

Web Assignment

Doğukan Çelik

20200808071

1 - CustomerService

```
import { HttpClient, HttpResponse, HttpParams } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable, throwError } from 'rxjs';
import { catchError, map } from 'rxjs/operators';
import { Duplicate } from '../common/duplicate';
import { AppError } from '../common/apperror';

@Injectable({
  providedIn: 'root'
})

export class CustomerService {
  private apiCreate = 'http://213.248.166.144:7070/customer/createCustomer';
  private apiGet = 'http://213.248.166.144:7070/customer/getCustomerByTcNoEmail';

  constructor(private http: HttpClient) { }

  createCustomer(customerData: any): Observable<any> {
    return this.http.post<any>(this.apiCreate, customerData).pipe(
      map((response) => {
        if (!response.ok) {
          window.alert("duplicate")
          return throwError(() => new Duplicate()) // Assuming your data is in a
property called 'data'
        } else {
          return response.json()
        }
      })
    ),
    catchError((error: HttpResponse) => {
      if (error.status === 409) {
        return throwError(() => new Duplicate());
        console.error("Customer already exist.")
      } else {
        return throwError(() => new AppError());
      }
    })
  )
}
```

```

    );
}

getCustomerByTcNoEmail(tcNo: string, email: string): Promise<any> {
    let params = new HttpParams();
    if (tcNo) params = params.append('tcNo', tcNo);
    if (email) params = params.append('email', email);

    return this.http.get<any>(this.apiGet, { params }).pipe(
        map((response) => {
            if (response && response.ok) {
                window.alert("selamlar")
                return response.json();
            } else {
                throw new AppError();
            }
        }),
        catchError((error: HttpErrorResponse) => {
            if (error.status === 409) {
                return throwError(() => new Duplicate());
            } else {
                return throwError(() => new AppError());
            }
        })
    ).toPromise();
}
}

```

2 - Component

```

import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { CustomerService } from '../services/customer.service';
import { Duplicate } from '../common/duplicate';
import { AppError } from '../common/apperror';

@Component({
    selector: 'app-create-customer-page',
    templateUrl: './create-customer-page.component.html',
    styleUrls: ['./create-customer-page.component.css']
})
export class CreateCustomerPageComponent {
    customerForm: FormGroup;
    errorMessage: string | null = null;
}

```

```

constructor(
  private formBuilder: FormBuilder,
  private customerService: CustomerService
) {
  this.customerForm = this.formBuilder.group({
    id: ['', Validators.required],
    tcNo: ['', Validators.required],
    passportNo: [''],
    nationality: ['', Validators.required],
    firstName: ['', Validators.required],
    lastName: ['', Validators.required],
    middleName: [''],
    gender: ['', Validators.required],
    streetAddress: ['', Validators.required],
    city: ['', Validators.required],
    country: ['', Validators.required],
    email: ['', [Validators.required, Validators.email]],
    phone: ['', Validators.required],
    notes: ['', Validators.required],
  });
}

async onSubmit(): Promise<void> {
  if (this.customerForm.valid) {
    const tcNo = this.customerForm.value.tcNo;
    const email = this.customerForm.value.email;
    this.errorMessage = null;

    const existingCustomer = await
this.customerService.getCustomerByTcNoEmail(tcNo, email)
      .catch((error) => {
        if (error instanceof AppError) {
          this.errorMessage = 'Customer control failed';
        }
      });

    console.log(existingCustomer)

    if (existingCustomer) {
      this.errorMessage = "Customer already exists";
    } else {
      const customerData = this.customerForm.value;
      const createdCustomer = await
this.customerService.createCustomer(customerData).toPromise()
        .catch((createError) => {
          if (createError instanceof Duplicate) {

```

```
        this.errorMessage = "Customer already exists";
    } else if (createError instanceof AppError) {
        console.error("There is an error ", createError)
    }
    });
    if (createdCustomer) {
        console.log("Customer added")
    }
}
} else {
    this.errorMessage = 'Please fill in all required fields correctly.';
}
}
}
```