ehave to import this file(quotes.ts) in librray.ts file to use it. We import it as quotes here, you can choose any name you want here because i am having a default export in quotes.ts, all i am exporting in this file is this array , it is not named export.

Quotes.ts-

export default [

{

category: 'inspirational',

quotes: [

{

id: '1',

person: 'Theodore Roosevelt',

text: 'Believe you can and you\'re halfway there'

},

{

id: '2',

person: 'Norm

now we got data in librray.ts, nect step is to get data in template a a list of items.

46)using the ionic “List” and “List Item” Components

in library.ts, we have data . Now we want that data on librray.html. In htmkl we have ion-header,ion-content . Then we have padding durective in ion-content because i want to have some spacing at the edges.

Then we add h3 with text-center directive to center our text. Then we use new component ion-lists. As you might have guessed it allows us to cretae list. There is important thing, you do not always have to use **ion-list** while ouputting list of items. So you might think that you ion-list needs to be used if you are using \*ngFor on some element, that you have to wrap this with ion-list

<ion-list>

<p \*ngFor="let item of items"></p>

</ion-list>

this is not the case. Ion-list is the compoennt that you can use to style certain things nicer in a list like view. Lets see it in docs, got to api->Lists. There you can see there are laods of ways of styling list and you can go through them to find list which fits your need. You can cobmine it with lot of other compoents, and we are going to see couple of ways throughout this course. Trdation useage is to use ion-list with buttons inside of it.

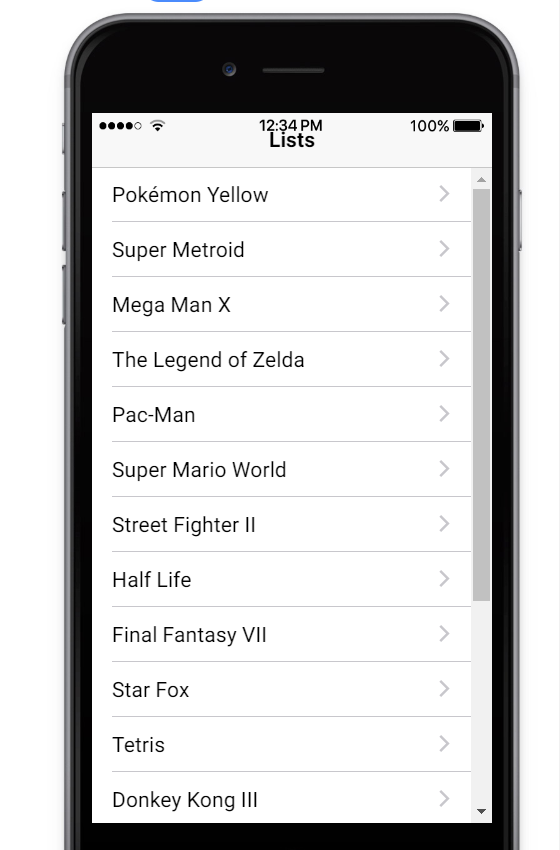
<ion-list>

<button ion-item \*ngFor="let item of items" (click)="itemSelected(item)"> {{ item }}

</button>

</ion-list>

this button have \*ngFor directive on it as you probably wanted to have multiple list items and we can simply handle the click on the button. Button needs to have this ion-item directive on it. This is needed because this connection of having wrapping ion-list element , then having ion-item element inside of it , will lead to this screen on –



ion-liest gives you this structure of having several rows, this nice separator between rows and since we use buttons with ion-item, we get this arrow.

this is all what ion-list in conjuction with button does.

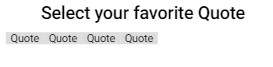
As an alternative to button with ion-item,You can also use ion-list with ion-item inside of it(see in docs under list-dividers), i will come back to it later. Since here we want to have clickable list we will go with button approach. We want to have non clickable lists we will use ion-item instead of button.

<ion-list>

<button \*ngFor="let quoteGroup of quoteCollection">Quote</button>

</ion-list>

This is what we will get if we do not use ion-item directive on buttons of ion-list.



By just adding this directive-

<ion-list>

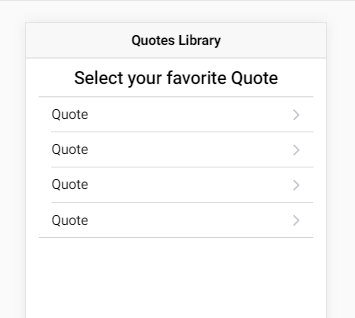
<button ion-item \*ngFor="let quoteGroup of quoteCollection">

Quote

</button>

</ion-list>

we get -



if we remove ion-list but use button with ion-icon then still get nice list but we do not get lines tat start and end. We got basic list but we want to have icons, quote count cat name etc on each list item.

47)How to create more complex list items

in last lecture we created this basic list with button inside of it. Lets make it loook nicer. The cool thing about ion-icon, directive that i place on button, whcih makes this button looks nice, is that we can actually place a lot of content inside of this button and it will automatically allign this nicely. We use another ionic compoennt, **ion-icon**, it allows you to add icons, this component has name property which decides which icon to display.

Quotes.html-

<ion-content padding>

<h3 text-center>Select your favorite Quote</h3>

<ion-list>

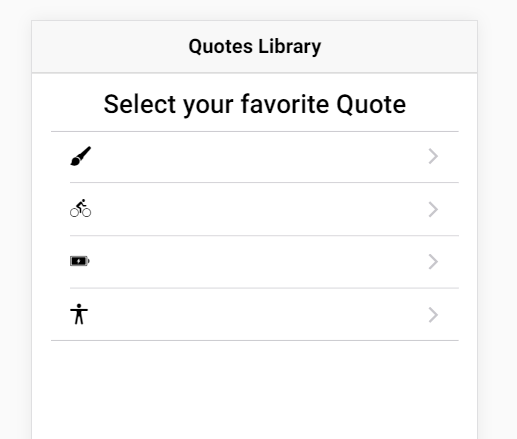
<button ion-item \*ngFor="let quoteGroup of quoteCollection">

<ion-icon [name]="quoteGroup.icon"></ion-icon>

</button>

</ion-list>

</ion-content>



lets add some heading and quote count-

<ion-content padding>

<h3 text-center>Select your favorite Quote</h3>

<ion-list>

<button ion-item \*ngFor="let quoteGroup of quoteCollection">

<ion-icon [name]="quoteGroup.icon"></ion-icon>

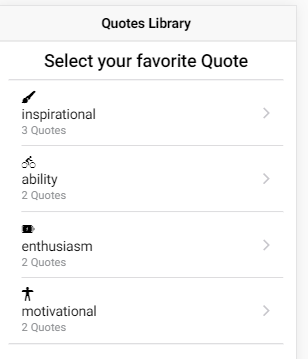
<h2>{{quoteGroup.category}}</h2>

<p>{{quoteGroup.quotes.length}} Quotes</p>

</button>

</ion-list>

</ion-content>



its ok but what i dnt like is that icon sits on top of this heading and also i want to chnage the heading later on. So how can we chnage the placing of icon. The ion-icon takes more than just name property , we can also add property to tell where it should align itself inside of an ion-item. This is important. This is how ionic 2 works in general. A lot of compoents works togather in some ways and you can always learn more about it in official docs for your use case. This use case is pretty common one. You have a icon inside of an ion item. Keep in mind this button is simple a ion item. It is button but with ion-item directive we tell ionic to treat it as a item in my ion-list.

Now since this button as a ion-item directive and icon is used inside of this button, we can add **item-left** directive to ion-item. It will tell icon to allign itself on left if you are placed inside a ion-item element, which is the case. Ofcourse you can also use **item-right.**

**<ion-content padding>**

<h3 text-center>Select your favorite Quote</h3>

<ion-list>

<button ion-item \*ngFor="let quoteGroup of quoteCollection">

<ion-icon [name]="quoteGroup.icon" item-left></ion-icon>

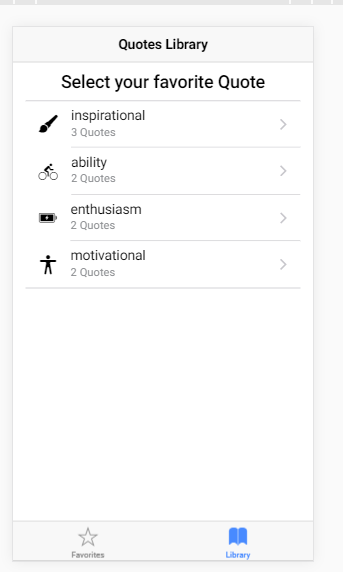
<h2>{{quoteGroup.category}}</h2>

<p>{{quoteGroup.quotes.length}} Quotes</p>

</button>

</ion-list>

</ion-content>

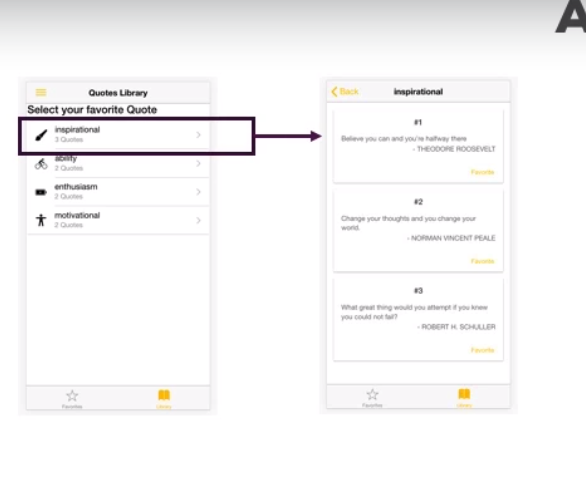
****

now misisng piece is i want to chnage the style of cat title. We want to make it uppercase. We simply do it by using uppercase pipe.

In next lecture i want to make these items clickable and go to quotes by clikeing on quote category.

48)Passing Quotes data between pages

we wnat to see quotes of cat when we click on qoute category. For this we are going to work with quotes page which is responsible for displaying all quotes of a choosen category. We want this-



So after clicking on cat we want to go to quotes page where we have couple of cards (nice boxes with shadow behind)displaying those categoty quotes.

First in in library.html , we pass data when on navigation, we pass all quotes of a library

Library.html-

<ion-list>

<button

ion-item

\*ngFor="let quoteGroup of quoteCollection"

[navPush]="quotesPage"

[navParams]="quoteGroup">

<ion-icon [name]="quoteGroup.icon" item-left></ion-icon>

<h2>{{quoteGroup.category |uppercase}}</h2>

<p>{{quoteGroup.quotes.length}} Quotes</p>

</button>

</ion-list>

On in quotes.ts we need to extract this data. We can do it in constructor, nothing wrong with this approach, but I want to show a new approach, we will do it in ionViewDidLoad, a page life cycle hook. Now this hook will not be executed if page was cached but that;s no problem because if the page was cached so data is already loaded so no need to fetch it again, because we are already displaying the quotes.

Quotes.ts-

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { Quote } from '../../data/quote.interface';

@IonicPage()

@Component({

selector: "page-quotes",

templateUrl: "quotes.html"

})

export class QuotesPage {

quoteGroup: { category: string; quotes: Quote[]; icon: string };

constructor(public navCtrl: NavController, public navParams: NavParams) {}

ionViewDidLoad() {

this.quoteGroup = this.navParams.data;

}

}

Then in hrml file we display category of quote clicked.

Quotes.html-

<ion-header>

<ion-navbar>

<ion-title>{{quoteGroup.category}}</ion-title>

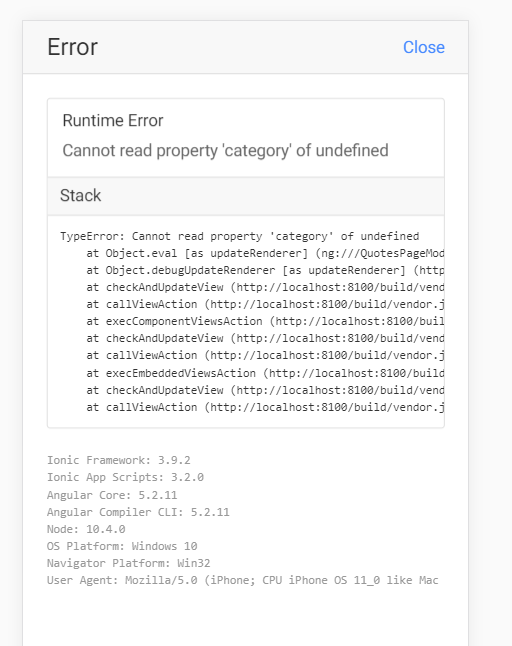
</ion-navbar>

</ion-header>

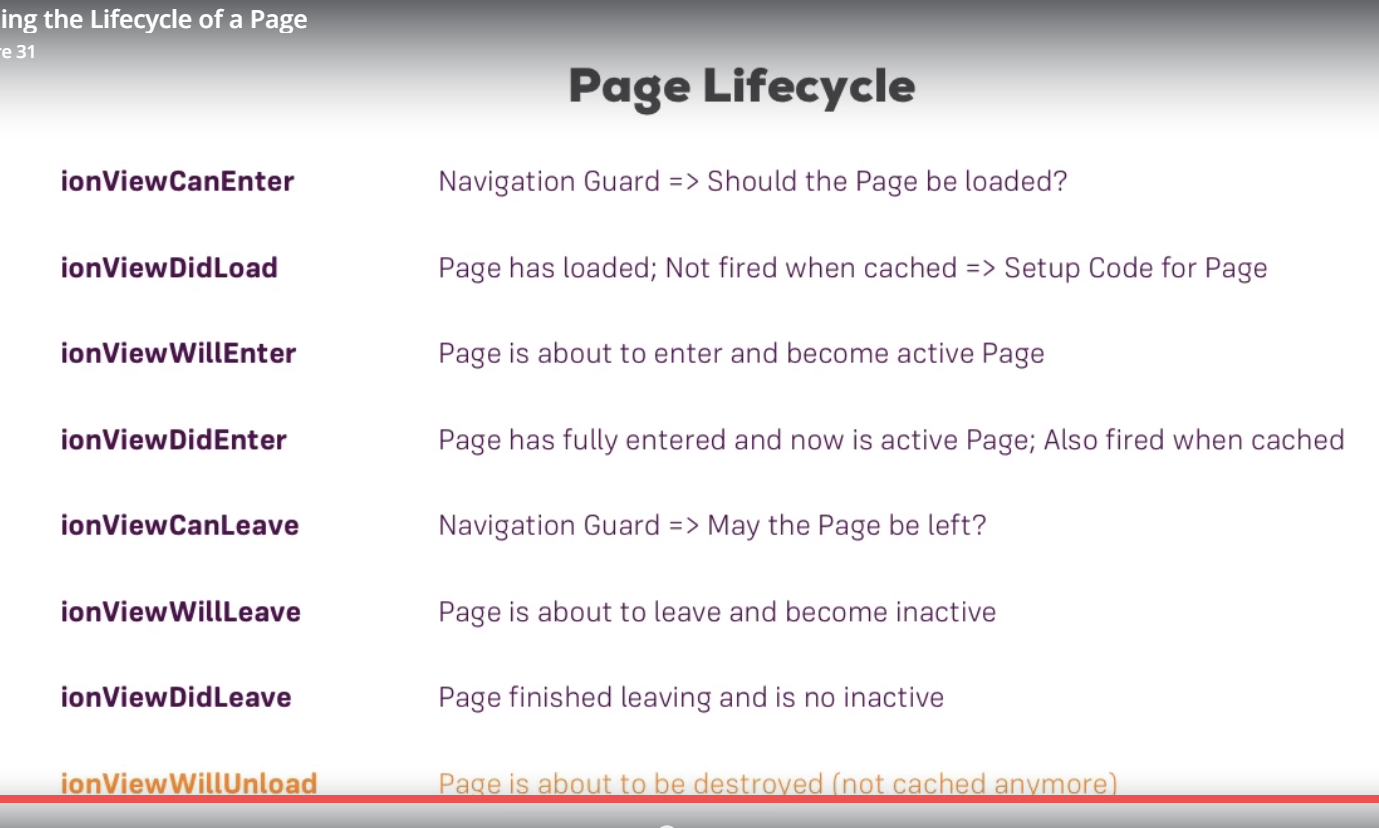
<ion-content padding>

</ion-content>

But when we click on category we get this error-



Why are we getting error. In theory there is nothing worng with fetching params in ionviewDidLoad except for one use case, if we want to display them in our template because templte will actually get created before we reach ionViewDidLoad, becaue template creation is handled by angular and thst is done before ionic triggers this hook.. Following is sequence of life cycle methods-



One solution of this problem would be to use ? in template

Quotes.html-

<ion-navbar>

<ion-title>{{quoteGroup?.category | uppercase}}</ion-title>

</ion-navbar>

This tells angular only try to access category if quote group is avalaible.

Now this works correctly because initially when angular first creates and renders template , it will simply skip that step(to display category), because it knows okay, its not avalaible yet and once it becomes avalaible because we did reach ionViewDidLoad, it will correctly rerender the DOM. This “?” is called elvis operator. but I will comment out this approach, we will get data in ngOnit.

Quotes.ts-

import { Component, OnInit } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { Quote } from '../../data/quote.interface';

@IonicPage()

@Component({

selector: "page-quotes",

templateUrl: "quotes.html"

})

export class QuotesPage implements OnInit {

quoteGroup: { category: string; quotes: Quote[]; icon: string };

constructor(public navCtrl: NavController, public navParams: NavParams) {}

// ionViewDidLoad() {

// this.quoteGroup = this.navParams.data;

// }

ngOnInit() {

this.quoteGroup = this.navParams.data;

}

}

Quotes.html-

<ion-header>

<ion-navbar>

<ion-title>{{quoteGroup.category | uppercase}}</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding>

</ion-content>

We did this experiment to know in which order hooks are reached and that you **ionViewDidLoad** nothing wrong with that but that it is executed after first time the template was rendered.

Now lets work on template of quotes page, we will use some new compoennts.

49)Using ionic cards components

I want to display quotes in cards I,e boxes with shadow. There is u=ionic compoents for it, cards. Go to docs-

Docs->compoennts->cards

Here you can see what are different types of cards you can use.

Lets use it

Quotes.html-

<ion-content padding>

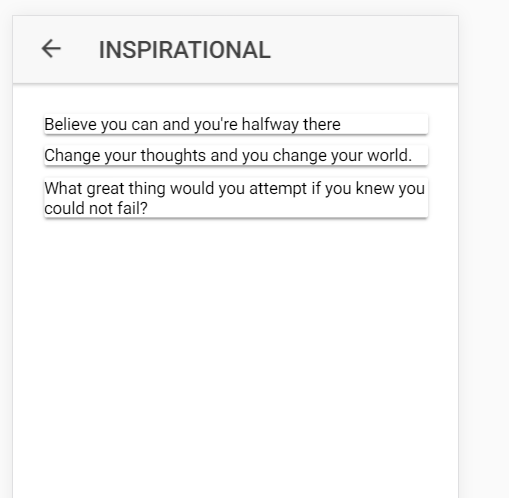
<ion-card \*ngFor="let quote of quoteGroup.quotes">

{{quote.text}}

</ion-card>

</ion-content>

That is what I meant earier you do not need to have wrap this(ion-card) if you do not want to have this ion list look with beginning and ending separator. You can use \*ngFor independent of ion list as we do here. We do see our cards, but they are super pretty.



There is better component to wrap your content in this, is ion-card-content. This is especially made to look your content nicer on the card. It gives some padding to our quote. Html-

<ion-content padding>

<ion-card \*ngFor="let quote of quoteGroup.quotes">

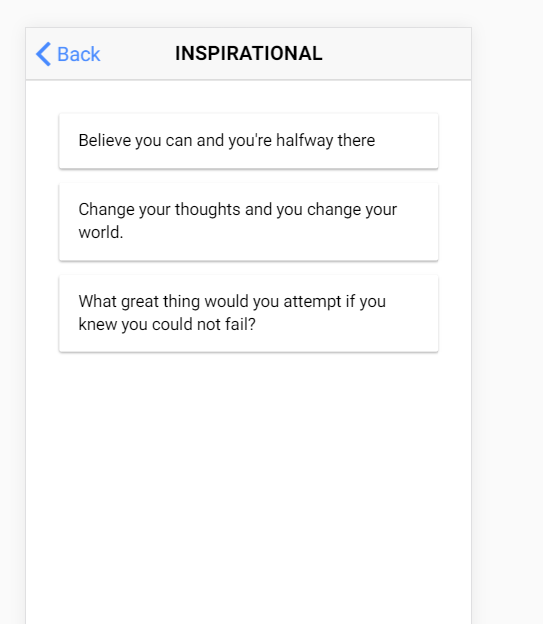
<ion-card-content>

{{quote.text}}

</ion-card-content>

</ion-card>

</ion-content>



Now we want to have header and favourite button on each card. There is another component called **ion-card-header.** so these 2 components are sub compoents components that you can use inside of ion-card.

Actually a smentioned before you can have a look at component docs for example with images, you can see that you can place some other special elements like image elements(<i>) additional to card content, so you can also us that in a card. Again have a look at this documentation to learn about your specific usecase. You can see ion list will aslo be another alternative(in docs, se e list in cards). So tere are 2 much options to go through them all.

But theory is same , you have your wrapping card element and then there you can use coule of building blocks, and you can play around with that and can try normal html elemnts like h1 , to see if it fits your use case. Always keep in mind that you are cerating normal html code with the help of angular 2 components.

In next lecture we will dive deeper into improving our cards and see which tools ionic 2 gives us to provide structure for our content here.

Quotes.html-

<ion-content padding>

<ion-card \*ngFor="let quote of quoteGroup.quotes; let i =index">

<ion-card-header>

#{{i+1}}

</ion-card-header>

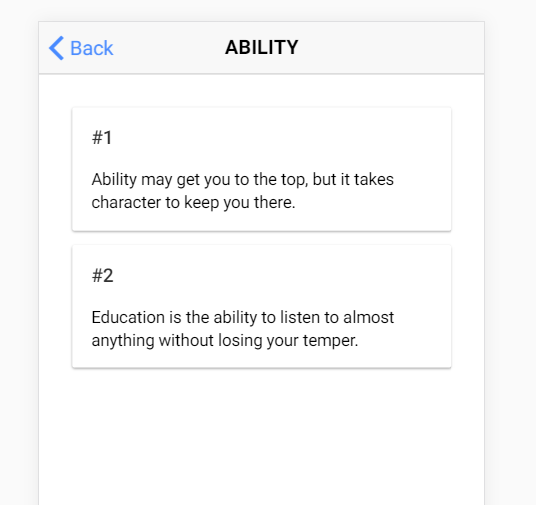
<ion-card-content>

{{quote.text}}

</ion-card-content>

</ion-card>

</ion-content>



50)Using ionic grid, Buttons and styling Cards

First we add a p tag to add autor. But problem is author does not stand out. I will come back to it later. Now I want to have buttons for mrking a quote as favourite. I want to have this button at bottm of card. There is couple of ways to do that. I am going to show you how to do this with **ionic-grid**. Ioic 2 shipd with ots own grids and you can learn it in docs.

Docs -> compoent ->grid

It’s a grid as you might know it from other css frameworks, where you this have grid wrapper, then you can place rows inside of it, your rows have columns and you can define width of columns. If you do not define the width they will automatically split up the valaibe content, since this gird uses flexbox a css feature. You can use this grid anywhere in ionic template, I am going to use it in ion-card.

So we use ion-row after card-content. This means this is my content and now I am adding row for buttons and so on. You can wrap ion-row into ion-grid but it will create additional spacing which I do not want to have. Then we have ion-col inside row, I am not defing width because I want it to starch to full screen. I can provide wisth like this-

<ion-row>

<ion-col width-50></ion-col>

</ion-row>

</ion-card>

This column will take up half the space. However I want to allgn any text inside of it to right. So we use **text-right**  directive. I showed you where you can find more about these helper directive in the end of last section.

Then use button in column and use ion-button directive to make it look nice.

Quotes.html-

<ion-content padding>

<ion-card \*ngFor="let quote of quoteGroup.quotes; let i =index">

<ion-card-header>

#{{i+1}}

</ion-card-header>

<ion-card-content>

<p>{{quote.text}}</p>

<p>{{quote.author}}</p>

</ion-card-content>

<ion-row>

<ion-col text-right>

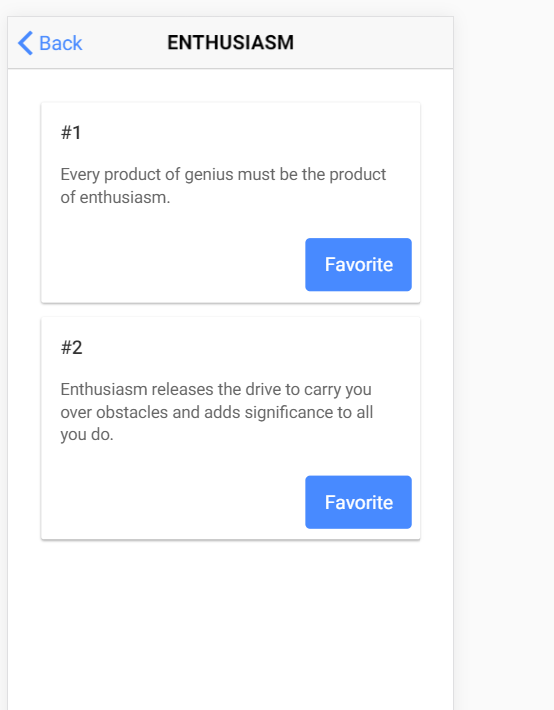
<button ion-button>Favorite</button>

</ion-col>

</ion-row>

</ion-card>

</ion-content>

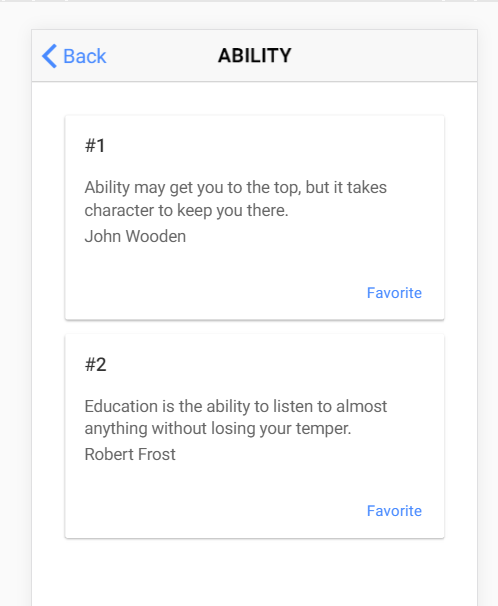


We get this fat favourite button. I dnt like hiw this button stands out. Thank fully we have got different ways of styling buttons. Go to-

Docs-> compoents => buttons

There we can see different ways to style our button. I like clear style. So we will use it. Just add **clear** directive to your button to have this style. This has no border no background. Thenwe small directive to make button looks small, other options are medium and large. the we add a click lister to button.

<button ion-button clear small>Favorite</button>



Later w ewill display either fav or unfav button. So will come back to button later. Now I want to change the style of author text.

51)Adding custom Styles

I want to change style of author text. I want to position it on right and change it’s text color. so I will write my own css for this. We do not have to stick to ionic 2/3 built in compoenents and helper directives, we can always restyle things on our own. In quotes.scss –

page-quotes {

.author {

color: #ccc;

text-align: right;

padding: 0;

margin: 0;

}

}

Now we want to have “-“ in front of author. We can do this normal css by, using before pseudo selector-

page-quotes {

.author {

color: #ccc;

text-align: right;

padding: 0;

margin: 0;

}

.author:before {

content: '-';

}

}

But since we are using scss, therefore saas , we can use shortcut which is used to use & and sign and then pseudo selector. it will be complied to css, so to above syntax.

page-quotes {

.author {

color: #ccc;

text-align: right;

padding: 0;

margin: 0;

&:before {

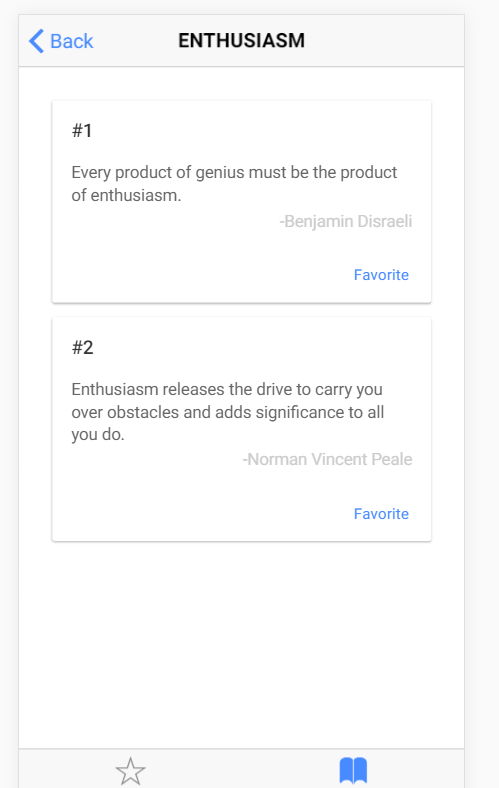
content: '-';

}

}

}

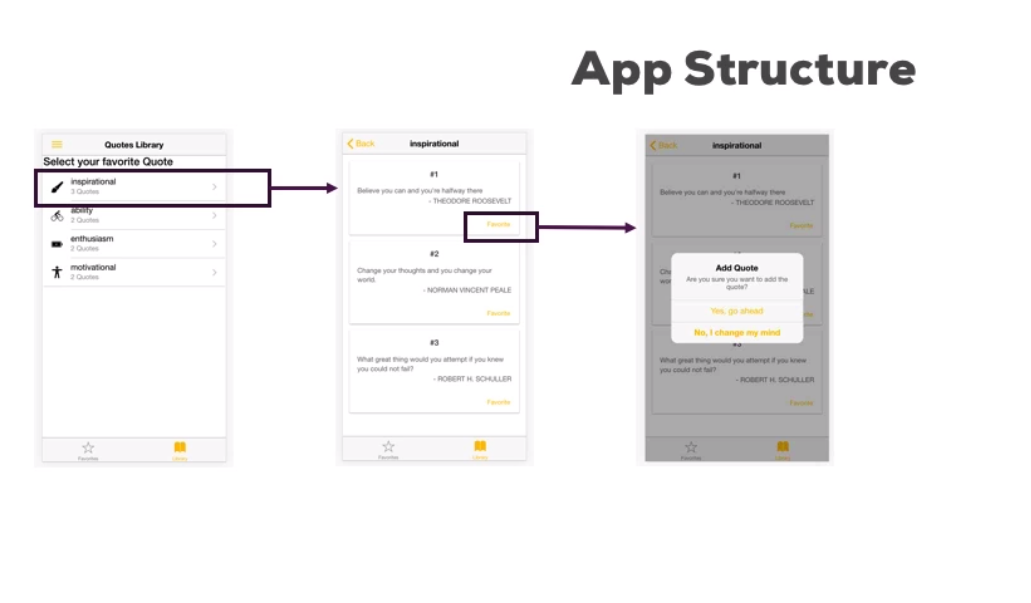
Then we add this author class to p tag, where we display author.



Now lets do something when we click on favourite button of quote.

52)Adding alerts to application

Now we want to display an alert when user clicks on favourite button on quote. In this way we will not accidentally mark a quote as favourite quote.



So in this lecture ww will see how to display such alert. As you knw you can learn more about in components docs. Here you can see all types of alerts and as always you will learn more about this in course when we reuse this feature in couple of apps. Lets create our basic alert.

On clik on fav button we call method. To have alert we need to inject AlertController. Then we call create method on alertController, this allow me to create a new alert. Then we store this alert in some variable or constant,so that we can use it. Create method takes js object as argument and in this object we can configure the alert. Again you can dive into docs to see all the possibilities, though I am going to show them thoughout the course too. Following properties are ones that you are going to use most though.

That might be overcode. you could leave subTitle, place your message in subtitle then leave message.you dnt have to use message and subtitle property togather, you can use one of them or none of them. Then we show the alert by calling present function on alert.

Quotes.ts-

onAddToFavorite(selectedQuote: Quote) {

console.log('methd called');

const alert = this.alertCtrl.create({

title: 'Add Quote',

subTitle: 'Are you sure?',

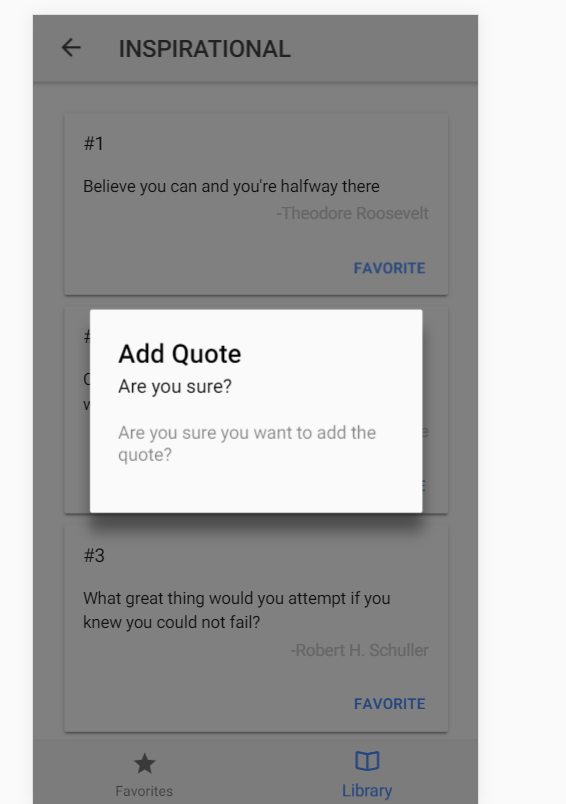
message: 'Are you sure you want to add the quote?'

});

alert.present();

}

This is what we get -



But we want to have buttons. W ecan add buttons by adding buttons property to js object that we pass to create method-

const alert = this.alertCtrl.create({

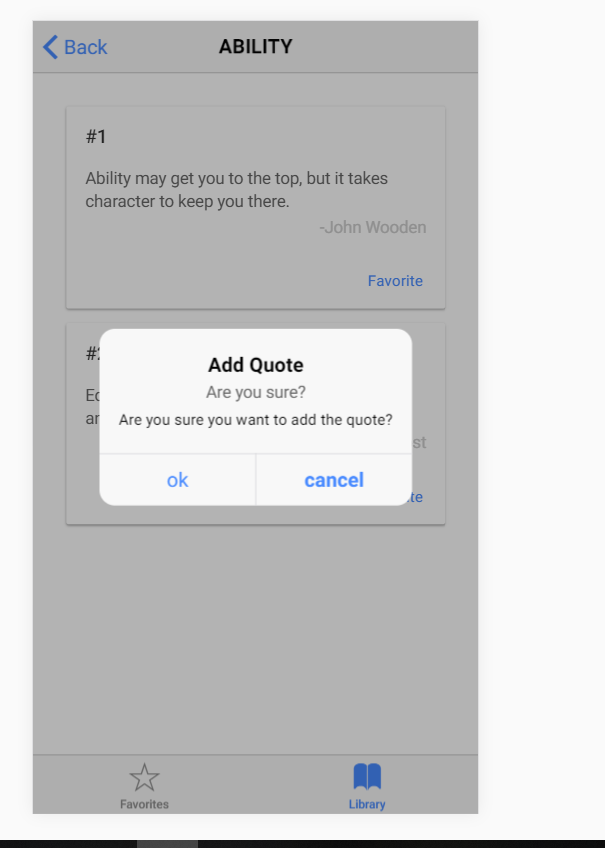
title: 'Add Quote',

subTitle: 'Are you sure?',

message: 'Are you sure you want to add the quote?',

buttons: ['ok', 'cancel']

});



If we click on any button then alert will go away.

To create more complex buttons, you dnt have to pass a string in this array, you can pass whole js object. this object has some proeprties.

Quotes.ts-

onAddToFavorite(selectedQuote: Quote) {

console.log('methd called');

const alert = this.alertCtrl.create({

title: 'Add Quote',

subTitle: 'Are you sure?',

message: 'Are you sure you want to add the quote?',

buttons: [

{

text: 'Yes,go ahead',

handler: () => {

console.log('ok');

}

},

{

text: 'No,I chnaged my mind',

handler: () => {

console.log('Cancelled!');

}

}]

});

alert.present();

}

Here text is test that displayed on button and handler is function that will be called after button is clicked. Now you can assign role to buttons by adding role property, again if you want to learn which properties you can use, go to components docs -> alerts. You can even create buttons inside of alert. Also look at api docs of AlertController which are bit more specific about all supported options on alerts.

Lets add rle to second button.What roles does is it tells ionic 2 that this button is there to dismiss the alert and not to do anything else. You can either have no role or cancel role. This simply means, well dismiss it. That;s why first button have no role, because here we want to do more than just dismiss it I,e we want to execute handler.

Quotes.ts-

onAddToFavorite(selectedQuote: Quote) {

console.log('methd called');

const alert = this.alertCtrl.create({

title: 'Add Quote',

subTitle: 'Are you sure?',

message: 'Are you sure you want to add the quote?',

buttons: [

{

text: 'Yes,go ahead',

handler: () => {

console.log('ok');

}

},

{

text: 'No,I chnaged my mind',

role: 'cancel',

handler: () => {

console.log('Cancelled!');

}

}]

});

alert.present();

}

If we click on second button, we still see console statement, i.e means handler is still execute. But in background ionic 2 knows that this is actually cancelled button. But could also leave it out. You dnt have to specify the role here. Why would you add roll then? well if you add cancel role here, it simply tells ionic 2 that it should always execute handler for this button whenever this alert is dismissed, so not by just clicking the button which would hve excuted the handler anyways but also if you clik the backdrop. So this is what role does, it tells the ionic if alert is dismissed execute this handler. So this is role does.

Now when we click on yes button we want to mark quote as favourite. So this is related to state management. We want a place in app where we can manage state of a quote to determine if a quote is favourite or not.

53)working with angular 2 services in ionic code

there are couple of ways to do state management in angular but built in one, one is avalaible out of box it to use services. So we create a folder service sin src. This is really great folder struture, to have separate folder for pages, sepaarte folder for services. If we want to have some compoents, (which we will do later) we will create a folder named components. As a side not , if your app becomes complex then you are ofcourse free to chnage the structure and structure by feature instead of structure by function a swe are currently doing it.

Then in this folder we create a service qutes.ts. There we have a class memeber favoriteQuotes which will hold all quotes that are favorite. Then we some function to add or remove from fav quotes. Keep in mind each quote has unique id, this will become when identifying quotes later on.

Quotes.ts(in services folder)-

import { Quote } from "../data/quote.interface";

export class QuotesService {

private favoriteQuotes: Quote[] = [];

addQuoteToFavorite(quote: Quote) {

this.favoriteQuotes.push(quote);

}

removeQuoteFromFavorites(quote: Quote) {

const position = this.favoriteQuotes.findIndex((quoteEl:Quote) => quoteEl.id === quote.id);

this.favoriteQuotes.splice(position,1);

}

getFavoriteQuotes() {

return this.favoriteQuotes.slice();

}

}

at last we have method to returns all fav quotes. We could hve just returned the whole array but we used the slice method because w eneed the copy of that array. Herew e want to pass copy of array because to not be able to directly edit the favoriteQuotes array from outside. So we want it to be edited only by methods in this service. Next step is we need to hook up qoyes page to add quote to ffavoriteQuote array in service.

54)Marking Quotes as favorite by using a Services

here we want to use the service thatw e crated in last lecture. First you need to provide that servive in provider array in app.moddule. There we already provide a error handler ionic 2 ships with. So we will get same instance of service applicatin wide, this is important becaue we want to manage one and the same array of quotes across our application. App.module-

providers: [

StatusBar,

SplashScreen,

{provide: ErrorHandler, useClass: IonicErrorHandler},

QuotesService

]

})

top 2 services are not in max’s code. The we inject this Quotes service in quotes compoent . And whenw e click on **yes , go ahead** button of alert, we call method of quotes service which adds the quote to FavoriteQuotes.

Quotes.ts(component)-

onAddToFavorite(selectedQuote: Quote) {

console.log('methd called');

const alert = this.alertCtrl.create({

title: 'Add Quote',

subTitle: 'Are you sure?',

message: 'Are you sure you want to add the quote?',

buttons: [

{

text: 'Yes,go ahead',

handler: () => {

this.quoteService.addQuoteToFavorite(selectedQuote);

}

},

{

text: 'No,I chnaged my mind',

role: 'cancel',

handler: () => {

console.log('Cancelled!');

}

}]

});

alert.present();

}

55)Preparing the favorite Quotes Page

right now we can add same quote multiple times to our FavQuote array, soon we will have check on that.

Here we will work on fav page. First we iject quotes service, the we need to fetch that fav quotes array. We can do this in constructor, we can do this in ngOnit, or ionViewDidLoad. But gain be ware that this will not be executed if page was cahed, so you will not get latest fav quote in some use cases. So ionViewWillEnter looks pretty good. This will always be executed even if page was cached and will be executed right before its displayed.

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { QuotesService } from '../../services/quotes';

import { Quote } from '../../data/quote.interface';

@IonicPage()

@Component({

selector: 'page-favorites',

templateUrl: 'favorites.html',

})

export class FavoritesPage {

quotes: Quote[];

constructor(public quotesService: QuotesService) {

}

ionViewWillEnter() {

this.quotes = this.quotesService.getFavoriteQuotes();

}

}

now i want to display these quotes in fav page.

56)Diving deeper into list items

we have use ion-list with buttons inside of it, there is nothing worng with it but i wnat to show new approach. We will use ion-items component, which allows us to create anice item like style with which split over mutiple rows and separator in between. d/w to using buttons is it will not show that arrow, which means we can clik on it and go to someother page.

Favorites.html-

<ion-content padding>

<ion-list>

<ion-item \*ngFor="let quote of quotes">

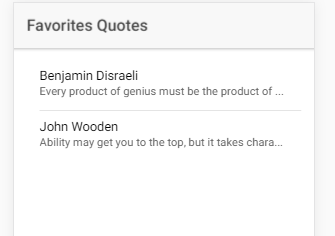
<h2>{{quote.person}}</h2>

<p>{{quote.text}}</p>

</ion-item>

</ion-list>

</ion-content>



dscvd

so you can see that if we select a long quote , it gets cut off here. This is default behaviour of ionic 2, there is only that much and such an ion-item. So its normal. Its not an issue because i want to have these quotes clickable but i also want to make sure that they do not load a new page, instead i want to show a modal, so an overlay over the current page. That why we have to used button, because we do not want that arraow indiacting that on lcik we will go to new page. Now missing thing is modal.

57)Preparing a Modal Page

a modal is overlay over current page and it actually uses a normal ionic 2 page for this. But it is not placing it on stack of pages. We are not loading a model through navController. We are not pushing it on stack on pages, instead again its just an overlay. Again dive into docs to learn more. But lets strat using it in this application.

Quote will serve as our modal page. So first lets work on it.

Here we use ion-cards. Then we add button at bottom using row and column. We use outline directive to have button with border but without any background color. We also assign it another color, you can do this with color directive and this is avalible on any elemnt inlcuding bult in html elements. It allows you to easily assign one of colors defined in variables.scss files in them folder. These are colors-

variable.scss-

$colors: (

primary: #488aff,

secondary: #32db64,

danger: #f53d3d,

light: #f4f4f4,

dark: #222

);

at the end of this section we are going to work with this file and add our own colors. Here we use danger color. Then we add another button after card to dismiss the modal. Then i wanted to center this all, so i add text-center directive to ion-content to have a centered card.

Quote.html-

<ion-header>

<ion-navbar>

<ion-title>Quote</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding text-center>

<ion-card>

<ion-card-content>

TEXT

</ion-card-content>

<ion-row>

<ion-col>

<button ion-button small outline [color]="danger">Unfavorite</button>

</ion-col>

</ion-row>

</ion-card>

<button ion-buttton color="danger">Close</button>

</ion-content>

i next lecture we will display the modal.

58)Presenting a modal

on fav page on clik of item we execute a fucntion. In this fucntion w ewant to diplay modal . So we have to inject ModalContrller. Then we call create function of this controller.

This is common thing we see in ionic 2, we saw it on