async

Http Interceptor

Reactive forms

Lazy loading

Redux

Unit testing

ViewEncapsulation

Best practices

RxJS

Links

1. Angular Cheat Sheet  
   <https://medium.com/@sakshi.chavan/angular-cheat-sheet-44f314b78f28>
2. Chart and observables  
   <https://stackblitz.com/edit/angular-j4185k?file=app.component.ts>  
   https://www.syncfusion.com/blogs/post/angular-promises-versus-observables.aspx
3. Promises and callback hell
4. How routing works with matching the patch => prefix and full  
   https://stackoverflow.com/questions/42992212/in-angular-what-is-pathmatch-full-and-what-effect-does-it-have
5. You tube channel  
   Programming with mosh

Revision:

1. Angular basic building blocks
   1. Component
   2. Directives
   3. Pipes
   4. Services
   5. Routing
2. Types of Directives
   1. Component => UI + BL [ in-built + custom components ]
   2. Structural => \*ngIf , \*ngFor [ Modify the structure of the webpage]
   3. Attribute => Enhance the behavour of existing HTML elements [ngModel, disabled, value..]
3. Binding Syntax
   1. String interpolation {{}}
   2. Directive binding []
   3. Event binding ()
   4. 2-way binding [()] **Import FormsModule that exposes ngModel**
4. Nested Components
5. Lifecycle : Common for directive and components
   1. Constructor : once
   2. ngOninit : once
   3. ngOnChanges : for every change in @Input primitive property
   4. **ngDoCheck: for every change in @Input property of any type**
   5. ngDestroy : once [ release any resouces, timer, unsubscribe ]
   6. specific lifecycle methods for components:
      1. ngViewChild
      2. ngViewInit
6. Pipes
   1. Built-in : json, uppercase ,lowercase, title, async..
   2. Custom pipe : class implements PipeTransform  
      transform(data, args){}
7. Directives:
   1. Built-in
   2. Custom
      1. ElementRef
      2. Renderer2
      3. @hostListener
      4. @HostBinding
8. Template Driven form
   1. **Import FormsModule**
   2. ngForm
   3. ngModel => name directive is mandatory
   4. ngModelGroup
   5. Various properties via template local reference varaibale initialized to ngModel  
      #title =”ngModel”  
      title.touched..
9. Services
   1. They are single instance per application unless added in providers:[] of a specific component
   2. REST API
      1. **Import HttpClientModule**
      2. HttpClient => get, post, put , delete…
      3. Default return type is Observables to which you need to subscribe
10. Routing
    1. Either enable routing while creating project using ng new or add Routes explicitly
    2. RouterModule
    3. Routes : [ {path:’’, component:}]
    4. Path match : prefix / full
    5. routerLink to add in HTML template in place of href
    6. routerLinkActive
    7. Router => navigate() to route programmatically
    8. route:ActivatedRoute
       1. Current route
       2. Path parameters =>   
          navigate([‘’, pathparam])  
          route.params.subscribe
       3. Parent path parameters =>   
          localhost:4200/parented/childpath  
          route.parent.params.subscribe
       4. Query parameter [?] =>   
          localhost:4200/parented?id=  
          navigate([‘’], {queryParams:{}, relativeTo:})  
          route.parent.queryParams.subscribe
    9. Guards:
       1. CanActivate : Authorization
       2. Candeactivate : for reroute from a url