**or create a new repository on the command line**

echo "# DatingApp\_ByBiniyam" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Amenbiniyam/DatingApp\_ByBiniyam.git

git push -u origin main

**Angular 12 Note on Training**

**Section 4: Authentication Basics**

* **How to store pass in DB**
* **Using inheritance in C# DRY**
* **Using the C# debugger**
* **Using data transfer objects (DTOs)**
* **Validation**
* **JSON Web tokens (JWTs)**
* **Using services in C#**
* **Middleware**
* **Extension method DRY**

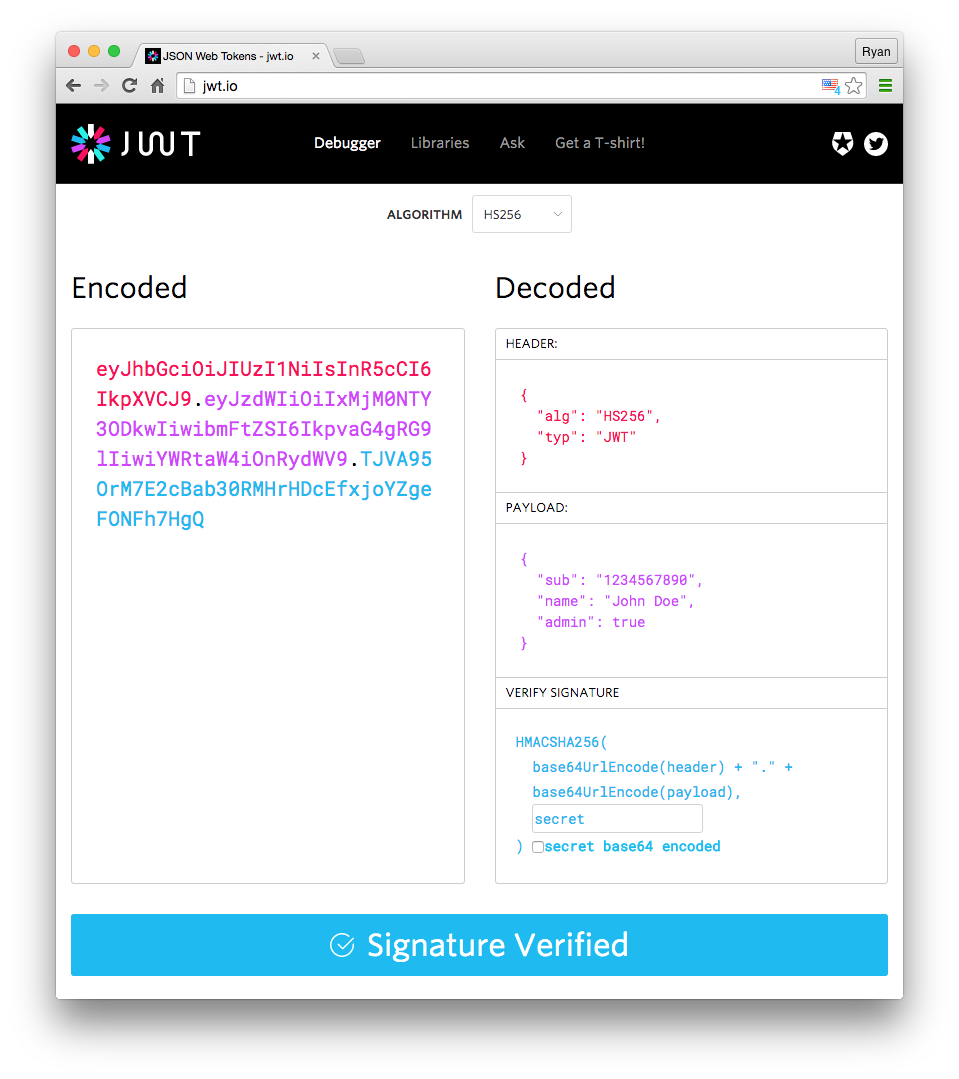
**Requirements**

**How to store pass in DB**

**We have to Hashing and Salting password method to store in db**

**JWT (JSON Web Token)**

**Format**



**Token Authentication**

**Client ---------send username & pass--------------------------------🡪 Server**

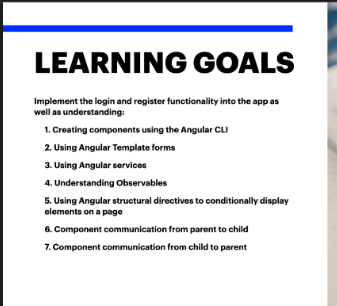
**< -------------validate credin and return JWT----------------------**

* **Client store JWT in web browser**
* **And in every request jwt also sent to server**
* **Server validate user without checking the database with that jwt**

**Advantage of JWT -----**

* **no session is require**
* **single token can be using with multiple backends**
* **mobile friendly**
* **performance**
* **no cookies require**

**Section 5**

****

* **Creating Component of CLI**

**Component creating without test file**

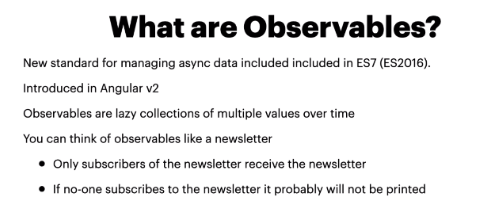
**#ng g c nav – skip-tests**

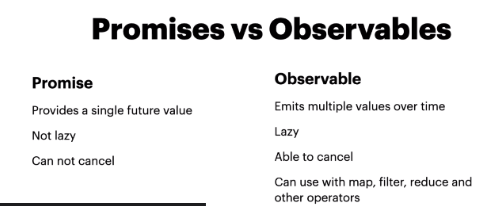
* **Angular provide two way binding that means binding data or properties on our component and display on our template that is called Two way Data Bindng**

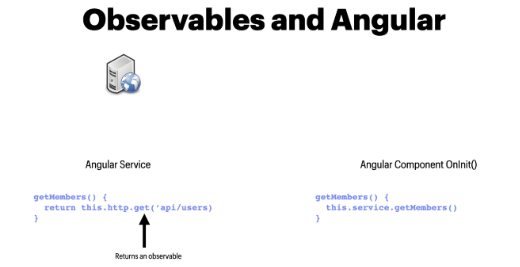
**To make a form an Angular form** <form #loginForm="ngForm" class="d-flex" (ngSubmit)="login()" autocomplete="off">

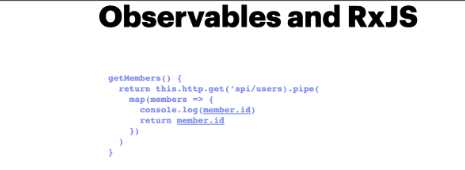
* **To create services #ng g s account --skip-tests**

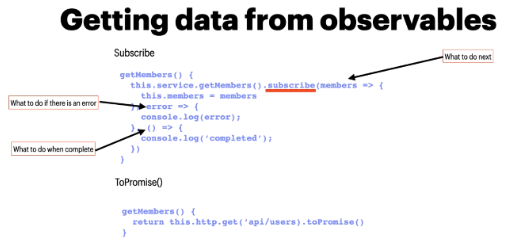
**Observable**

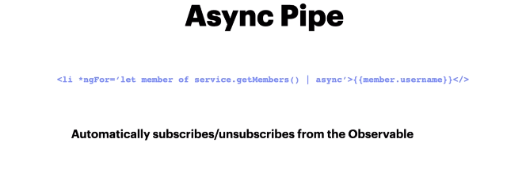
****

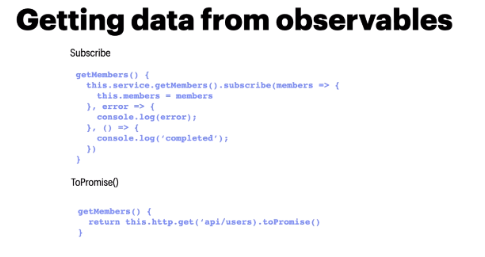
****

****

****

****

****

****

**Parent to Child and Child to parent Communication**

 <app-register [usersFromHomeComponent]="users"  (cancelRegister)="cancelRegisterMode($event)" ></app-register>

[usersFromHomeComponent]="users --- parent to child

(cancelRegister)="cancelRegisterMode($event)" ---- child to parent

**Section 6**

**  
1. Adding Angular Routings**

**2. Adding links to nav bar link based on the routing we have given**

**To make selected link active in nav bar use** routerLinkActive="active"

**Routing or redirecting in component level use this code**       this.router.navigateByUrl('/members');

**Adding Toaster in Angular use this command**

**PS D:\DatingApp\client> npm install ngx-toastr**

**Generating guards for our component using**

**PS D:\DatingApp\client\src\app\\_guards> ng g guard auth --skip-tests**

**Using guards for multiple links**

const routes: Routes = [

  {path: '', component: HomeComponent},

  {path: '',

   runGuardsAndResolvers: 'always',

   canActivate: [AuthGuard],

   children: [

    {path: 'members', component: MemberListComponent, canActivate: [AuthGuard]},

    {path: 'members/:id', component: MemberDetailComponent},

    {path: 'lists', component: ListComponent},

    {path: 'messages', component: MessagesComponent},

   ]

  },

  {path: '\*\*', component: HomeComponent, pathMatch: 'full'},

];

**Adding Bootswatch**

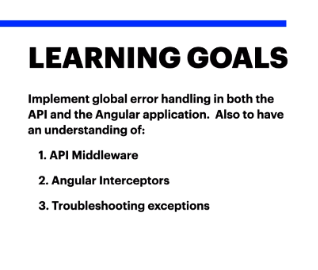
**PS D:\DatingApp\client> npm install bootswatch**

**Creating Shared Modules**

**PS D:\DatingApp\client\src\app\\_modules> ng g m shared --flat**

**Section 7**

**Error Handling**

****

**Creating Interceptors**