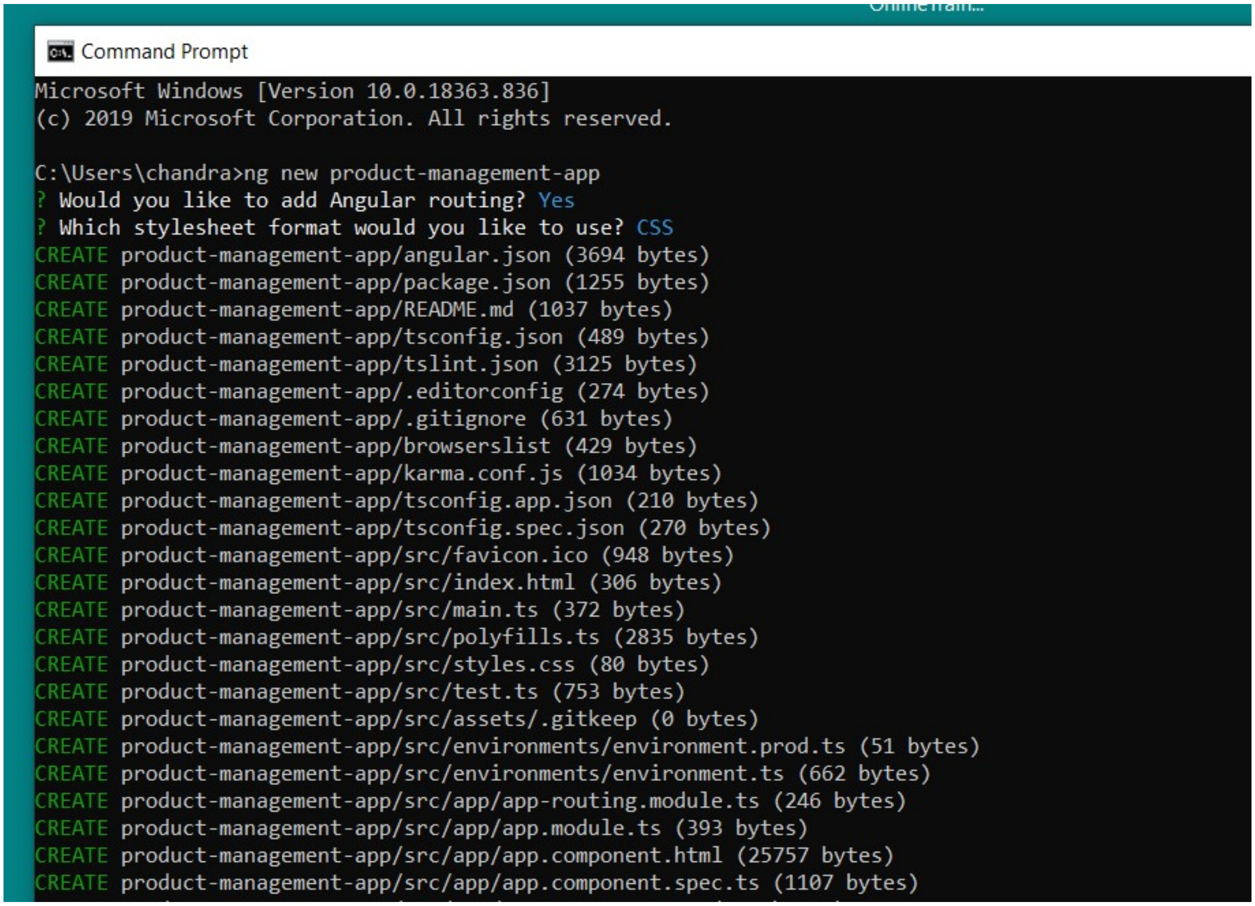
Angular CURD Operations using Array

step1:create a angular project by using following command



```
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\chandra>ng new product-management-app
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE product-management-app/angular.json (3694 bytes)
CREATE product-management-app/package.json (1255 bytes)
CREATE product-management-app/README.md (1037 bytes)
CREATE product-management-app/tsconfig.json (489 bytes)
CREATE product-management-app/tslint.json (3125 bytes)
CREATE product-management-app/.editorconfig (274 bytes)
CREATE product-management-app/.gitignore (631 bytes)
CREATE product-management-app/browserslist (429 bytes)
CREATE product-management-app/karma.conf.js (1034 bytes)
CREATE product-management-app/tsconfig.app.json (210 bytes)
CREATE product-management-app/tsconfig.spec.json (270 bytes)
CREATE product-management-app/src/favicon.ico (948 bytes)
CREATE product-management-app/src/index.html (306 bytes)
CREATE product-management-app/src/main.ts (372 bytes)
CREATE product-management-app/src/polyfills.ts (2835 bytes)
CREATE product-management-app/src/styles.css (80 bytes)
CREATE product-management-app/src/test.ts (753 bytes)
CREATE product-management-app/src/assets/.gitkeep (0 bytes)
CREATE product-management-app/src/environments/environment.prod.ts (51 bytes)
CREATE product-management-app/src/environments/environment.ts (662 bytes)
CREATE product-management-app/src/app/app-routing.module.ts (246 bytes)
CREATE product-management-app/src/app/app.module.ts (393 bytes)
CREATE product-management-app/src/app/app.component.html (25757 bytes)
CREATE product-management-app/src/app/app.component.spec.ts (1107 bytes)
```

step2: open project into Visual Studio Code after add bootstrap code into index.html

->copy css from getbootstrap.com site or else use following one
https://getbootstrap.com/docs/4.5/getting-started/introduction/

Just copy css link and add into index.html head section

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
```

->same kind of way copy js file and into index.html before closing body tag

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>
```

step3:just run and check boot strap working fine or not by using following command.

ng serve –o to see the effect add following code in to app.component.ts

```
<div class="container">
  <div class="row">
      <div class="col-md-8">
           <h3 class="text-center" style="width: 1200px;color:red;margin: 10px 10px 10px -
54px;">{{title}}</h3>
      </div>
      <div class="col-md-4">
          <div class="alert alert-success alert-dismissable" *ngIf="msg">
          <a href="#" class="close" (click)="clickMe()">&times;</a>
         {{msg}}
          </div>
      </div>
  </div>
```

step4: add FormsModule,HttpClientModule(future purpose) in app.modules.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AppComponent } from './app.component';
@NgModule({
  declarations: [
    AppComponent,

  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,

  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

step5: first add product Array into app.component.ts file

```
products = [
    {name:"Rice", qty:"25", price:"1300",},
    {name:"Soap", qty:"25", price:"85",},
    {name:"Oil", qty:"1L", price:"95",}
  ];
```

Step6: To display the array Elements in app.component.html add the following code along with bootstrap

```
<h2>Read data & (Delete data):</h2>
  <table class="table table-bordered">
  <thead>
```

```
    <tr>
      <th width=400>Name</th>
      <th width=400>Quantity</th>
      <th width=400>Price</th>
      <th width=250>Actions</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let product of products; let i=index">
      <td>{{product.name}}</td>
      <td>{{product.qty}}</td>
      <td>{{product.price}}</td>
      <td>
          <a class="btn btn-success" (click)="editProduct(i)">Edit</a>
          <a class="btn btn-danger" (click)="deleteProduct(i)">Delete</a>
      </td>
    </tr>
  </tbody>
</table>
Note:editProduct()/deleteProduct() methods at once we can implement in just app.component.ts
following code coming steps fully implementing

  editProduct(k){

  }
  deleteProduct(i){

  }
```

Step 7: now concentrate to add product element for that add the following code in app.component.html above display array code

```
<div class="clearfix"></div>
  <hr>
  <h2>Create data:</h2>
  <form class="form-horizontal">
  <div class="form-group">
      <label class="control-label col-sm-2" for="name">Name :</label>
      <div class="col-sm-10">
      <input type="text" class="form-control" id="name"
      name="name" [(ngModel)]="model.name" placeholder="Enter your name">
      </div>
  </div>
  <div class="form-group">
      <label class="control-label col-sm-2" for="name">Quantity :</label>
      <div class="col-sm-10">
      <input type="text" class="form-control" id="qty"
      name="qty" [(ngModel)]="model.qty" placeholder="Enter your Quantity">
      </div>
  </div>

  <div class="form-group">
      <label class="control-label col-sm-2" for="position">Price :</label>
      <div class="col-sm-10">
```

```
    <input type="text" class="form-control" id="price"
    name="price" [(ngModel)]="model.price" placeholder="Enter your Product Price">
    </div>
</div>

<div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-outline-secondary" (click)="addProduct()">Save</button>
    </div>
</div>
</form>
```

Step8: now add the app.component.ts file level to implement addProduct() functionality use following code

```
model:any={};
model2:any={};
msg:any="";
addProduct(){
  this.products.push(this.model);
  this.model = {};
  this.msg = "Record is successfully added..... ";
}
```

Step9: to edit product record code add into app.component.html level below array records table

```
<h2>Update data:</h2>
  <form class="form-horizontal">
  <div class="form-group">
      <label class="control-label col-sm-2" for="name">Name:</label>
      <div class="col-sm-10">
      <input type="text" class="form-control" id="name"
      name="name" [(ngModel)]="model2.name" placeholder="Enter your Product Name">
      </div>
  </div>
  <div class="form-group">
      <label class="control-label col-sm-2" for="name">Quantity :</label>
      <div class="col-sm-10">
      <input type="text" class="form-control" id="qty"
      name="qty" [(ngModel)]="model2.qty" placeholder="Enter your Product Quantity">
      </div>
  </div>

  <div class="form-group">
      <label class="control-label col-sm-2" for="position">Price:</label>
      <div class="col-sm-10">
      <input type="text" class="form-control" id="price"
      name="price" [(ngModel)]="model2.price" placeholder="Enter your Product Price">
      </div>
  </div>
  <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-outline-
secondary" (click)="updateProduct()">Update</button>
```

```
      </div>
  </div>
  </form>
```

Step10: Now add app.component.ts level to implement deleteProduct()/updateProduct()

```
  deleteProduct(i){
    this.products.splice(i,1);
    this.msg = "Record is successfully deleted..... ";

  }


myValue;
  editProduct(k){
    this.model2.name = this.products [k].name;
    this.model2.qty = this.products[k].qty;
   this.model2.price = this.products[k].price;

    this.myValue = k;

  }
  updateProduct(){
    let k= this.myValue;
    for(let i=0; i<this.products.length;i++){
      if(i==k){
        this. products [i]= this.model2;
        this.model2 = {};
        this.msg = "Record is successfully updated..... ";
      }
    }
  }
  clickMe(){
    this.msg = "";
  }
```

Step 11: add css look and feel changing purpose in app.component.css file

```
.btn{
    width:90px;
    margin-left: 7px;
}
.container{
    border: 5px lightseagreen solid;
    margin-top: 7px;
    margin-bottom: 7px;
}
.alert{
    margin-top: 15px;
    margin-bottom: 0px;
}
```