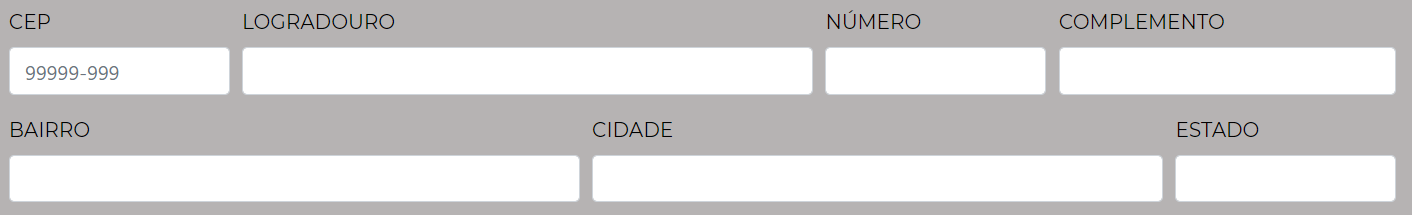
Na estrutura abaixo, você coloca o CEP e tecla TAB , e automaticamente ele atualiza o Logradouro, Bairro, Cidade e Estado. Veja que no meu CEP tem mask=”99999-999”.



Não sei qual padrão você está usando no seu formulário. Eu estou usando o padrão PRIMEND.

Então as definições do meu form é de acordo com o PRIMEND. Se você está usando outro padrão, não tem problema, é só ir adaptando conforme seu código.

Então vamos la!

Codificando o HTML no meu “**formulário.component.html**”.

Codificando o cep, logradouro, numero e complemento.

<div class="form-row">

            <div class="form-group col-md-2">

                <label for="cep">CEP</label>

                <p-inputMask styleClass="form-control" name="cep" placeholder="99999-999"

                    [(ngModel)]="administradorGeral.cep" mask="99999-999" (keydown)="buscarCep(administradorGeral.cep)">

                </p-inputMask>

            </div>

            <div class="form-group col-md-5">

                <label for="logradouro">LOGRADOURO</label>

                <input type="text" class="form-control" name="logradouro" [(ngModel)]="administradorGeral.logradouro">

            </div>

            <div class="form-group col-md-2">

                <label for="numero">NÚMERO</label>

                <input type="text" class="form-control" name="numero" [(ngModel)]="administradorGeral.numero">

            </div>

            <div class="form-group col-md-3">

                <label for="complemento">COMPLEMENTO</label>

                <input name="complemento" type="text" class="form-control" [(ngModel)]="administradorGeral.complemento">

            </div>

       </div>

Na definição do cep, tem a chamada (keydown)”=buscarCep(administradorGeral.cep)”, onde:

**keydown** é quando você muda de campo.

**buscarCEP** é a função.

**administradorGeral** é nome do meu array no arquivo JSON.

**cep** é o campo dentro do meu array JSON.

Codificando o bairro, cidade e estado.

<div class="form-row">

            <div class="form-group col-md-5">

                <label for="bairro">BAIRRO</label>

                <input name="bairro" type="text" class="form-control" [(ngModel)]="administradorGeral.bairro">

            </div>

            <div class="form-group col-md-5">

                <label for="cidade">CIDADE</label>

                <input name="cidade" type="text" class="form-control" [(ngModel)]="administradorGeral.cidade">

            </div>

            <div class="form-group col-md-2">

                <label for="estado">ESTADO</label>

                <input name="estado" type="text" class="form-control" [(ngModel)]="administradorGeral.estado">

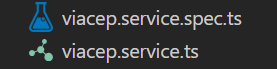
            </div>

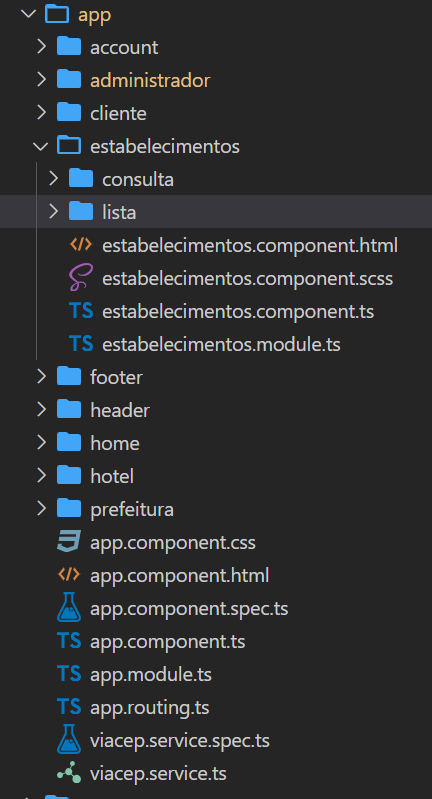
        </div>

Pronto. HTML codificado! Perceba que nos [(ngModel)] estou referenciando, por exemplo,

“administradorGeral.estado”, que é o nome do campo estado no meu array JSON. Igual ao mencionei no campo cep.

Agora é necessário criar os serviços para o viacep. Veja que os serviços viacep.service ficam dentro da pasta “app”.

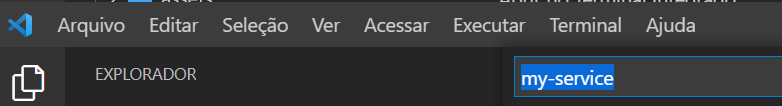




Para fazer isso, voce deve clicar sobre a pasta app, depois clicar com o botão direito do mouse e escolher a opção “Generate Service”.



Ele vai abrir para voce colocar o nome do serviço a ser criado. Voce coloca “**viacep.service**” e tecla ENTER.



Então serão criados as duas pastas acima. O “**viacep.service.spec.ts**” e o “**viacep.service.ts**”. Igual acima.

No “**viacep.service.spec.ts**” voce cola o código abaixo:

import { TestBed, async, inject } from '@angular/core/testing';

import { ViacepService } from './viacep.service';

describe('Service: Viacep', () => {

  beforeEach(() => {

    TestBed.configureTestingModule({

      providers: [ViacepService]

    });

  });

  it('should ...', inject([ViacepService], (service: ViacepService) => {

    expect(service).toBeTruthy();

  }));

});

No “**viacep.service.ts**” voce cola o código abaixo:

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class ViacepService {

  baseUrl = 'https://viacep.com.br/ws/';

  constructor(

    private http: HttpClient,

  ) { }

  public search(cep: string): Observable<any> {

    return this.http.get<any>(`https://viacep.com.br/ws/${cep}/json/`);

  }

  get(cep: string) {

    return new Observable<any>((x) => {

      const request = new XMLHttpRequest();

      request.open('get', `https://viacep.com.br/ws/${cep}/json/`, true);

      request.send();

      request.onload = function() {

        const data = JSON.parse(this.response);

        x.next(data);

      };

    });

  }

}

Não esqueça do CTRL S para salvar os ajustes em todos os lugares que voce alterar.

Agora vamos codificar o “**formulário.component.ts**”.

Na parte dos imports, bem no inicio do arquivo:

import { ViacepService } from '../../viacep.service';

Veja que o meu caminho não é o mesmo que o seu. Então vou colocar embaixo o caminho completo, par você ter o exemplo em relação ao seu caminho.

“c:/DigitalHouse/ProjetoIntegador/PCDEVS1G2/src/app/viacep.service"

Aqui é onde esta as declarações do meu FormularioComponent.

export class FormularioComponent implements OnInit {

Dentro eu declaro, o meu array JSON, onde consta o cep, logradouro, numero, complemento, bairro, cidade e estado.

 administradorGeral: any = {

    id: "",

    razaoSocial: "",

    cnpj: "",

    nomeFantasia: "",

    responsavel:"",

    email: "",

    site: "",

    telefoneFixo: "",

    telefoneCel: "",

    descricao:"",

    categoria:"",

    segmento:"",

    cep: "",

    logradouro: "",

    numero: "",

    complemento: "",

    bairro: "",

    cidade: "",

    estado: "",

    mapa:"",

    referencia:"",

    distancia:"",

    horario:""

}

Abaixo, no meu construtor, declarei a variável ViacepService:

constructor(

    private service: AdministradorService,

    private messageService: MessageService,

    private route: ActivatedRoute,

    private title: Title,

    private router: Router,

    private viacepService: ViacepService

//    private formBuilder: FormBuilder

    ) {

Ao final do ”**formulario.component.ts**”, declarei a função buscarCep:

buscarCep(cep:any){

  let cep1 = cep.replace('\_','');

  if(cep1.length == 9){

    this.viacepService.search(cep.replace('-','')).subscribe(data => {

      this.administradorGeral.logradouro = data.logradouro;

      this.administradorGeral.bairro = data.bairro;

      this.administradorGeral.cidade = data.localidade;

      this.administradorGeral.estado = data.uf;

    })

  }

 }

Neste código, estou retirando a formatação da máscara com traço para buscar os dados relativos ao cep,

que são logradouro, bairro, localidade e uf, ode localidade é a cidade e uf é o estado.

Veja que não coloquei o numero e o complemento, pois depende de cada endereço.

Sugestão:

Tenta colocar os códigos nos seus devidos lugares, salvar e testar.

Se não der certo, me avisa que marco uma meeting para revisarmos na quarta.

Qualquer dúvida, me fala.

Boa sorte!!!