**Lec 24**

We can have the page created manually. So for that we need to create a ts file inside the existing pages folder. Let us assume we are taking the page users. We have the folder named as users . Inside the users folder we can again create a user folder .Inside this user folder we can create a user.ts file.

import {Component} from ‘@angular/core’;

@Component({

selector : ‘page-user’,

templateUrl : ‘user.html’

})

export class UserPage{

private name : string;

}

Now, let us create a user.html page.

<ion-header>

<ion-navbar>

<ion-title>{{name}}</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding> // padding to get nice padding along the edges

Hii , I am {{name}} // Here we will use string interpolation

</ion-content>

<ion-footer>

</ion-footer>

One more thing required to register the UserPage inside the app.module.ts

Inside **declarations** and **entryComponents**

Now the challenge in front of us is that ,when we click on any user ie : Pankaj,mohan, we need to send this name to next page that us user.html.

For this , we need to apply some logic , let us see the user link in the users.html

<button ion-button (click)="showMe('mohan')">User ‘Mohan’</button>

Now when we click on the **User ‘Mohan’**it will call the **showMe(‘mohan’)**function inside the users.html.

**showMe(nm : string){**

**this.navCtrl.push(UserPage,{name : nm});**

**}**

Here **navCtrl** is the object reference of NavController class. Also, we know that the navigation in ionic 2 works on the basis og**stack (LIFO).**

So, this.navCtrl.push() will push the page at the top of the stack and thus it will be displayed first.

Note :If we want to pass some value to the next page user.html then we need to pass the second argument to the push function. The second argument can be number , string or an object/array.

this.navCtrl.push(UserPage**,{name : nm}**);

Here **UserPage** is the name of component.

Fine , it is ok upto here. Now let us get the parameters inside the UserPage. For this we need to use the new class **NavParams**

**Now to use the** NavParams, we need to import the class first.

import {NavParams} from ‘ionic-angular’;

After this , we need to pass the NavParams as an dependency injection.

The complete code is as follows :

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

import {NavParams} from 'ionic-angular';

@Component({

selector : 'page-user',

templateUrl : 'user.html'

})

export class UserPage implements OnInit{

private name : string;

constructor(private navPara : NavParams){

}

ngOnInit(){

this.name = this.navPara.data.name;

}}

Here **name** is the data member of the component class. We have initialized the name inside the ngOnInit() method. The template for the user.ts is already defined above area.