Child to parent

1. Parent pass custom event to child and this custom event has the function of a parent.
2. <child-comp (passtochild)=”callchild()fromparent”>

View encapsulation

“ViewEncapsulation.Emulated “ will make browser understand apply css to the particular component only.

Observables can be unsubscribed unlike promise.

Observables-> rxjs->httpclient

This.httpclient.get<modelname>(url);

Promise->httpmodule & observables->httpclientmodule

SPA

Not loading the page again , only loading the section(view/component).

It loads components based on click

1. Link can have data attach route parameters.
2. Link can have acess privileges.

Routing

Import (RouteModuel,Routes)from ‘@angular/routes’;

Create a route array const routes :Routes =[{},{}]

{path:”<linkname>”,component.HomeComponent}

{path:”mobiles/:price/:brand”,component.HomeComponent}// /: means variable with value

{path:”\*\*”,redirectTo:’component.ErrorComponent’}// \*if linkname not exists\*

**Register routes in imports**

RouterModule.forRoot(routes)

**Validation**

Pristine,touch,dirty,untouch -refer

Reactive Forms

For custom validations , complex forms and dynamic form generation.

**Build Reactive forms**

1.Add entry to appmodule “@angular/forms”

**2.Create member variable in component of FormGroups type**

Eg : registerForm:FormGroup

**3.Create formgroup object**

ngOnInIt(){

this.registerForm = new FormGroup({‘fieldname’:new FormControl(null)})

}

UNIT TESTING

Karma uses jasmine framework for unit testing

Install Karma-firefox-launcher -> npm install karma-firefox-launcher –save-dev

Dev dependencies – exclusively for development

Dependencies in package.json for deployment.

If need add browsers to karma.conf then need to install that browser .

The spec files are unit tests for your source files. The convention for Angular applications is to have a .spec.ts file for each .ts file. They are run using the Jasmine javascript test framework through the Karma task runner when you use the 'ng test' command.

Model.spec.ts

Import {emp } from ‘./model’;

Describe(‘Model Suite’,()=>{

beforeEach(()=>{

cons temp=newEmp(data);

)}

It(‘’,()=>expect(salary).toBeGreaterThan(emp.salary);

});

Component life cycle

1.constructor is te first thing when component class object is created

2.ngoninit() method from oninit ingterface – gathers the data

3.ngoncahnges- track changes for @input parameters

4.Ngdocheck- after ngonit and after event fire this fire.

5.ngafterviewinit- after view children id loaded

6.ngondestroy- when component is removed , mostly used with ngif or routing

**INTERNATIONALISATION-**i18n

Triggerint the files