

Source Code

Tutorial 7

Mongo Database Connection (app.js)

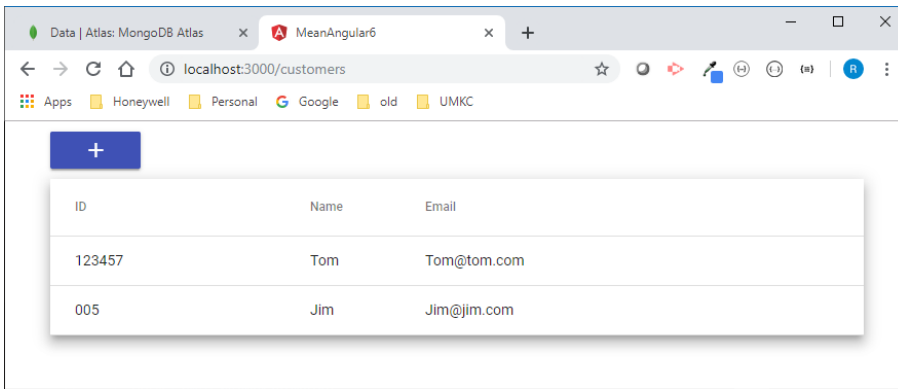
```
app.js
7 var mongoose = require('mongoose');
8 mongoose.connect('mongodb+srv://ase:ase@cluster0-rtt01.mongodb.net/asecustdb?retryWrites=true&w=majority')
9   .then(() => console.log('connection successful'))
10  .catch((err) => console.error(err));
11
```

Mongo Schema (Customer.js)

```
JS Customer.js x
12 var CustomerSchema = new mongoose.Schema({
13   custid: String,
14   name: String,
15   address: String,
16   phone: String,
17   email: String
18 });
```

Home Page (customer.component.html)

```
customer.component.html - ASE-Project-7 - Visual Studio Code
1 <div class="button-row">
2   <a mat-raised-button color="primary" [routerLink]="['/customer-create']">
3     <mat-icon>add</mat-icon>
4   </a>
5 </div>
6 <div class="example-container mat-elevation-z8">
7   <table mat-table #table [dataSource]="dataSource">
8
9     <!-- Note that these columns can be defined in any order.
10      | | The actual rendered columns are set as a property on the row definition -->
11
12     <!-- Title Column -->
13     <ng-container matColumnDef="custid">
14       <th mat-header-cell *matHeaderCellDef> ID</th>
15       <td mat-cell *matCellDef="let element" class="custid-col"> {{element.custid}} </td>
16     </ng-container>
17
18     <!-- Title Column -->
19     <ng-container matColumnDef="name">
20       <th mat-header-cell *matHeaderCellDef> Name</th>
21       <td mat-cell *matCellDef="let element"> {{element.name}} </td>
22     </ng-container>
23
24     <!-- Author Column -->
25     <ng-container matColumnDef="email">
26       <th mat-header-cell *matHeaderCellDef> Email</th>
27       <td mat-cell *matCellDef="let element"> {{element.email}} </td>
28     </ng-container>
29
30     <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
31     <tr mat-row *matRowDef="let row; columns: displayedColumns;" [routerLink]="['/customer-details/', row._id]"></tr>
32   </table>
33 </div>
34
```



Home Page TS file (customer.component.ts)

```
customer.component.ts
1 import {Component, OnInit} from '@angular/core';
2 import {ApiService} from '../api.service';
3 import {DataSource} from '@angular/cdk/collections';
4 import {Observable} from 'rxjs';
5
6 @Component({
7   selector: 'app-customer',
8   templateUrl: './customer.component.html',
9   styleUrls: ['./customer.component.css']
10 })
11 export class CustomerComponent implements OnInit {
12
13   customers: any;
14   displayedColumns = ['custid', 'name', 'email'];
15   dataSource = new CustomerDataSource(this.api);
16
17   constructor(private api: ApiService) {
18   }
19
20   ngOnInit() {
21     this.api.getCustomers()
22       .subscribe(res => {
23         console.log(res);
24         this.customers = res;
25       }, err => {
26         console.log(err);
27       });
28   }
29 }
30
31 export class CustomerDataSource extends DataSource<any> {
32   constructor(private api: ApiService) {
33     super();
34   }
35
36   connect() {
37     return this.api.getCustomers();
38   }
39
40   disconnect() {
41   }
42 }
43
44
```


Create Customer Page (customer-create.component.html)

```
1 <div class="button-row">
2   <a mat-raised-button color="primary" [routerLink]="['/customers']">
3     <mat-icon>list</mat-icon>
4   </a>
5 </div>
6 <form [formGroup]="customerForm" (ngSubmit)="onFormSubmit(customerForm.value)">
7   <mat-form-field class="example-full-width">
8     <input matInput placeholder="CustID" formControlName="custid"
9       [errorStateMatcher]="matcher">
10    <mat-error>
11      <span *ngIf="!customerForm.get('custid').valid && customerForm.get('custid').touched">Please enter Customer ID</span>
12    </mat-error>
13  </mat-form-field>
14  <mat-form-field class="example-full-width">
15    <input matInput placeholder="Name" formControlName="name"
16      [errorStateMatcher]="matcher">
17    <mat-error>
18      <span *ngIf="!customerForm.get('name').valid && customerForm.get('name').touched">Please enter Customer Name</span>
19    </mat-error>
20  </mat-form-field>
21  <mat-form-field class="example-full-width">
22    <input matInput placeholder="Address" formControlName="address"
23      [errorStateMatcher]="matcher">
24    <mat-error>
25      <span *ngIf="!customerForm.get('address').valid && customerForm.get('address').touched">Please enter Customer Address</span>
26    </mat-error>
27  </mat-form-field>
28  <mat-form-field class="example-full-width">
29    <textarea matInput placeholder="Phone" formControlName="phone"
30      [errorStateMatcher]="matcher"></textarea>
31    <mat-error>
32      <span *ngIf="!customerForm.get('phone').valid && customerForm.get('phone').touched">Please enter Customer Phone</span>
33    </mat-error>
34  </mat-form-field>
35  <mat-form-field class="example-full-width">
36    <input matInput placeholder="Email" formControlName="email"
37      [errorStateMatcher]="matcher">
38    <mat-error>
39      <span *ngIf="!customerForm.get('email').valid && customerForm.get('email').touched">Please enter Customer Email</span>
40    </mat-error>
41  </mat-form-field>
42  <div class="button-row">
43    <button type="submit" [disabled]="!customerForm.valid" mat-raised-button color="primary">
44      <mat-icon>save</mat-icon>
45    </button>
46  </div>
47 </form>
48
```

Data | Atlas | x MeanAngula x +

localhost:3000/customer...

Apps Honeywell Personal Google old




CustID

Name

Address

Phone

Email

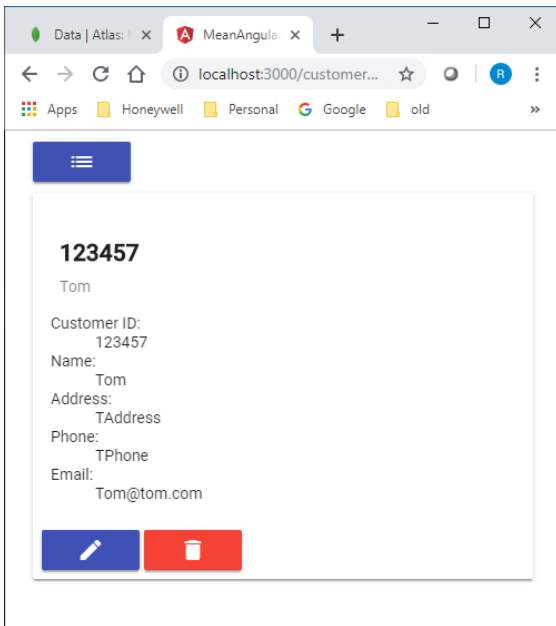


Create Customer TS Page (customer-create.component.ts)

```
TS customer.component.ts TS customer-create.component.ts x
1 import {Component, OnInit} from '@angular/core';
2 import {Router} from '@angular/router';
3 import {ApiService} from '../api.service';
4 import {FormControl, FormGroupDirective, FormBuilder, FormGroup, NgForm, Validators} from '@angular/forms';
5
6 @Component({
7   selector: 'app-customer-create',
8   templateUrl: './customer-create.component.html',
9   styleUrls: ['./customer-create.component.css']
10 })
11 export class CustomerCreateComponent implements OnInit {
12
13   customerForm: FormGroup;
14   custid: string = '';
15   name: string = '';
16   address: string = '';
17   phone: string = '';
18   email: string = '';
19
20   constructor(private router: Router, private api: ApiService, private formBuilder: FormBuilder) {
21   }
22
23   ngOnInit() {
24     this.customerForm = this.formBuilder.group({
25       'custid': [null, Validators.required],
26       'name': [null, Validators.required],
27       'address': [null, Validators.required],
28       'phone': [null, Validators.required],
29       'email': [null, Validators.required]
30     });
31   }
32
33   onFormSubmit(form: NgForm) {
34     this.api.postCustomer(form)
35       .subscribe(res => {
36         let id = res['_id'];
37         this.router.navigate(['/customer-details', id]);
38       }, (err) => {
39         console.log(err);
40       });
41   }
42 }
43
44 master 0 0 0 Go Live Ln 18, Col 22 Spaces: 2 UTF-8 CRLF TypeScript 3.4.3 1
```

Customer Details Page (customer-detail.component.html)

```
TS customer.component.ts customer-detail.component.html x
1 <div class="button-row">
2   <a mat-raised-button color="primary" [routerLink]="['/customers']"><mat-icon>list</mat-icon></a>
3 </div>
4 <mat-card class="example-card">
5   <mat-card-header>
6     <mat-card-title><h2>{{customer.custid}}</h2></mat-card-title>
7     <mat-card-subtitle>{{customer.name}}</mat-card-subtitle>
8   </mat-card-header>
9   <mat-card-content>
10     <dl>
11       <dt>Customer ID:</dt>
12       <dd>{{customer.custid}}</dd>
13       <dt>Name:</dt>
14       <dd>{{customer.name}}</dd>
15       <dt>Address:</dt>
16       <dd>{{customer.address}}</dd>
17       <dt>Phone:</dt>
18       <dd>{{customer.phone}}</dd>
19       <dt>Email:</dt>
20       <dd>{{customer.email}}</dd>
21     </dl>
22   </mat-card-content>
23   <mat-card-actions>
24     <a mat-raised-button color="primary" [routerLink]="['/customer-edit', customer._id]"><mat-icon>edit</mat-icon></a>
25     <a mat-raised-button color="warn" (click)="deleteCustomer(customer._id)"><mat-icon>delete</mat-icon></a>
26   </mat-card-actions>
27 </mat-card>
28
44 master 0 0 0 Go Live Ln 28, Col 1 Spaces: 2 UTF-8 CRLF HTML 1
```



Customer Details TS Page (customer-detail.component.ts)

```
1 import {Component, OnInit} from '@angular/core';
2 import {ActivatedRoute, Router} from '@angular/router';
3 import {ApiService} from '../api.service';
4
5 @Component({
6   selector: 'app-customer-detail',
7   templateUrl: './customer-detail.component.html',
8   styleUrls: ['./customer-detail.component.css']
9 })
10 export class CustomerDetailComponent implements OnInit {
11
12   customer = {};
13
14   constructor(private route: ActivatedRoute, private api: ApiService, private router: Router) {
15   }
16
17   ngOnInit() {
18     this.getCustomerDetails(this.route.snapshot.params['id']);
19   }
20
21   getCustomerDetails(id) {
22     this.api.getCustomer(id)
23       .subscribe(data => {
24         console.log(data);
25         this.customer = data;
26       });
27   }
28
29   deleteCustomer(id) {
30     this.api.deleteCustomer(id)
31       .subscribe(res => {
32         this.router.navigate(['/customers']);
33       }, (err) => {
34         console.log(err);
35       });
36   }
37
38 }
39
40
```


Customer Edit Page (customer-edit.component.html)

```
customer-edit.component.html x
1 <div class="button-row">
2   <a mat-raised-button color="primary" [routerLink]="['/customers']">
3     <mat-icon>list</mat-icon>
4   </a>
5 </div>
6 <form [formGroup]="customerForm" (ngSubmit)="onFormSubmit(customerForm.value)">
7   <mat-form-field class="example-full-width">
8     <input matInput placeholder="CustID" formControlName="custid"
9       [errorStateMatcher]="matcher">
10    <mat-error>
11      <span *ngIf="!customerForm.get('custid').valid && customerForm.get('custid').touched">Please enter Customer ID</span>
12    </mat-error>
13  </mat-form-field>
14  <mat-form-field class="example-full-width">
15    <input matInput placeholder="Name" formControlName="name"
16      [errorStateMatcher]="matcher">
17    <mat-error>
18      <span *ngIf="!customerForm.get('name').valid && customerForm.get('name').touched">Please enter Customer Name</span>
19    </mat-error>
20  </mat-form-field>
21  <mat-form-field class="example-full-width">
22    <input matInput placeholder="Address" formControlName="address"
23      [errorStateMatcher]="matcher">
24    <mat-error>
25      <span *ngIf="!customerForm.get('address').valid && customerForm.get('address').touched">Please enter Customer Address</span>
26    </mat-error>
27  </mat-form-field>
28  <mat-form-field class="example-full-width">
29    <textarea matInput placeholder="Phone" formControlName="phone"
30      [errorStateMatcher]="matcher"></textarea>
31    <mat-error>
32      <span *ngIf="!customerForm.get('phone').valid && customerForm.get('phone').touched">Please enter Customer Phone Number</span>
33    </mat-error>
34  </mat-form-field>
35  <mat-form-field class="example-full-width">
36    <input matInput placeholder="Email" formControlName="email"
37      [errorStateMatcher]="matcher">
38    <mat-error>
39      <span *ngIf="!customerForm.get('email').valid && customerForm.get('email').touched">Please enter Customer Email</span>
40    </mat-error>
41  </mat-form-field>
42  <div class="button-row">
43    <button type="submit" [disabled]="!customerForm.valid" mat-raised-button color="primary">
44      <mat-icon>save</mat-icon>
45    </button>
46  </div>
47 </form>
48
```

Data | Atlas: | x MeanAngula: x

localhost:3000/customer...

Apps Honeywell Personal Google old




CustID
123457

Name
Tom

Address
TAddress

Phone
TPhone

Email
Tom@tom.com



Customer Edit TS Page (customer-edit.component.ts)

```
1 import {Component, OnInit} from '@angular/core';
2 import {Router, ActivatedRoute} from '@angular/router';
3 import {ApiService} from '../api.service';
4 import {FormControl, FormGroupDirective, FormBuilder, FormGroup, NgForm, Validators} from '@angular/forms';
5
6 @Component({
7   selector: 'app-customer-edit',
8   templateUrl: './customer-edit.component.html',
9   styleUrls: ['./customer-edit.component.css']
10 })
11 export class CustomerEditComponent implements OnInit {
12   customer = {};
13   customerForm: FormGroup;
14   custid: string = '';
15   name: string = '';
16   address: string = '';
17   phone: string = '';
18   email: string = '';
19   constructor(private router: Router, private route: ActivatedRoute, private api: ApiService, private formBuilder: FormBuilder) {
20   }
21
22   ngOnInit() {
23     this.customerForm = this.formBuilder.group({
24       'custid': [null, Validators.required],
25       'name': [null, Validators.required],
26       'address': [null, Validators.required],
27       'phone': [null, Validators.required],
28       'email': [null, Validators.required]
29     });
30     this.getCustomer(this.route.snapshot.params['id']);
31   }
32   getCustomerDetails(id) {
33     this.api.getCustomer(id)
34       .subscribe(data => {
35         console.log(data);
36         this.customer = data;
37       });
38   }
39   onFormSubmit(form: NgForm) {
40     let id = this.route.snapshot.params['id'];
41     console.log(form)
42     this.api.updateCustomer(id, form)
43       .subscribe(res => {
44         this.router.navigate(['/customer-details', id]);
45       }, (err) => {
46         console.log(err);
47       });
48   }
49   getCustomer(id) {
50     this.api.getCustomer(id).subscribe(data => {
51       id = data._id;
52       this.customerForm.setValue({
53         custid: data.custid,
54         name: data.name,
55         address: data.address,
56         phone: data.phone,
57         email: data.email
58       });
59     });
60   }
61 }
```

MongoDB Atlas (MLab) Database

The screenshot shows the MongoDB Atlas web interface. The browser tabs include 'Data | Atlas: MongoDB Atlas' and 'MeanAngular6'. The URL is 'cloud.mongodb.com/v2/5d9bf084cf09a2463487b322#metrics/replicaSet/5d9cf445553...'. The interface features a sidebar on the left with navigation options: ATLAS (Clusters, Data Lake BETA), SECURITY (Database Access, Network Access, Advanced), PROJECT (Access Management, Activity Feed, Alerts 0, Integrations, Settings), SERVICES (Charts, Stitch, Triggers), and HELP. The main content area is titled 'asecustdb.asecustomerscollections'. It shows the collection size as 127B, total documents as 1, and indexes total size as 36KB. There are tabs for 'Find', 'Indexes', and 'Aggregation'. A filter bar contains the text 'FILTER {"filter":"example"}' with 'Find' and 'Reset' buttons. Below the filter, it says 'QUERY RESULTS 1-2 OF 2'. Two document snippets are displayed:

```
{
  "_id": ObjectId("5d9eb189437c1733cc4c8e74"),
  "custid": "123457",
  "name": "Tom",
  "address": "TAddress",
  "phone": "TPhone",
  "email": "Tom@tom.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId("5d9f83bc28d04539d4ee0994"),
  "custid": "005",
  "name": "Jim",
  "address": "121212",
  "phone": "777-8888",
  "email": "Jim@jim.com",
  "__v": 0
}
```