

=====

=====

NODE JS e ANGULA

=====

=====

ERRO CORS

este erro acontece porque o browser bloqueia localhost

para resolver isso eu instalei um plugin do chromes

Pesquisar...

chrome plugin Moesif Origin & CORS Changer

=====

=====

site base

<https://loiane.com/>

<https://valor-software.com/ngx-bootstrap/#/>

usar o Visual Studio Code como editor

baixar o projeto inicial

<https://github.com/angular/quickstart>

abrir o cmd

md pasta (criar uma pasta)

rd pasta (deltar pasta)

rd pasta (delatar e perguntar se deseja)

cls limpa a tela

rd pasta /s /q (deleta a pasta e arquivos)

para ver a versao

```
node -v
```

```
*****
```

```
*****
```

CRIAR UM PROJETO PELO NODE

A melhor pratica para gerar um projeto angular de acordo com a documentação e com o angular-cli.

Para instalar o angular cli globalmente

```
npm install -g @angular/cli
```

depois de gerar o projeto

```
ng update @angular/cli
```

```
ng update @angular/core
```

```
ng update @angular/material
```

```
ng update rxjs
```

Depois para gerar o projeto:

```
ng new NOME_DO_PROJETO
```

Para gerar um componente por exemplo

```
ng generate component meu-component
```

```
ng g c nomecomponente
```

Também pode se usar somente g para ficar mais curto veja o

exemplo de como gerar um serviço:

CRIAR UM SERVICE

ng g service meu-service

CRIAR UM MODULE

ng g m curso

entrar na pasta onde o module vai ficar

O que você deseja criar Comando usado

Componente (Component)

ng g component nome-do-componente

Diretiva (Directive)

ng g directive nome-da-diretiva

Filtro (Pipe)

ng g pipe nome-do-filtro

Serviço (Service)

ng g service nome-do-servico

Módulo (Module)

ng g module nome-do-modulo

+++++

INSTALAR PACOTES

+++++

INSTALAR EXPRESS

```
npm install express -save
```

INSTALAR BOOTSTRAP

```
npm install bootstrap --save
```

no arquivo styles.css

```
@import '~bootstrap/dist/css/bootstrap.min.css';
```

desta forma todas as paginas vao ter o bootstrap

INSTALAR ngx-bootstrap

<https://valor-software.com/ngx-bootstrap/#/>

<https://valor-software.com/ngx-bootstrap/#/documentation#getting-started>

exemplos de componentes do bootstrap

<https://getbootstrap.com/docs/4.3/components/alerts/>

```
npm install ngx-bootstrap --save
```

```
npm install bootstrap ngx-bootstrap --save
```

Adding the required Bootstrap modules in app.module.ts

```
+++++++
```

```
import { NgModule } from '@angular/core';
```

```
import { CommonModule } from '@angular/common';
```

```
import { BsDropdownModule } from 'ngx-bootstrap/dropdown';
```

```
import { TooltipModule } from 'ngx-bootstrap/tooltip';
```

```
import { ModalModule } from 'ngx-bootstrap/modal';
```

```
@NgModule({
```

```
  imports: [
```

```

CommonModule,
BsDropdownModule.forRoot(),
TooltipModule.forRoot(),
ModalModule.forRoot()
],
exports: [BsDropdownModule, TooltipModule, ModalModule]
})
export class AppBootstrapModule {}
+++++++

```

```

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

```

```

<link href="../node_modules/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="../node_modules/bootstrap/dist/js/jquery.js"></script>
<script src="../node_modules/bootstrap/dist/js/bootstrap.min.js"></script>

```

Install Bootstrap 4

```
npm install --save bootstrap@next
```

Bootstrap 4 library added into the folder node_modules/bootstrap.

Import bootstrap.css file into the project by adding the below line in style.css file.

Bootstrap 4 library added into the folder node_modules/bootstrap.

```
@import "~bootstrap/dist/css/bootstrap.css";
```

```
npm install --save @ng-bootstrap/ng-bootstrap
```

After installation, NgbModule must be imported in app.module.js.

```
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
```

INSTALAR JQUERY

```
npm install jquery --save
```

INSTALAR TYPESCRIPT

```
npm install -g typescript
```

PARA TESTAR TYPESCRIPT

<https://babeljs.io/>

<http://es6-features.org/#Constants>

START

```
npm start
```

+++++

+++++

INSTALAR EXPRESS

```
npm install express -save
```

INSTALAR BOOTSTRAP

```
npm install bootstrap --save
```

INSTALAR JQUERY

```
npm install jquery --save
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

```
<link href="../node_modules/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="../node_modules/bootstrap/dist/js/jquery.js"></script>
<script src="../node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
```

entrar na pasta

f:

```
cd c_teste
```

```
md nodeangula
```

```
cd nodeangula
```

```
npm install -g typescript
```

instalar o angular

```
npm install -g angular-cli
```

ver a versao node

node -v

baixar o projeto inicial

<https://github.com/angular/quickstart>

crie um diretorio e copie e descompact pronto

agora volte ao cmd acesse este diretorio

agora digite

npm install

enter e aguarde esta configurando

para verificar

npm start

para abrir o hello angular

<http://localhost:3000/>

Ctrl + C = parar a execução

Installing Angular CLI

npm install -g @angular/cli

Create Your First Angular App

ng new newapp

Updating to Angular 7

ng update @angular/cli @angular/core

Running your First App

cd newapp

ng server

Installing Bootstrap 4

```
npm install bootstrap jquery popper
```

After installation, you should see a bootstrap subfolder in the node_modules folder.. Add bootstrap to the angular.json file like so:

// scroll down to the style section and add bootstrap line below

```
"styles": [  
  "src/styles.scss",  
  "node_modules/bootstrap/dist/css/bootstrap.min.css"  
],
```

NAVEGANDO

```
<nav>
```

```
  <a routerLink="/crisis-center" routerLinkActive="active">Crisis Center</a>
```

```
  <a routerLink="/heroes" routerLinkActive="active">Heroes</a>
```

```
</nav>
```

```
<router-outlet></router-outlet>
```

COMANDOS BASICOS

Comando para instalar

```
npm install -g @angular/cli
```

Comando para criar um projeto

```
ng new NOME-DO-PROJETO
```

Comando para testar o app localmente

```
ng serve
```

Comando para criar componentes

```
ng g component NOME-DO-MEU-COMPONENTE
```

Comando para criar uma classe

```
ng g class NOME-DA-MINHA-CLASSE
```

Comando para criar um service

```
ng g service NOME-DO-MEU-SERVIÇO
```

Comando para gerar a aplicação para produção/desenvolvimento

Build em produção

```
ng build --prod
```

Build em desenvolvimento

```
ng build --dev
```

FIM COMANDOS BASICOS

NOVO PROJETO

ng new NomeProjeto

Adicionando a classe Tarefa ao projeto

ng generate class Tarefa

Criar um novo componente

ng generate component meu-component

#ng g component my_new_compoente

ng g c my_new_componente

criar um serviço

ng g s cursos

acessar antes a pasta onde vai ficar o servico

serviço é usado pra trafegar dados ou criar

class para execução de metodos

SNIPPETS

sao atalhos digita o começo

e dar um tab ai o sistema traz um exemplo

INSTALAR OS PLUGUINHOS

```
*****
```

```
*****
```

DATA BIDDING

```
*****
```

```
*****
```

POPUP

```
*****
```

```
*****
```

Variável de Ambiente

localizar a pasta environments

abrir o arquivo environment.ts

```
LocalHostAPI: 'http://localhost:54199/api'
```

abrir o arquivo envioment.prod.ts

```
LocalHostAPI: '/'
```

abrir o arquivo de servico por exemplo

```
import { environment } from './../environments/environment';  
private readonly API = environment.LocalHostAPI+'APIPESSOA/GetAll';  
private readonly API = `${environment.LocalHostAPI}/APIPESSOA/GetAll`;
```

ERROR - ERRO

* _

npm install @angular-devkit/build-angular

ROTA INTERNA LINK NO CODIGO

```
constructor(private router:Router){
```

```
this.router.navigate(["home"]);
```

Get dados da tela

HTMLInputElement

html :

```
<input type="text" (keyup)="onKeyUp($event) />
```

```
<input #campinput type="text" (keyup.enter)="salvarvalor(campinput.value)
```

```
<p>valor salvo: {{ valorsalvo }}</p>
```

no component

```
salvarvalor(valor){  
  this.valorsalvo = valor;
```

```
}
```

```
onKeyup(evento: KeyboardEvent){  
  console.log((<HTMLInputElement>evento.target).value);  
}
```

```
+++++
```

```
+++++
```

Two-way data binding

```
<input type="text" [value]="nome" (input)="nome = $event.target.value" />
```

```
<p> vc digitou {{ nome }}</p>
```

nome variavel declarada no componente]

```
*****
```

```
*****
```

ngModel = trabalhando com classe

```
[(ngModel)]="nomecampo"
```

```
<input type="text" [(ngModel)]="pessoa.nome" />
```

```
<input type="text" [(ngModel)]="pessoa.idade" />
```

<p>{{pessoa.nome}} e temos tambem {{pessoa.idade}}

no componente criar o objeto

```
pessoa: any = {  
  nome: 'abcde',  
  idade: 20  
}
```

ob: coloquei any porque eu nao queria usar uma class

usar variaveis entre dois componente

```
@Input('nomevariavel') nomevariavellocal: string = '';
```

LISTA DE EVENTOS

onBlur = remove o foco do elemento

onChange = muda o valor do elemento

onClick = o elemento é clicado pelo usuário

onFocus = o elemento é focado

onKeyPress = o usuário pressiona uma tecla sobre o elemento

onLoad = carrega o elemento por completo

onMouseOver = define ação quando o usuário passa o mouse sobre o elemento

onMouseOut = define ação quando o usuário retira o mouse sobre o elemento

onSubmit = define ação ao enviar um formulário

ngOnInit = quando component e inicializado

Subject - variaveis globais

```
private logado$ = new Subject();
```

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

```
import { Subject } from 'rxjs/Subject';
```

```
@Injectable()
```

```
export class MessageService {
```

```
  public message = new Subject<string>();
```

```
  setMessage(value: string) {
```

```
    this.message.next(value);
```

```
  }
```

```
}
```

nos componentes

```
this.messageService.setMessage(event.value);
```

=====
=====

API C# MVC

```
public class BaseApiController : ApiController
{
    protected readonly UserDBEntities UserDB = new UserDBEntities();
    protected HttpResponseMessage ToJson(dynamic obj)
    {
        var response = Request.CreateResponse(HttpStatusCode.OK);
        response.Content = new StringContent(JsonConvert.SerializeObject(obj),
        Encoding.UTF8, "application/json");
        return response;
    }
}
```

+++++

adicionar um controller normalmente e escolher ApiController

```
[RoutePrefix("api/APIPESSOA")]
public class APIPESSOAController : BaseApiController
{
    facadeADIPESSOA facade = new facadeADIPESSOA();

    //api/APIPESSOA/GetAll
    [AcceptVerbs("GET")]
```

```

//[Route("GetAll/{codigo}")]
[Route("GetAll")]
public HttpResponseMessage GetAll()
{
    List<ADIPESSOA> Lista = new List<ADIPESSOA>();
    Lista = facade.APIListAll();
    return ToJson(Lista.AsEnumerable());
}

```

Route

a rota tem de ficar igual o de cima

no webconfig do projeto

```

<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Headers" value="Content-Type" />
      <add name="Access-Control-Allow-Methods" value="GET, POST, PUT, DELETE, OPTIONS"
/>
    </customHeaders>
  </httpProtocol>

```

WebApiConfig

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;

using System.Net.Http.Formatting;

using System.Net.Http.Headers;

using System.Web.Http;

using System.Web.Http.Cors;

namespace smartcontrol
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services

            //config.EnableCors(cors);

            // Web API routes

            /*
            config.MapHttpAttributeRoutes();

            config.Formatters.JsonFormatter.SupportedMediaTypes.Add(new
MediaHeaderValue("application/octet-stream"));

            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );
            */

            config.Formatters.Clear();

            config.Formatters.Add(new JsonMediaTypeFormatter());

            config.MapHttpAttributeRoutes();

```

```

config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = RouteParameter.Optional }
);

//var cors = new EnableCorsAttribute("*", "*", "*");
//config.EnableCors(cors);

//var corsAttr = new EnableCorsAttribute("http://localhost:4200", "*", "*");
//config.EnableCors(corsAttr);

//config.EnableCors();

}
}
}

```

```

=====
=====

```

C# - METODO RETORNA JSON

[HttpGet]

```

public JsonResult GetDadosLayout()
{
    AcessoSessao.GetAcessoSessao();
    modalLAYOUT dados = new modalLAYOUT();
}

```

```
datos = GetDatosUsuarioLogado();
```

```
return Json(datos, JsonRequestBehavior.AllowGet);
```

```
}
```

```
=====
```

```
=====
```

```
=====
```

```
=====
```

=====
=====

ANGULA SELEC UPDAT DELETE DADOS BANCO

+++++

NO C#

+++++

API C# MVC

```
public class BaseApiController : ApiController
{
    protected readonly UserDBEntities UserDB = new UserDBEntities();
    protected HttpResponseMessage ToJson(dynamic obj)
    {
        var response = Request.CreateResponse(HttpStatusCode.OK);
        response.Content = new StringContent(JsonConvert.SerializeObject(obj),
        Encoding.UTF8, "application/json");
        return response;
    }
}
```

+++++

adicionar um controller normalmente e escolher ApiController

```
[RoutePrefix("api/APIPESSOA")]
```

```
public class APIPESSOAController : BaseApiController
```

```

{
    facadeADIPESSOA facade = new facadeADIPESSOA();

    //api/APIPESSOA/GetAll
    [AcceptVerbs("GET")]
    //[Route("GetAll/{codigo}")]
    [Route("GetAll")]
    public HttpResponseMessage GetAll()
    {
        List<ADIPESSOA> Lista = new List<ADIPESSOA>();
        Lista = facade.APIListAll();
        return ToJson(Lista.AsEnumerable());
    }
}

```

Route

a rota tem de ficar igual o de cima

no webconfig do projeto

```

<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Headers" value="Content-Type" />
      <add name="Access-Control-Allow-Methods" value="GET, POST, PUT, DELETE, OPTIONS" />
    </customHeaders>
  </httpProtocol>

```

+++++

MENU E ROTAS

+++++

no app.component.html

<!--The content below is only a placeholder and can be replaced.-->

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand" href="#">Navbar</a>

  <a [routerLink]="['/']" class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </a>

  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-item nav-link" [routerLink]="['/home']" routerLinkActive="active" >Home
<span class="sr-only">(current)</span></a>

      <a class="nav-item nav-link" [routerLink]="['/curso']" routerLinkActive="active"
>Curso</a>

      <a class="nav-item nav-link" [routerLink]="['/Cursoparametro','20','vanderlei']"
routerLinkActive="active" >Curso Parametros</a>

      <a class="nav-item nav-link" [routerLink]="['/database']" routerLinkActive="active"
>database</a>
    </div>
  </div>
</nav>
<div class="container" >
  <router-outlet></router-outlet>
</div>
```


criar um arquivo

clicar com botão direito na pasta app novo arquivo

nome : app-routing.module.ts

```
import { DatabaseComponent } from './database/database.component';  
import { NgModule } from '@angular/core';  
import { Routes, RouterModule } from '@angular/router';  
import { HomeComponent } from './home/home.component';  
import { CursoComponent } from './curso/curso.component';  
import { CursoparametroComponent } from './cursoparametro/cursoparametro.component';  
import { ModuleWithProviders } from '@angular/compiler/src/core';
```

```
const routes: Routes = [  
  {path: 'home', component: HomeComponent},  
  {path: 'curso', component: CursoComponent},  
  {path: 'Cursoparametro/:id/:nome', component: CursoparametroComponent},  
  {path: 'database', component: DatabaseComponent},  
  
  {path: '', component: HomeComponent}
```

```
];
```

```
export const routing: ModuleWithProviders = RouterModule.forRoot(routes);
```

abrir o arquivo app.module.ts

adicionar o routing no import

```
imports: [  

```



```
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { CursoComponent } from './curso/curso.component';
import { Curso02Component } from './curso02/curso02.component';
import { DatabaseComponent } from './database/database.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    CursoComponent,
    Curso02Component,
    DatabaseComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    AppRoutingModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

----- SERVICES -----

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Pessoa } from './pessoa';
import { tap } from 'rxjs/operators';

@Injectable({
  providedIn: 'root'
})
export class DatabaseService {

  private readonly api = 'http://localhost:54199/api/APIPESSOA/GetAll';
  constructor(private http: HttpClient) { }

  list(){
    return this.http.get<Pessoa[]>(this.api)
      .pipe(tap(console.log));
  }
}

```

----- COMPONENTES -----

```

import { DatabaseService } from './database.service';
import { Component, OnInit } from '@angular/core';

```

```
import { Pessoa } from './pessoa';

@Component({
  selector: 'app-database',
  templateUrl: './database.component.html',
  styleUrls: ['./database.component.css']
})
export class DatabaseComponent implements OnInit {

  ListaPessoa: Pessoa[];

  constructor(private services: DatabaseService) {}

  ngOnInit() {
    //this.services.list().subscribe(console.log);
    this.services.list().subscribe(dados => this.ListaPessoa = dados);
  }

}
```

----- HTML -----

<section class="property-binding">

```
<article>
```

```
<h1>Interpolação</h1>
```

```
<br>{{ url }}
```

```
<table class="table table-hover">
```

```
<thead>
```

```
<tr>
```

```
<th>Id</th>
```

```
<th>Curso</th>
```

```
<th></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr *ngFor="let pessoa of ListaPessoa" >
```

```
<td>{{ pessoa.ID }}</td>
```

```
<td>{{ pessoa.NOME }}</td>
```

```
<td>
```

```
<button class="btn btn-outline-warning mb-1 btn-sn" >Atualizar </button>
```

```
<button class="btn btn-outline-warning mb-1 btn-sn" >Remover </button>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
</article>
```

```
</section>
```

----- CLASSES -----

Comando para criar uma classe

ng g class NOME-DA-MINHA-CLASSE

```
export class Pessoa {
  public ID: string;
  public ADMEMPRESA: string;
  public ADMFILIAL: string;
  public ADITIPOSTATUS: string;
  public ADITIPOPESSOA: string;
  public ADITIPOLIDER: string;
  public IDFAMILIA: string;
  public CATEGORIA: string;
  public NOME: string;
  public SOBRENOME: string;
  public CPF: string;
  public RG: string;
  public DATANASCIMENTO: string;
  public DATACADASTRO: string;
  public DATASAIDA: string;
  public DATABATISMO: string;
  public DATAEXCLUIDO: string;
  public ADIPESSOAPAI: string;
  public ADIPESSOAMAE: string;
  public TELEFONE: string;
```



```
public CELULAR: string;
public SALARIO: string;
public DIZIMISTA: string;
public CEP: string;
public ENDERECO: string;
public COMPLEMENTO: string;
public NUMERO: string;
public ADIUF: string;
public ADICIDADE: string;
public ADIBAIRRO: string;
public OBSERVACAO: string;
public FOTO: string;
public ATIVO: string;
}
```

exemplo de uso das classes

componente

```
export class ADIPESSOAComponent implements OnInit {
  titulo = 'Controle de classes';
  dados = new ADIPESSOA;
  constructor() {
    this.dados.NOME = "Vanderlei Soares de Oliveira";
    this.dados.SOBRENOME = "Oliveira"
    this.dados.ATIVO = "SIM";
  }
}
```

html

<p>

adipessoa works!

</p>

<p>Tituo:{{titulo}}</p>

<p>dados:

Nome: {{dados.NOME}}

SobreNome: {{dados.SOBRENOME}}

Ativo: {{dados.ATIVO}}

</p>

LISTA OBJETO

+++++

FIM NO ANGULA USAR O HTTP

+++++

VARIAVEIS

let a: number //e.g 1, 2, 3

let b: boolean //e.g true, false

let c: string //e.g "abel agoi"

let d: any //this can take any other types

let e: number[] //array of numbers e.g [1, 3, 54]

let f: any[] //any array e.g [1, "abel agoi", true]

+++++

MENSAGEM LENDO PROCESSANDO

+++++

no componentes usando async

ListaPessoa\$: Observable<Pessoa[]>;

this.ListaPessoa\$ = this.services.list();

no html

so exibir a tabela se existir dados

*ngIf="ListaPessoa\$ | async as ListaPessoa; else loading">

<table class="table table-hover" *ngIf="ListaPessoa\$ | async as ListaPessoa; else loading">

<thead>

<tr>

<th>Id</th>

<th>Curso</th>

<th></th>

</tr>

</thead>

<tbody>

<tr *ngFor="let pessoa of ListaPessoa" >

<td>{{ pessoa.ID }}</td>

<td>{{ pessoa.NOME }}</td>

<td>

<button class="btn btn-outline-warning mb-1 btn-sn" >Atualizar </button>

<button class="btn btn-outline-warning mb-1 btn-sn" >Remover </button>

</td>

```
</tr>
</tbody>
</table>
<br>
<ng-template #loading>
  <span>Carregando Dados</span>
</ng-template>
```

```
+++++
FIM MENSAGEM LENDO PROCESSANDO
+++++
```

```
+++++
MENSAGEM DE ERRO
+++++
no arquivo de componentes
```

```
error$ = new Subject<boolean>();

//ListaPessoa: Pessoa[];
ListaPessoa$: Observable<Pessoa[]>;

constructor(private services: DatabaseService) { }
```

```
ngOnInit() {  
  //this.services.list().subscribe(console.log);  
  //this.services.list().subscribe(dados => this.ListaPessoa = dados);  
  //this.ListaPessoa$ = this.services.list();  
  this.ListaPessoa$ = this.services.list()  
  .pipe(  
    catchError(error =>{  
      //console.error(error);  
      this.error$.next(true);  
      return empty();  
    })  
  )  
}  
  
}
```

no arquivo html

```
<table class="table table-hover" *ngIf="ListaPessoa$ | async as ListaPessoa; else  
loadingNoError">
```

```
  <thead>  
    <tr>  
      <th>Id</th>  
      <th>Curso</th>  
      <th></th>  
    </tr>  
  </thead>
```

```
<tbody>
  <tr *ngFor="let pessoa of ListaPessoa" >
    <td>{{ pessoa.ID }}</td>
    <td>{{ pessoa.NOME }}</td>
    <td>
      <button class="btn btn-outline-warning mb-1 btn-sn" >Atualizar </button>
      <button class="btn btn-outline-warning mb-1 btn-sn" >Remover </button>
    </td>
  </tr>
</tbody>
</table>
<br>
<ng-template #loadingNoError>
  <div *ngIf="error$ | async; else loading">Erro ao carregar dados.</div>
  <ng-template #loading >
    <span>Carregando Dados</span>
  </ng-template>
</ng-template>
```

+++++

FIM MENSAGEM DE ERRO

+++++

+++++

FORM - NOVO formgroup

+++++

obs:

pra atualizar put e pra inserir post

html - databaseform.component.html

```
<div class="card">
  <div class="card-header">
    Nova Cadastro
  </div>
  <div class="card-body">
    <h5 class="card-title">Pessoa</h5>

    <form style="width:300px;" [formGroup]="pessoaFormGroup" (ngSubmit)="addDados()">
      <div class="form-group col-sm-5">
        <label for="lbcodigo">Codigo:</label>
        <input type="text" class="form-control" formControlName="ID">
      </div>
      <div class="form-group col-sm-12">
        <label for="lenome">Nome:</label>
        <input type="text" class="form-control" formControlName="NOME"
placeholder="Nome">
      </div>
      <div class="form-group col-sm-12">
        <label for="lesobrenome">Sobrenome:</label>
        <input type="text" class="form-control" formControlName="SOBRENOME"
placeholder="Sobrenome">
      </div>

      <button type="submit" class="btn btn-primary">Salvar</button>
      <button type="button" class="btn btn-primary" style="margin-left:3px;"
>Cancelar</button>
    </form>
  </div>
</div>
```

```
</div>
```

```
</div>
```

```
*****
```

```
databaseform.component.ts
```

```
import { Component, OnInit } from '@angular/core';
```

```
import { FormGroup, FormBuilder, FormControl } from '@angular/forms';
```

```
import { Pessoa } from '../database/pessoa';
```

```
import { DatabaseService } from '../database/database.service';
```

```
@Component({
```

```
  selector: 'app-databaseform',
```

```
  templateUrl: './databaseform.component.html',
```

```
  styleUrls: ['./databaseform.component.css']
```

```
})
```

```
export class DatabaseformComponent implements OnInit {
```

```
  pessoaFormGroup: FormGroup;
```

```
  private pessoa: Pessoa;
```

```
  constructor(private services: DatabaseService, private fb: FormBuilder) {}
```

```
  ngOnInit() {
```

```
    this.pessoaFormGroup = this.fb.group({
```

```
      ID: new FormControl(""),
```

```
      NOME: new FormControl(""),
```



```

        SOBRENOME: new FormControl('')
    });

}

addDados() {

    console.log('submit');
    this.pessoa = <Pessoa> this.pessoaFormGroup.value;
    //console.log(this.pessoa instanceof Pessoa);
    //console.log(this.pessoa);
    //console.log(' ----- nome ---- -');
    //console.log(this.pessoa.NOME);

    this.services.addNovo(this.pessoa);
    console.log(this.services.retposdata);

}

}

```

app.module.ts

```

import { HttpClientModule } from '@angular/common/http';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { ReactiveFormsModule,FormsModule } from '@angular/forms';

```

```
import { BsDropdownModule } from 'ngx-bootstrap/dropdown';
import { TooltipModule } from 'ngx-bootstrap/tooltip';
import { ModalModule } from 'ngx-bootstrap/modal';

import { AppComponent } from './app.component';
import { CursoComponent } from './curso/curso.component';
import { Curso02Component } from './curso02/curso02.component';
import { DatabaseComponent } from './database/database.component';
import { DatabaseformComponent } from './databaseform/databaseform.component';
import { HomeComponent } from './home/home.component';
import { routing } from './app-routing.module';
import { CursoparametroComponent } from './cursoparametro/cursoparametro.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    CursoComponent,
    Curso02Component,
    DatabaseComponent,
    DatabaseformComponent,
    HomeComponent,
    CursoparametroComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    ReactiveFormsModule,//para usar formularios com formgroup
    HttpClientModule,
    BsDropdownModule.forRoot(),
    TooltipModule.forRoot(),
    ModalModule.forRoot(),
```

```
    routing
  ],
  exports: [BsDropdownModule, TooltipModule, ModalModule],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

classe pessoa.ts

```
export class Pessoa {
  public ID: string;
  public ADMEMPRESA: string;
  public ADMFILIAL: string;
  public ADITIPOSTATUS: string;
  public ADITIPOPESSOA: string;
  public ADITIPOLIDER: string;
  public IDFAMILIA: string;
  public CATEGORIA: string;
  public NOME: string;
  public SOBRENOME: string;
  public CPF: string;
  public RG: string;
  public DATANASCIMENTO: string;
  public DATACADASTRO: string;
  public DATASAIDA: string;
  public DATABATISMO: string;
  public DATAEXCLUIDO: string;
  public ADIPESSOAPAI: string;
```

```
public ADIPESSOAMAE: string;
public TELEFONE: string;
public CELULAR: string;
public SALARIO: string;
public DIZIMISTA: string;
public CEP: string;
public ENDERECO: string;
public COMPLEMENTO: string;
public NUMERO: string;
public ADIUF: string;
public ADICIDADE: string;
public ADIBAIRRO: string;
public OBSERVACAO: string;
public FOTO: string;
public ATIVO: string;
}
```

database.service.ts

```
import { environment } from './../environments/environment';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Injectable, NgModule } from '@angular/core';
import { Pessoa } from './pessoa';
import { tap, delay, take, catchError } from 'rxjs/operators';
import { Observable } from 'rxjs';
```

```
@Injectable({
  providedIn: 'root'
})
```

```

export class DatabaseService {

    //LocalHostAPI: 'http://localhost:54199/api/APIPESSOA/GetAll'
    //private readonly API = environment.LocalHostAPI+'/APIPESSOA/GetAll';
    private readonly API = `${environment.LocalHostAPI}/APIPESSOA/GetAll`;
    private readonly APIADICIONAR = `${environment.LocalHostAPI}/APIPESSOA`;
    private readonly APIATUALIZAR = `${environment.LocalHostAPI}/APIPESSOA/Put`;

    constructor(private http: HttpClient) { }

    list(){
        return this.http.get<Pessoa[]>(this.API)
            .pipe(
                //delay(2000),
                tap(console.log)
            );
    }

    /*
    addNovo(pessoa: Pessoa){
        return this.http.post(this.APIADICIONAR,pessoa).pipe(take(1));
    }
    */

    /*
    addNovo (pessoa: Pessoa): Observable<Pessoa> {

        const httpOptions = {
            headers: new HttpHeaders({
                'Content-Type': 'application/json',
            })
        };
    }
    */
}

```

```
    'Authorization': 'my-auth-token'
  })
};
```

```
return this.http.post<Pessoa>(this.APIADICIONAR, pessoa, httpOptions)
  .pipe(
    //catchError(this.handleError('addNovo', pessoa))
    take(1)
  );
}
*/
```

```
public retposdata;
//http://localhost:54199/api/APIPESSOA/
addNovo(pessoa: Pessoa){
  /*
  const httpOptions = {
    headers: new HttpHeaders({
      'Content-Type': 'application/json',
      'Accept': 'application/json',
      'access-token': window.localStorage.getItem("access-token"),
      'client': window.localStorage.getItem("client"),
      'uid': window.localStorage.getItem("uid")
    })
  };
  */
  const retval =
  this.http.post(this.APIADICIONAR,{FstVarValue:'222',SndVarValue:'3333'}).subscribe
  (
    data => {this.retposdata = data;
```

```
});
```

```
}
```

```
atualizar(pessoa: Pessoa){
```

```
    return this.http.put(this.APIATUALIZAR,pessoa).pipe(take(1));
```

```
}
```

```
}
```

```
+++++
```

```
EDITAR - FORM - formgroup
```

```
+++++
```

```
+++++
```

```
ARRAY JSON
```

```
+++++
```

```
var my_obj = {'teste':{'name':'Teste 123'},'lorem':{'name':'Lorem Ipsum'}};
```

```
{{ my_obj['lorem']['name'] }}
```

```
{{ my_obj.lorem.name }}
```

```
{{ my_obj['lorem'].name }}
```

```
-----
```

```
var my_array = [{ 'name' : 'Teste 123' }, { 'name' : 'Lorem Ipsum' }];  
{{ my_array[0].name }}
```

```
var my_array = ['Teste 123', 'Lorem Ipsum'];  
{{ my_array[0] }}
```

```
<table>  
  <thead>  
    <th>Name</th>  
    <th>Index</th>  
  </thead>  
  <tbody>  
    <tr *ngFor="let heros of HEROES">  
      <td>{{heros.id}}</td>  
      <td>{{heros.name}}</td>  
    </tr>  
  </tbody>  
</table>
```

```
HEROES = [  
  {id: 1, name:'Superman'},  
  {id: 2, name:'Batman'},  
  {id: 5, name:'BatGirl'},  
  {id: 3, name:'Robin'},  
  {id: 4, name:'Flash'}  
];
```

outros exemplos

```
<tr *ngFor="let hero of heroes; let i = index">
```



```
<td>{{hero.name}}</td>
<td>{{i}}</td>
</tr>
```

```
<tr *ngFor="let hero of heroes; let first = first; let last = last"
  [ngClass]="{ first: first, last: last }">
  <td>{{hero.name}}</td>
</tr>
```

```
<div *ngFor="let hero of HEROES">{{hero.name}}</div>
<div *ngFor="let hero of heroes; let i=index">{{i + 1}} - {{hero.name}}</div>
```

=====

```
<input #newHero
  (keyup.enter)="addHero(newHero.value)"
  (blur)="addHero(newHero.value); newHero.value=" ">
<button (click)="addHero(newHero.value)">Add</button>
<ul><li *ngFor="let hero of heroes">{{hero}}</li></ul>
```

```
heroes = ['Windstorm', 'Bombasto', 'Magneta', 'Tornado'];
addHero(newHero: string) {
  if (newHero) {
    this.heroes.push(newHero);
  }
}
```

```
}
```

```
+++++
```

```
Clique me
```

```
+++++
```

```
<button (click)="onClickMe()">Click me!</button>
```

```
  {{clickMessage}}
```

```
clickMessage = '';
```

```
onClickMe() {
```

```
  this.clickMessage = 'You are my hero!';
```

```
}
```

```
=====
```

```
<input (keyup)="onKey($event)">
```

```
<p>{{values}}</p>
```

```
values = '';
```

```
onKey(event: any) { // without type info
```

```
  this.values += event.target.value + ' | ';
```

```
}
```

+++++

ngIf

+++++

```
<div *ngIf="currentHero">Hello, {{currentHero.name}}</div>
```

```
<div *ngIf="nullHero">Hello, {{nullHero.name}}</div>
```

+++++

onSubmit

+++++

```
<form (ngSubmit)="onSubmit(heroForm)" #heroForm="ngForm">
```

```
  <div class="form-group">
```

```
    <label for="name">Name
```

```
      <input class="form-control" name="name" required [(ngModel)]="hero.name">
```

```
    </label>
```

```
  </div>
```

```
  <button type="submit" [disabled]="!heroForm.form.valid">Submit</button>
```

```
</form>
```

```
<div [hidden]="!heroForm.form.valid">
```

```
  {{submitMessage}}
```

```
</div>
```

+++++

ng-select - select

+++++

```
<ng-select [items]="people"
  bindLabel="name"
  bindValue="id"
  [(ngModel)]="selectedPersonId2">
</ng-select>
```

```
<h2>Selecione o Pais: </h2>
```

```
<select class="form-control" (change)="onSelect($event.target.value)" id="sel1" >
  <option *ngFor="let pais of paises" value= {{pais.id}}>
    {{pais.nome}}
  </option>
</select>
<br>
<br>
id = {{seleccionado}}
<br>
<br>
```

```
seleccionado: string;
//paises : Pais[];
paises = [
  new Pais(1, 'Argentina' ),
  new Pais(2, 'Bolivia' ),
  new Pais(3, 'Brasil' ),
  new Pais(4, 'Chile' ),
  new Pais(5, 'Equador')
];

constructor ( ) {
```

```
}
```

```
//constructor(private _dataService: DataserviceService) {
```

```
// this.paises = this._dataService.getPaises();
```

```
// }
```

```
onSelect(id) {
```

```
  this.seleccionado = id;//this._dataService.getCidades().filter((item)=> item.paisid == id);
```

```
}
```

```
ngOnInit() {
```

```
}
```

```
}
```

```
export class Pais {
```

```
  constructor(public id: number, public nome: string) { }
```

```
}
```

```
+++++
```

```
+++++
```

```
+++++
```

EXEMPLO TABELA E FORM

+++++

main.ts

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
```

```
import { AppModule } from './app.module';
```

```
platformBrowserDynamic().bootstrapModule(AppModule);
```

app.routing.ts

```
import { ModuleWithProviders } from '@angular/core';
```

```
import { Routes, RouterModule } from '@angular/router';
```

```
import { UserComponent } from './components/user.component';
```

```
import { HomeComponent } from './components/home.component';
```

```
const appRoutes: Routes = [
```

```
  { path: '', redirectTo: 'home', pathMatch: 'full' },
```

```
  { path: 'home', component: HomeComponent },
```

```
  { path: 'user', component: UserComponent }]
```

```
];
```

```
export const routing: ModuleWithProviders =
```

```
  RouterModule.forRoot(appRoutes);
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { APP_BASE_HREF } from '@angular/common';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
import { Ng2Bs3ModalModule } from 'ng2-bs3-modal/ng2-bs3-modal';
import { HttpClientModule } from '@angular/http';
import { routing } from './app.routing';
import { UserComponent } from './components/user.component';
import { HomeComponent } from './components/home.component';
import { UserService } from './Service/user.service'
```

```
@NgModule({
  imports: [BrowserModule, ReactiveFormsModule, HttpClientModule, routing,
    Ng2Bs3ModalModule],
  declarations: [AppComponent, UserComponent, HomeComponent],
  providers: [{ provide: APP_BASE_HREF, useValue: '/' }, UserService],
  bootstrap: [AppComponent]
})
```

```
export class AppModule { }
```

app.component.ts

```
import { Component } from "@angular/core"
```

```
@Component({
  selector: "user-app",
  template: `
```

```
<div>
  <nav class='navbar navbar-inverse'>
    <div class='container-fluid'>
      <ul class='nav navbar-nav'>
        <li><a [routerLink]="['home']">Home</a></li>
        <li><a [routerLink]="['user']">Users Management</a></li>
      </ul>
    </div>
  </nav>
  <div class='container'>
    <router-outlet></router-outlet>
  </div>
</div>
```

```
})
```

```
export class AppComponent {
```

```
}
```

pasta - Model

user.ts

```
export interface IUser {
```

```
  Id: number,
```

```
  FirstName: string,
```

```
  LastName: string,
```

```
  Gender: string
```

```
}
```


pasta Service

user.service.ts

```
import { Injectable } from '@angular/core';  
import { Http, Response, Headers, RequestOptions } from '@angular/http';  
import { Observable } from 'rxjs/Observable';  
import 'rxjs/add/operator/map';  
import 'rxjs/add/operator/do';  
import 'rxjs/add/operator/catch';
```

```
@Injectable()
```

```
export class UserService {
```

```
  constructor(private _http: Http) { }
```

```
  get(url: string): Observable<any> {
```

```
    return this._http.get(url)
```

```
      .map((response: Response) => <any>response.json())
```

```
      // .do(data => console.log("All: " + JSON.stringify(data)))
```

```
      .catch(this.handleError);
```

```
  }
```

```
  post(url: string, model: any): Observable<any> {
```

```
    let body = JSON.stringify(model);
```

```
    let headers = new Headers({ 'Content-Type': 'application/json' });
```

```
    let options = new RequestOptions({ headers: headers });
```

```
    return this._http.post(url, body, options)
```

```
      .map((response: Response) => <any>response.json())
```

```
      .catch(this.handleError);
```

```
  }
```

```

put(url: string, id: number, model: any): Observable<any> {
  let body = JSON.stringify(model);
  let headers = new Headers({ 'Content-Type': 'application/json' });
  let options = new RequestOptions({ headers: headers });
  return this._http.put(url+id, body, options)
    .map((response: Response) => <any>response.json())
    .catch(this.handleError);
}

```

```

delete(url: string, id: number): Observable<any> {
  let headers = new Headers({ 'Content-Type': 'application/json' });
  let options = new RequestOptions({ headers: headers });
  return this._http.delete(url+id,options)
    .map((response: Response) => <any>response.json())
    .catch(this.handleError);
}

```

```

private handleError(error: Response) {
  console.error(error);
  return Observable.throw(error.json().error || 'Server error');
}

```

```

}

```

pasta component

user.component.ts

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { UserService } from '../Service/user.service';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { ModalComponent } from 'ng2-bs3-modal/ng2-bs3-modal';

```

```

import { IUser } from '../Model/user';
import { DBOperation } from '../Shared/enum';
import { Observable } from 'rxjs/Rx';
import { Global } from '../Shared/global';

@Component({
  templateUrl: 'app/Components/user.component.html'
})

export class UserComponent implements OnInit {

  @ViewChild('modal') modal: ModalComponent;
  users: IUser[];
  user: IUser;
  msg: string;
  indLoading: boolean = false;
  userFrm: FormGroup;
  dbops: DBOperation;
  modalTitle: string;
  modalBtnTitle: string;

  constructor(private fb: FormBuilder, private _userService: UserService) {}

  ngOnInit(): void {
    this.userFrm = this.fb.group({
      Id: [''],
      FirstName: ['', Validators.required],
      LastName: [''],
      Gender: ['', Validators.required]
    });
    this.LoadUsers();
  }

```

```
}
```

```
LoadUsers(): void {  
    this.indLoading = true;  
    this._userService.get(Global.BASE_USER_ENDPOINT)  
        .subscribe(users => { this.users = users; this.indLoading = false; },  
            error => this.msg = <any>error);  
}
```

```
addUser() {  
    this.dbops = DBOperation.create;  
    this.SetControlsState(true);  
    this.modalTitle = "Add New User";  
    this.modalBtnTitle = "Add";  
    this.userFrm.reset();  
    this.modal.open();  
}
```

```
editUser(id: number) {  
    this.dbops = DBOperation.update;  
    this.SetControlsState(true);  
    this.modalTitle = "Edit User";  
    this.modalBtnTitle = "Update";  
    this.user = this.users.filter(x => x.Id == id)[0];  
    this.userFrm.setValue(this.user);  
    this.modal.open();  
}
```

```
deleteUser(id: number) {  
    this.dbops = DBOperation.delete;  
    this.SetControlsState(false);  
}
```

```
    this.modalTitle = "Confirm to Delete?";
    this.modalBtnTitle = "Delete";
    this.user = this.users.filter(x => x.Id == id)[0];
    this.userFrm.setValue(this.user);
    this.modal.open();
}
```

```
onSubmit(formData: any) {
```

```
    this.msg = "";
```

```
    switch (this.dbops) {
```

```
        case DBOperation.create:
```

```
            this._userService.post(Global.BASE_USER_ENDPOINT, formData._value).subscribe(
```

```
                data => {
```

```
                    if (data == 1) //Success
```

```
                    {
```

```
                        this.msg = "Data successfully added.";
```

```
                        this.LoadUsers();
```

```
                    }
```

```
                else
```

```
                {
```

```
                    this.msg = "There is some issue in saving records, please contact to system administrator!"
```

```
                }
```

```
            this.modal.dismiss();
```

```
        },
```

```
        error => {
```

```
            this.msg = error;
```

```
        }
```

```
    );
```

```
        break;

        case DBOperation.update:

            this._userService.put(Global.BASE_USER_ENDPOINT, formData._value.Id,
            formData._value).subscribe(

                data => {

                    if (data == 1) //Success

                    {

                        this.msg = "Data successfully updated.";

                        this.LoadUsers();

                    }

                    else {

                        this.msg = "There is some issue in saving records, please contact to system
administrator!"

                    }

                    this.modal.dismiss();

                },

                error => {

                    this.msg = error;

                }

            );

            break;

        case DBOperation.delete:

            this._userService.delete(Global.BASE_USER_ENDPOINT,
            formData._value.Id).subscribe(

                data => {

                    if (data == 1) //Success

                    {

                        this.msg = "Data successfully deleted.";

                        this.LoadUsers();

                    }

                    else {
```

```
        this.msg = "There is some issue in saving records, please contact to system administrator!"
```

```
    }
```

```
        this.modal.dismiss();
```

```
    },
```

```
    error => {
```

```
        this.msg = error;
```

```
    }
```

```
);
```

```
break;
```

```
    }
```

```
}
```

```
SetControlsState(isEnable: boolean)
```

```
{
```

```
    isEnable ? this.userFrm.enable() : this.userFrm.disable();
```

```
}
```

```
}
```

```
user.component.html
```

```
<div class='panel panel-primary'>
```

```
    <div class='panel-heading'>
```

```
        User Management
```

```
    </div>
```

```
    <div class='panel-body'>
```

```
        <div class='table-responsive'>
```

```
            <div style="padding-bottom:10px"><button class="btn btn-primary" (click)="addUser()">Add</button></div>
```

```
<div class="alert alert-info" role="alert" *ngIf="indLoading"> Loading...</div>
```

```
<div *ngIf='users && users.length==0' class="alert alert-info" role="alert">No record  
found!</div>
```

```
<table class='table table-striped' *ngIf='users && users.length'>
```

```
<thead>
```

```
<tr>
```

```
<th>First Name</th>
```

```
<th>Last Name</th>
```

```
<th>Gender</th>
```

```
<th></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr *ngFor="let user of users">
```

```
<td>{{user.FirstName}}</td>
```

```
<td>{{user.LastName}}</td>
```

```
<td>{{user.Gender}}</td>
```

```
<td>
```

```
<button title="Edit" class="btn btn-primary"  
(click)="editUser(user.Id)">Edit</button>
```

```
<button title="Delete" class="btn btn-danger"  
(click)="deleteUser(user.Id)">Delete</button>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
<div>
```

```
</div>
```

```
</div>
```

```
<div *ngIf="msg" role="alert" class="alert alert-info alert-dismissible">
```

```
<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-  
hidden="true">&times;</span></button>
```



```
<span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
<span class="sr-only">Error:</span>
  {{msg}}
</div>
</div>
</div>
```

```
<modal #modal>
```

```
<form novalidate (ngSubmit)="onSubmit(userFrm)" [formGroup]="userFrm">
```

```
<modal-header [show-close]="true">
```

```
<h4 class="modal-title">{{modalTitle}}</h4>
```

```
</modal-header>
```

```
<modal-body>
```

```
<div class="form-group">
```

```
<div>
```

```
<span>Full name*</span>
```

```
<input type="text" class="form-control" placeholder="First Name"
formControlName="FirstName">
```

```
</div>
```

```
<div>
```

```
<span>Full name</span>
```

```
<input type="text" class="form-control" placeholder="Last Name"
formControlName="LastName">
```

```
</div>
```

```
<div>
```

```
<span>Gender*</span>
```

```
<select formControlName="Gender" class="form-control">
```

```
<option>Male</option>
```

```
<option>Female</option>
```

```
</select>
```

```
</div>
```

```
    </div>
</modal-body>
<modal-footer>
  <div>
    <a class="btn btn-default" (click)="modal.dismiss()">Cancel</a>
    <button type="submit" [disabled]="userFrm.invalid" class="btn btn-
primary">{{modalBtnTitle}}</button>
  </div>
</modal-footer>
</form>
</modal>
```

pasta - Shared

enum.ts

```
export enum DBOperation {
  create = 1,
  update = 2,
  delete = 3
}
```

global.ts

```
export class Global {
  public static BASE_USER_ENDPOINT = 'api/userapi/';
}
```

_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
  <script src="/node_modules/core-js/client/shim.min.js"></script>
  <script src="/node_modules/zone.js/dist/zone.js"></script>
  <script src="/node_modules/systemjs/dist/system.src.js"></script>
  <script src="/systemjs.config.js"></script>
  <script>
    System.import('app').catch(function(err){ console.error(err); });
  </script>
</head>
<body>
  <div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
      <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
    </footer>
  </div>

  @Scripts.Render("~/bundles/jquery")
  @Scripts.Render("~/bundles/bootstrap")
  @RenderSection("scripts", required: false)
</body>
</html>
```

Index.cshtml

```
@{
```

```
    ViewBag.Title = "Index";
```

```
}
```

```
<body>
```

```
    <user-app>Loading...</user-app>
```

```
</body>
```

```
=====
=====
ANGULARJS
=====
=====
```

site como referencia

<https://www.w3schools.com/angular/>

ANGULARJS

link pra trabalhar com angular

```
<link rel="stylesheet" href="/w3css/4/w3.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

tag controle ng quando usada na div criar um controler olhe o nome do ng-model

```
<div ng-app="">  
  <p>Name : <input type="text" ng-model="name"></p>  
  <h1>Hello {{name}}</h1>  
</div>
```

calculo e variavel

```
<div ng-app="">  
  <p>My first expression: {{ 5 + 5 }}</p>  
</div>
```

```
<div ng-app="" ng-init="myCol='lightblue'">
```

```
<input style="background-color:{{myCol}}" ng-model="myCol">
```

```
</div>
```

```
<div ng-app="" ng-init="quantity=1;cost=5">
```

```
<p>Total in dollar: {{ quantity * cost }}</p>
```

```
</div>
```

```
<div ng-app="" ng-init="quantity=1;cost=5">
```

```
<p>Total in dollar: <span ng-bind="quantity * cost"></span></p>
```

```
</div>
```

```
*****
```

```
<div ng-app="" ng-init="firstName='John';lastName='Doe'">
```

```
<p>The name is {{ firstName + " " + lastName }}</p>
```

```
</div>
```

```
*****
```

```
<div ng-app="" ng-init="person={firstName:'John',lastName:'Doe'}">
```

```
<p>The name is {{ person.lastName }}</p>
```

```
</div>
```

```
*****
```

```
<div ng-app="" ng-init="points=[1,15,19,2,40]">
```

```
<p>The third result is <span ng-bind="points[2]"></span></p>
```

```
</div>
```

```
*****
```

```
<div ng-app="meuApp" ng-controller="meucontrolerCtrl">
```

```
    {{ firstName + " " + lastName }}
```

```
</div>
```

```
<script>
```

```
    var app = angular.module("meuApp", []);
```

```
    app.controller("meucontrolerCtrl", function ($scope) {
```

```
$scope.firstName = "Vanderlei";  
$scope.lastName = "Soares";  
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" w3-vanderlei></div>
```

```
<script>
```

```
var app = angular.module("myApp", []);
```

```
app.directive("w3Vanderlei", function () {
```

```
return {
```

```
template: "I was made in a directive constructor!"
```

```
};
```

```
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
{{ firstName + " " + lastName }}
```

```
</div>
```

```
<script src="myApp.js"></script>
```

```
<script src="myCtrl.js"></script>
```

criar um arquivo pra colocar os codigos

myApp.js

```
var app = angular.module("myApp", []);
```

criar um arquivo pra colocar os codigos

myCtrl.js

```
app.controller("myCtrl", function($scope) {
    $scope.firstName = "John";
    $scope.lastName= "Doe";
});
*****

<div ng-app="" ng-init="quantity=1;price=5">

Quantity: <input type="number" ng-model="quantity">
Costs: <input type="number" ng-model="price">

Total in dollar: {{ quantity * price }}

</div>
*****
```

LISTA LIST

```
<div ng-app="" ng-init="names=['Vanderlei','Jussara','Davi','Kayo']">
    <ul>
        <li ng-repeat="x in names">
            {{ x }}
        </li>
    </ul>
</div>
```

```
<div ng-app="" ng-init="names=[
{name:'Jani',country:'Norway'},
{name:'Hege',country:'Sweden'},
{name:'Kai',country:'Denmark'}]">
```



```
<ul>
  <li ng-repeat="x in names">
    {{ x.name + ', ' + x.country }}
  </li>
</ul>
```

```
</div>
```

```
*****
```

BODY

```
<body ng-app="myApp">
```

```
<w3-test-directive></w3-test-directive>
```

```
<script>
```

```
var app = angular.module("myApp", []);
```

```
app.directive("w3TestDirective", function() {
```

```
  return {
```

```
    template : "<h1>Made by a directive!</h1>"
```

```
  };
```

```
});
```

```
</script>
```

```
</body>
```

```
*****
```

USANDO O COMANDO NA CLASSE

```
<body ng-app="myApp">
```

```
<div class="w3-test-directive"></div>
```

```
<script>
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
  return {
    restrict : "C",
    template : "<h1>Made by a directive!</h1>"
  };
});
</script>
```

USANDO O COMANDO NA DIV COM UM NOME

```
<body ng-app="myApp">
```

```
<div w3-test-directive></div>
```

```
<script>
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
  return {
    template : "<h1>Made by a directive!</h1>"
  };
});
</script>
```

CRIANDO O COMANDO CRIANDO UMA TAG

```
<body ng-app="myApp">
```

```
<w3-test-directive></w3-test-directive>
```

```
<script>
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
  return {
    template : "<h1>Made by a directive!</h1>"
  };
});
</script>
```

VALIDAR EMAIL

```
<form ng-app="" name="myForm">
  Email:
  <input type="email" name="myAddress" ng-model="text">
  <span ng-show="myForm.myAddress.$error.email">Not a valid e-mail address</span>
</form>
```

```
<div ng-app="myApp" ng-controller="myCtrl">
  Name: <input ng-model="firstname">
  <h1>{{firstname}}</h1>
</div>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function ($scope) {
  $scope.firstname = "John";
  $scope.lastname = "Doe";
});
</script>
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
First Name: <input type="text" ng-model="firstName"><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.firstName = "John";
  $scope.lastName = "Doe";
});
```

```
</script>
```

```
*****
```

```
FUNCAO
```

```
<div ng-app="myApp" ng-controller="personCtrl">
```

```
First Name: <input type="text" ng-model="firstName"><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{fullName()}}
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('personCtrl', function($scope) {
  $scope.firstName = "John";
  $scope.lastName = "Doe";
```

```

$scope.fullName = function() {
    return $scope.firstName + " " + $scope.lastName;
};
});
</script>
*****

<div ng-app="myApp">

    <p>The rootScope's favorite color:</p>
    <h1>{{color}}</h1>

    <div ng-controller="myCtrl">
        <p>The scope of the controller's favorite color:</p>
        <h1>{{color}}</h1>
    </div>

    <p>The rootScope's favorite color is still:</p>
    <h1>{{color}}</h1>

    <script>
        var app = angular.module('myApp', []);
        app.run(function ($rootScope) {
            $rootScope.color = 'blue';
        });
        app.controller('myCtrl', function ($scope) {
            $scope.color = "red";
        });
    </script>
</div>
*****

```

AngularJS Filters

AngularJS provides filters to transform data:

- *currency Format a number to a currency format.
- *date Format a date to a specified format.
- *filter Select a subset of items from an array.
- *json Format an object to a JSON string.
- *limitTo Limits an array/string, into a specified number of elements/characters.
- *lowercase Format a string to lower case.
- *number Format a number to a string.
- *orderBy Orders an array by an expression.
- *uppercase Format a string to upper case.

```
<div ng-app="myApp" ng-controller="personCtrl">
```

```
<p>The name is {{ lastName | uppercase }}</p>
```

```
lowercase
```

```
</div>
```

```
<script>
```

```
angular.module('myApp', []).controller('personCtrl', function($scope) {
```

```
    $scope.firstName = "John",
```

```
    $scope.lastName = "Doe"
```

```
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="namesCtrl">
```

<p>Looping with objects:</p>

<li ng-repeat="x in names | orderBy:'country'">

 {{ x.name + ', ' + x.country }}

</div>

<script>

angular.module('myApp', []).controller('namesCtrl', function(\$scope) {

 \$scope.names = [

 {name:'Jani',country:'Norway'},

 {name:'Carl',country:'Sweden'},

 {name:'Margareth',country:'England'},

 {name:'Hege',country:'Norway'},

 {name:'Joe',country:'Denmark'},

 {name:'Gustav',country:'Sweden'},

 {name:'Birgit',country:'Denmark'},

 {name:'Mary',country:'England'},

 {name:'Kai',country:'Norway'}

];

});

</script>

<div ng-app="myApp" ng-controller="costCtrl">

<h1>Price: {{ price | currency }}</h1>

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);  
app.controller('costCtrl', function($scope) {  
    $scope.price = 58;  
});  
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="namesCtrl">
```

```
<ul>
```

```
<li ng-repeat="x in names | filter : 'i'">
```

```
    {{ x }}
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
<script>
```

```
angular.module('myApp', []).controller('namesCtrl', function($scope) {  
    $scope.names = [  
        'Jani',  
        'Carl',  
        'Margareth',  
        'Hege',  
        'Joe',  
        'Gustav',  
        'Birgit',  
        'Mary',
```



```
    'Kai'  
  ];  
});  
</script>
```

TABELA ORDEM MUITO LEGAL

<p>Click the table headers to change the sorting order:</p>

```
<div ng-app="myApp" ng-controller="namesCtrl">
```

```
<table border="1" width="100%">  
<tr>  
<th ng-click="orderByMe('name')">Name</th>  
<th ng-click="orderByMe('country')">Country</th>  
</tr>  
<tr ng-repeat="x in names | orderBy:myOrderBy">  
<td>{{x.name}}</td>  
<td>{{x.country}}</td>  
</tr>  
</table>
```

```
</div>
```

```
<script>  
angular.module('myApp', []).controller('namesCtrl', function($scope) {  
  $scope.names = [  
    {name:'Jani',country:'Norway'},  
    {name:'Carl',country:'Sweden'},  
    {name:'Margareth',country:'England'},  
    {name:'Hege',country:'Norway'},
```

```

    {name:'Joe',country:'Denmark'},
    {name:'Gustav',country:'Sweden'},
    {name:'Birgit',country:'Denmark'},
    {name:'Mary',country:'England'},
    {name:'Kai',country:'Norway'}
  ];
  $scope.orderByMe = function(x) {
    $scope.myOrderBy = x;
  }
});
</script>
*****
<ul ng-app="myApp" ng-controller="namesCtrl">
  <li ng-repeat="x in names">
    {{x | myFormat}}
  </li>
</ul>

<script>
var app = angular.module('myApp', []);
app.filter('myFormat', function() {
  return function(x) {
    var i, c, txt = "";
    for (i = 0; i < x.length; i++) {
      c = x[i];
      if (i % 2 == 0) {
        c = c.toUpperCase();
      }
      txt += c;
    }
    return txt;
  }
});

```

```
};
});
app.controller('namesCtrl', function($scope) {
  $scope.names = [
    'Jani',
    'Carl',
    'Margareth',
    'Hege',
    'Joe',
    'Gustav',
    'Birgit',
    'Mary',
    'Kai'
  ];
});
```

```
</script>
```

```
*****
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>This header will change after two seconds:</p>
```

```
<h1>{{myHeader}}</h1>
```

```
</div>
```

```
<p>The $timeout service runs a function after a specified number of milliseconds.</p>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('myCtrl', function($scope, $timeout) {
  $scope.myHeader = "Hello World!";
  $timeout(function () {
    $scope.myHeader = "How are you today?";
  }, 2000);
});
</script>
```

RELOGIO - TIME

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>The time is:</p>
```

```
<h1>{{theTime}}</h1>
```

```
</div>
```

```
<p>The $interval service runs a function every specified millisecond.</p>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $interval) {
  $scope.theTime = new Date().toLocaleTimeString();
  $interval(function () {
    $scope.theTime = new Date().toLocaleTimeString();
  }, 1000);
});
</script>
```

HEXADECIMAL VALUE

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>The hexadecimal value of 255 is:</p>
```

```
<h1>{{hex}}</h1>
```

```
</div>
```

```
<p>A custom service with a method that converts a given number into a hexadecimal number.</p>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.service('hexafy', function() {
```

```
  this.myFunc = function (x) {
```

```
    return x.toString(16);
```

```
  }
```

```
});
```

```
app.controller('myCtrl', function($scope, hexafy) {
```

```
  $scope.hex = hexafy.myFunc(255);
```

```
});
```

```
</script>
```

Methods

The example above uses the `.get` method of the `$http` service.

The `.get` method is a shortcut method of the `$http` service. There are several shortcut methods:

`.delete()`

`.get()`

`.head()`

`.jsonp()`

`.patch()`

`.post()`

`.put()`

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>Today's welcome message is:</p>
```

```
<h1>{{myWelcome}}</h1>
```

```
</div>
```

```
<p>The $http service requests a page on the server, and the response is set as the value of the "myWelcome" variable.</p>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('myCtrl', function($scope, $http) {
```

```
  $http({
```

```
    method : "GET",
```

```
    url : "welcome.htm"
```

```
  }).then(function mySuccess(response) {
```

```
    $scope.myWelcome = response.data;
```

```
  }, function myError(response) {
```

```
    $scope.myWelcome = response.statusText;
```

```
});  
});  
</script>
```

Properties

The response from the server is an object with these properties:

- .config the object used to generate the request.
- .data a string, or an object, carrying the response from the server.
- .headers a function to use to get header information.
- .status a number defining the HTTP status.
- .statusText a string defining the HTTP status.

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>Data : {{content}}</p>
```

```
<p>Status : {{statusCode}}</p>
```

```
<p>StatusText : {{statustext}}</p>
```

```
</div>
```

<p>The response object has many properties. This example demonstrate the value of the data, status, and statusText properties.</p>

```
<script>
```

```
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope, $http) {  
  $http.get("welcome.htm")  
  .then(function(response) {  
    $scope.content = response.data;
```

```
    $scope.statuscode = response.status;
    $scope.statustext = response.statusText;
  });
});
</script>
```

ERRO

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<h1>{{content}}</h1>
```

```
</div>
```

```
<p>The $http service executes different functions on success and failure.</p>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $http) {
  $http.get("wrongfilename.htm")
    .then(function(response) {
      $scope.content = response.data;
    }, function(response) {
      $scope.content = "Something went wrong";
    });
});
</script>
```

JSON

The data you get from the response is expected to be in JSON format.

JSON is a great way of transporting data, and it is easy to use within AngularJS, or any other JavaScript.

Example: On the server we have a file that returns a JSON object containing 15 customers, all wrapped in array called records.

customers.php

```
{
  "records": [
    {
      "Name": "Alfreds Futterkiste",
      "City": "Berlin",
      "Country": "Germany"
    },
    {
      "Name": "Ana Trujillo Emparedados y helados",
      "City": "México D.F.",
      "Country": "Mexico"
    },
    {
      "Name": "Antonio Moreno Taquería",
      "City": "México D.F.",
      "Country": "Mexico"
    },
    {
      "Name": "Around the Horn",
      "City": "London",
      "Country": "UK"
    },
    {
```

```
"Name": "B's Beverages",
"City": "London",
"Country": "UK"
},
{
  "Name": "Berglunds snabbköp",
  "City": "Luleå",
  "Country": "Sweden"
},
{
  "Name": "Blauer See Delikatessen",
  "City": "Mannheim",
  "Country": "Germany"
},
{
  "Name": "Blondel père et fils",
  "City": "Strasbourg",
  "Country": "France"
},
{
  "Name": "Bólido Comidas preparadas",
  "City": "Madrid",
  "Country": "Spain"
},
{
  "Name": "Bon app'",
  "City": "Marseille",
  "Country": "France"
},
{
  "Name": "Bottom-Dollar Marketse",
```

```
"City": "Tsawassen",
"Country": "Canada"
},
{
  "Name": "Cactus Comidas para llevar",
  "City": "Buenos Aires",
  "Country": "Argentina"
},
{
  "Name": "Centro comercial Moctezuma",
  "City": "México D.F.",
  "Country": "Mexico"
},
{
  "Name": "Chop-suey Chinese",
  "City": "Bern",
  "Country": "Switzerland"
},
{
  "Name": "Comércio Mineiro",
  "City": "São Paulo",
  "Country": "Brazil"
}
]
}
```

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<ul>
```

```
<li ng-repeat="x in myData">
```

```
  {{ x.Name + ', ' + x.Country }}
```

```
</li>
</ul>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("customers.php").then(function (response) {
    $scope.myData = response.data.records;
  });
});
```

```
</script>
```

```
*****
```

AngularJS Tables

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("customers.php")
```

```
.then(function (response) {$scope.names = response.data.records;});
});
</script>
*****
<style>
table, th , td {
    border: 1px solid grey;
    border-collapse: collapse;
    padding: 5px;
}
table tr:nth-child(odd) {
    background-color: #f1f1f1;
}
table tr:nth-child(even) {
    background-color: #ffffff;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="customersCtrl">

<table>
<tr ng-repeat="x in names">
<td>{{ x.Name }}</td>
<td>{{ x.Country }}</td>
</tr>
</table>

</div>
```

```

<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
    $http.get("customers.php")
        .then(function (response) {$scope.names = response.data.records;});
});
</script>
*****

<style>
table, th , td {
    border: 1px solid grey;
    border-collapse: collapse;
    padding: 5px;
}
table tr:nth-child(odd) {
    background-color: #f1f1f1;
}
table tr:nth-child(even) {
    background-color: #ffffff;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="customersCtrl">

<table>
<tr ng-repeat="x in names | orderBy : 'Country'">
    <td>{{ x.Name }}</td>
    <td>{{ x.Country }}</td>
</tr>

```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
    $http.get("customers.php")
```

```
        .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

```
*****
```

```
<style>
```

```
table, th , td {
```

```
    border: 1px solid grey;
```

```
    border-collapse: collapse;
```

```
    padding: 5px;
```

```
}
```

```
table tr:nth-child(odd) {
```

```
    background-color: #f1f1f1;
```

```
}
```

```
table tr:nth-child(even) {
```

```
    background-color: #ffffff;
```

```
}
```

```
</style>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```
<body>
```

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```

<tr ng-repeat="x in names">
  <td>{{ x.Name }}</td>
  <td>{{ x.Country | uppercase }}</td>
</tr>
</table>

</div>

<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("customers.php")
    .then(function (response) {$scope.names = response.data.records;});
});
</script>
*****

<style>
table, th , td {
  border: 1px solid grey;
  border-collapse: collapse;
  padding: 5px;
}
table tr:nth-child(odd) {
  background-color: #f1f1f1;
}
table tr:nth-child(even) {
  background-color: #ffffff;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

```



```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ $index + 1 }}</td>
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
    $http.get("customers.php")
```

```
        .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td ng-if="$odd" style="background-color:#f1f1f1">
```

```
    {{ x.Name }}</td>
```

```
<td ng-if="$even">
```

```
    {{ x.Name }}</td>
```

```
<td ng-if="$odd" style="background-color:#f1f1f1">
```

```
    {{ x.Country }}</td>
```

```
<td ng-if="$seven">
  {{ x.Country }}</td>
</tr>
</table>
```

```
</div>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("customers.php")
    .then(function (response) {$scope.names = response.data.records;});
});
</script>
```

```
*****
```

```
DROPBOX - SELECT - COMBOBOX
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>Select a car:</p>
```

```
<select ng-model="selectedCar" ng-options="y.brand for (x, y) in cars">
</select>
```

```
<h1>You selected: {{selectedCar.brand}}</h1>
```

```
<h2>Model: {{selectedCar.model}}</h2>
```

```
<h3>Color: {{selectedCar.color}}</h3>
```

```
<p>The visible text inside the dropdown list can also be a property of the value object.</p>
```

```
</div>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.cars = {
        car01 : {brand : "Ford", model : "Mustang", color : "red"},
        car02 : {brand : "Fiat", model : "500", color : "white"},
        car03 : {brand : "Volvo", model : "XC90", color : "black"}
    }
});
</script>
```

```
*****
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>Select a car:</p>
```

```
<select ng-model="selectedCar" ng-options="x for (x, y) in cars">
</select>
```

```
<h1>You selected: {{selectedCar}}</h1>
```

```
</div>
```

<p>This example demonstrates the use of an object as the data source when creating a dropdown list.</p>

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.cars = {
        car01 : "Ford",
```

```

        car02 : "Fiat",
        car03 : "Volvo"
    }
});
</script>
*****

<div ng-app="myApp" ng-controller="myCtrl">

<p>Select a car:</p>

<select ng-model="selectedCar" ng-options="x.model for x in cars">
</select>

<h1>You selected: {{selectedCar.model}}</h1>
<p>Its color is: {{selectedCar.color}}</p>

</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.cars = [
        {model : "Ford Mustang", color : "red"},
        {model : "Fiat 500", color : "white"},
        {model : "Volvo XC90", color : "black"}
    ];
});
</script>
*****

```

MYSQL E PHP

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
    $http.get("customers_mysql.php")
```

```
        .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

```
*****
```

```
<?php
```

```
header("Access-Control-Allow-Origin: *");
```

```
header("Content-Type: application/json; charset=UTF-8");
```

```
$conn = new mysqli("myServer", "myUser", "myPassword", "Northwind");
```

```
$result = $conn->query("SELECT CompanyName, City, Country FROM Customers");
```

```
$outp = "";
```

```
while($rs = $result->fetch_array(MYSQLI_ASSOC)) {
```

```
    if ($outp != "") {$outp .= ",";}
```

```
    $outp .= '{"Name":"' . $rs["CompanyName"] . "'';
```

```
    $outp .= '"City":"' . $rs["City"] . "'';
```

```
    $outp .= '"Country":"' . $rs["Country"] . "'}';
```

```
}
```

```
$outp = '{"records":[' . $outp . ']'}';
```

```
$conn->close();
```

```
echo($outp);
```

```
?>
```

```
*****
```

```
*****
```

```
<style>
```

```
table, th, td {
```

```
    border: 1px solid grey;
```

```
    border-collapse: collapse;
```

```
    padding: 5px;
```

```
}
```

```
table tr:nth-child(odd) {
```

```
    background-color: #f1f1f1;
```

```
}
```

```
table tr:nth-child(even) {
```

```
    background-color: #ffffff;
```

```
}
```

```
</style>
```

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
    $http.get("customers_sql.aspx")
```

```
        .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

```
*****
```

```
<?php
```

```
header("Access-Control-Allow-Origin: *");
```

```
header("Content-Type: application/json; charset=ISO-8859-1");
```

```
$conn = new COM("ADODB.Connection");
```

```
$conn->open("PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=Northwind.mdb");
```

```
$rs = $conn->execute("SELECT CompanyName, City, Country FROM Customers");
```

```
$outp = "";
```

```

while (!$rs->EOF) {
    if ($outp != "") {$outp .= ",";}
    $outp .= '{"Name":"' . $rs["CompanyName"] . "',';
    $outp .= "City":"' . $rs["City"] . "',';
    $outp .= "Country":"' . $rs["Country"] . "'}';
    $rs->MoveNext();
}
$outp = '{"records":[' . $outp . ']'}';

$conn->close();

echo ($outp);
?>
*****

@{
    Response.AppendHeader("Access-Control-Allow-Origin", "*")
    Response.AppendHeader("Content-type", "application/json")
    var db = Database.Open("Northwind");
    var query = db.Query("SELECT CompanyName, City, Country FROM Customers");
    var outp = ""
    var c = chr(34)
}
@foreach(var row in query){
    if (outp != "") {outp = outp + ","}
    outp = outp + "{" + c + "Name" + c + ":" + c + @row.CompanyName + c + ","
    outp = outp + c + "City" + c + ":" + c + @row.City + c + ","
    outp = outp + c + "Country" + c + ":" + c + @row.Country + c + "}"
}
outp = "{" + c + "records" + c + ":[ " + outp + "]"
@outp

```

checkbox

```
<div ng-app="" ng-init="mySwitch=true">
<p>
<button ng-disabled="mySwitch">Click Me!</button>
</p>
<p>
<input type="checkbox" ng-model="mySwitch"/>Button
</p>
<p>
{{ mySwitch }}
</p>
</div>
```

EXIBIR OU NAO UMA DIV

```
<div ng-app="">
<p ng-show="true">I am visible.</p>
<p ng-show="false">I am not visible.</p>
</div>
```

```
<div ng-app="" ng-init="hour=13">
<p ng-show="hour > 18">I am visible.</p>
```

</div>

<div ng-app="">

<p ng-hide="true">I am not visible.</p>

<p ng-hide="false">I am visible.</p>

</div>

AngularJS Events

You can add AngularJS event listeners to your HTML elements by using one or more of these directives:

ng-blur

ng-change

ng-click

ng-copy

ng-cut

ng-dblclick

ng-focus

ng-keydown

ng-keypress

ng-keyup

ng-mousedown

ng-mouseenter

ng-mouseleave
ng-mousemove
ng-mouseover
ng-mouseup
ng-paste

Mouse Events

Mouse events occur when the cursor moves over an element, in this order:

ng-mouseover
ng-mouseenter
ng-mousemove
ng-mouseleave

Or when a mouse button is clicked on an element, in this order:

ng-mousedown
ng-mouseup
ng-click

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<h1 ng-mousemove="count = count + 1">Mouse Over Me!</h1>
```

```
<h2>{{ count }}</h2>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
    $scope.count = 0;  
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<button ng-click="count = count + 1">Click Me!</button>
```

```
<p>{{ count }}</p>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('myCtrl', function($scope) {
```

```
    $scope.count = 0;
```

```
});
```

```
</script>
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<button ng-click="myFunc()">Click Me!</button>
```

```
<div ng-show="showMe">
```

```
    <h1>Menu:</h1>
```

```
    <div>Pizza</div>
```

```
    <div>Pasta</div>
```

```
    <div>Pesce</div>
```

```
</div>
```

```
</div>
```

```
<script>
```

```

var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.showMe = false;
    $scope.myFunc = function() {
        $scope.showMe = !$scope.showMe;
    }
});
</script>

```

```

<div ng-app="myApp" ng-controller="myCtrl">

<h1 ng-mousemove="myFunc($event)">Mouse Over Me!</h1>

<p>Coordinates: {{x + ', ' + y}}</p>

```

```

</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.myFunc = function(myE) {
        $scope.x = myE.clientX;
        $scope.y = myE.clientY;
    }
});
</script>

```

AngularJS Forms

```

<div ng-app="myApp" ng-controller="formCtrl">
<form>
    First Name: <input type="text" ng-model="firstname">

```

```
</form>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('formCtrl', function($scope) {
```

```
    $scope.firstname = "John";
```

```
});
```

```
</script>
```

```
*****
```

```
<div ng-app="">
```

```
<form>
```

```
    First Name: <input type="text" ng-model="firstname">
```

```
</form>
```

```
<h1>You entered: {{firstname}}</h1>
```

```
</div>
```

```
*****
```

```
<div ng-app="">
```

```
<form>
```

```
    Check to show a header:
```

```
    <input type="checkbox" ng-model="myVar">
```

```
</form>
```

```
<h1 ng-show="myVar">My Header</h1>
```

```
</div>
```

```
*****
```

```
<form>
```

```
    Pick a topic:
```

```
<input type="radio" ng-model="myVar" value="dogs">Dogs
```

```
<input type="radio" ng-model="myVar" value="tuts">Tutorials
```

```
<input type="radio" ng-model="myVar" value="cars">Cars
```

```
</form>
```

```
<div ng-switch="myVar">
```

```
  <div ng-switch-when="dogs">
```

```
    <h1>Dogs</h1>
```

```
    <p>Welcome to a world of dogs.</p>
```

```
  </div>
```

```
  <div ng-switch-when="tuts">
```

```
    <h1>Tutorials</h1>
```

```
    <p>Learn from examples.</p>
```

```
  </div>
```

```
  <div ng-switch-when="cars">
```

```
    <h1>Cars</h1>
```

```
    <p>Read about cars.</p>
```

```
  </div>
```

```
</div>
```

```
*****
```

```
<body ng-app="">
```

```
<form>
```

```
  Select a topic:
```

```
  <select ng-model="myVar">
```

```
    <option value="">
```

```
    <option value="dogs">Dogs
```

```
    <option value="tuts">Tutorials
```

```
    <option value="cars">Cars
```

```
  </select>
```

```
</form>
```

```
<div ng-switch="myVar">
```

```
  <div ng-switch-when="dogs">
```

```

    <h1>Dogs</h1>

    <p>Welcome to a world of dogs.</p>
</div>
<div ng-switch-when="tuts">
    <h1>Tutorials</h1>
    <p>Learn from examples.</p>
</div>
<div ng-switch-when="cars">
    <h1>Cars</h1>
    <p>Read about cars.</p>
</div>
</div>
*****
<div ng-app="myApp" ng-controller="formCtrl">
    <form novalidate>
        First Name:<br>
        <input type="text" ng-model="user.firstName"><br>
        Last Name:<br>
        <input type="text" ng-model="user.lastName">
        <br><br>
        <button ng-click="reset()">RESET</button>
    </form>
    <p>form = {{user}}</p>
    <p>master = {{master}}</p>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('formCtrl', function($scope) {
    $scope.master = {firstName:"John", lastName:"Doe"};
    $scope.reset = function() {

```



```
    $scope.user = angular.copy($scope.master);  
  };  
  $scope.reset();  
});  
</script>
```

AngularJS Form Validation

Required

Use the HTML5 attribute required to specify that the input field must be filled out:

```
<body ng-app="">
```

```
<p>Try writing in the input field:</p>
```

```
<form name="myForm">  
<input name="myInput" ng-model="myInput" required>  
</form>
```

```
<p>The input's valid state is:</p>
```

```
<h1>{{myForm.myInput.$valid}}</h1>
```

E-mail

Use the HTML5 type email to specify that the value must be an e-mail:

```
<body ng-app="">
```

```
<p>Try writing an E-mail address in the input field:</p>
```

```
<form name="myForm">  
<input type="email" name="myInput" ng-model="myInput">  
</form>
```

<p>The input's valid state is:</p>

<h1>{{myForm.myInput.\$valid}}</h1>

Form State and Input State

AngularJS is constantly updating the state of both the form and the input fields.

Input fields have the following states:

\$untouched The field has not been touched yet

\$touched The field has been touched

\$pristine The field has not been modified yet

\$dirty The field has been modified

\$invalid The field content is not valid

\$valid The field content is valid

They are all properties of the input field, and are either true or false.

Forms have the following states:

\$pristine No fields have been modified yet

\$dirty One or more have been modified

\$invalid The form content is not valid

\$valid The form content is valid

\$submitted The form is submitted

They are all properties of the form, and are either true or false.

<body ng-app="">

<p>Try leaving the first input field blank:</p>

<form name="myForm">

<p>Name:

```
<input name="myName" ng-model="myName" required>
<span ng-show="myForm.myName.$touched && myForm.myName.$invalid">The name is
required.</span>
</p>

<p>Address:
<input name="myAddress" ng-model="myAddress" required>
</p>

</form>
*****
```

CSS Classes

AngularJS adds CSS classes to forms and input fields depending on their states.

The following classes are added to, or removed from, input fields:

ng-untouched The field has not been touched yet

ng-touched The field has been touched

ng-pristine The field has not been modified yet

ng-dirty The field has been modified

ng-valid The field content is valid

ng-invalid The field content is not valid

ng-valid-key One key for each validation. Example: ng-valid-required, useful when there are more than one thing that must be validated

ng-invalid-key Example: ng-invalid-required

The following classes are added to, or removed from, forms:

ng-pristine No fields has not been modified yet

ng-dirty One or more fields has been modified

ng-valid The form content is valid

ng-invalid The form content is not valid

ng-valid-key One key for each validation. Example: ng-valid-required, useful when there are more than one thing that must be validated

ng-invalid-key Example: ng-invalid-required

The classes are removed if the value they represent is false.

```
<style>
input.ng-invalid {
  background-color:pink;
}
input.ng-valid {
  background-color:lightgreen;
}
</style>
<body ng-app="">

<p>Try writing in the input field:</p>

<form name="myForm">
<input name="myName" ng-model="myName" required>
</form>
*****
<style>
form.ng-pristine {
  background-color:lightblue;
}
form.ng-dirty {
  background-color:pink;
}
</style>
<body ng-app="">
```

```
<form name="myForm">
<p>Try writing in the input field:</p>

<input name="myName" ng-model="myName" required>
```

```
<p>The form gets a "ng-dirty" class when the form has been modified, and will therefore turn pink.</p>
```

```
</form>
```

```
*****
```

Custom Validation

```
<body ng-app="myApp">
```

```
<p>Try writing in the input field:</p>
```

```
<form name="myForm">
<input name="myInput" ng-model="myInput" required my-directive>
</form>
```

```
<p>The input's valid state is:</p>
```

```
<h1>{{myForm.myInput.$valid}}</h1>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.directive('myDirective', function() {
  return {
    require: 'ngModel',
    link: function(scope, element, attr, mCtrl) {
      function myValidation(value) {
        if (value.indexOf("e") > -1) {
```

```
        mCtrl.$setValidity('charE', true);
    } else {
        mCtrl.$setValidity('charE', false);
    }
    return value;
}
mCtrl.$parsers.push(myValidation);
}
};
});
```

</script>

```
<form ng-app="myApp" ng-controller="validateCtrl"
name="myForm" novalidate>
```

```
<p>Username:<br>
```

```
<input type="text" name="user" ng-model="user" required>
```

```
<span style="color:red" ng-show="myForm.user.$dirty && myForm.user.$invalid">
```

```
<span ng-show="myForm.user.$error.required">Username is required.</span>
```

```
</span>
```

```
</p>
```

```
<p>Email:<br>
```

```
<input type="email" name="email" ng-model="email" required>
```

```
<span style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">
```

```
<span ng-show="myForm.email.$error.required">Email is required.</span>
```

```
<span ng-show="myForm.email.$error.email">Invalid email address.</span>
```

```
</span>
```

```
</p>
```

```
<p>
```

```
<input type="submit"
ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||
myForm.email.$dirty && myForm.email.$invalid">
</p>
```

```
</form>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('validateCtrl', function($scope) {
    $scope.user = 'John Doe';
    $scope.email = 'john.doe@gmail.com';
});
```

```
</script>
```

```
*****
```

```
*****
```

AngularJS Global API

isString

isNumber

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<p>{{ x1 }}</p>
```

```
<p>{{ x2 }}</p>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.x1 = "JOHN";
```

```
    $scope.x2 = angular.lowercase($scope.x1);
  });
</script>
```

```
<div ng-app="myApp" ng-controller="myCtrl">
  <p>{{ x1 }}</p>
  <p>{{ x2 }}</p>
</div>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.x1 = "JOHN";
  $scope.x2 = angular.isString($scope.x1);
});
</script>
```

```
<div ng-app="myApp" ng-controller="myCtrl">
  <p>{{ x1 }}</p>
  <p>{{ x2 }}</p>
</div>
```

```
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.x1 = "JOHN";
  $scope.x2 = angular.isNumber($scope.x1);
});
</script>
```

AngularJS and W3.CSS

```
<!DOCTYPE html>
```

```
<html>
```

```
<link rel="stylesheet" href="/w3css/4/w3.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```
<body ng-app="myApp" ng-controller="userCtrl">
```

```
<div class="w3-container">
```

```
<h3>Users</h3>
```

```
<table class="w3-table w3-bordered w3-striped">
```

```
<tr>
```

```
<th>Edit</th>
```

```
<th>First Name</th>
```

```
<th>Last Name</th>
```

```
</tr>
```

```
<tr ng-repeat="user in users">
```

```
<td>
```

```
<button class="w3-btn w3-ripple" ng-click="editUser(user.id)">&#9998; Edit</button>
```

```
</td>
```

```
<td>{{ user.fName }}</td>
```

```
<td>{{ user.lName }}</td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<button class="w3-btn w3-green w3-ripple" ng-click="editUser('new')">&#9998; Create New User</button>
```

```
<form ng-hide="hideform">
  <h3 ng-show="edit">Create New User:</h3>
  <h3 ng-hide="edit">Edit User:</h3>
  <label>First Name:</label>
  <input class="w3-input w3-border" type="text" ng-model="fName" ng-disabled="!edit"
placeholder="First Name">
  <br>
  <label>Last Name:</label>
  <input class="w3-input w3-border" type="text" ng-model="lName" ng-disabled="!edit"
placeholder="Last Name">
  <br>
  <label>Password:</label>
  <input class="w3-input w3-border" type="password" ng-model="passw1"
placeholder="Password">
  <br>
  <label>Repeat:</label>
  <input class="w3-input w3-border" type="password" ng-model="passw2"
placeholder="Repeat Password">
  <br>
  <button class="w3-btn w3-green w3-ripple" ng-disabled="error || incomplete">&#10004;
Save Changes</button>
</form>

</div>

<script src="myUsers.js"></script>

</body>
</html>
```

myUsers.js

```
angular.module('myApp', []).controller('userCtrl', function($scope) {  
  $scope.fName = "";  
  $scope.lName = "";  
  $scope.passw1 = "";  
  $scope.passw2 = "";  
  $scope.users = [  
    {id:1, fName:'Hege', lName:"Pege" },  
    {id:2, fName:'Kim', lName:"Pim" },  
    {id:3, fName:'Sal', lName:"Smith" },  
    {id:4, fName:'Jack', lName:"Jones" },  
    {id:5, fName:'John', lName:"Doe" },  
    {id:6, fName:'Peter',lName:"Pan" }  
  ];  
  $scope.edit = true;  
  $scope.error = false;  
  $scope.incomplete = false;  
  $scope.hideform = true;  
  $scope.editUser = function(id) {  
    $scope.hideform = false;  
    if (id == 'new') {  
      $scope.edit = true;  
      $scope.incomplete = true;  
      $scope.fName = "";  
      $scope.lName = "";  
    } else {  
      $scope.edit = false;  
      $scope.fName = $scope.users[id-1].fName;  
      $scope.lName = $scope.users[id-1].lName;  
    }  
  };  
});
```

```
$scope.$watch('passw1',function() {$scope.test();});
$scope.$watch('passw2',function() {$scope.test();});
$scope.$watch('fName', function() {$scope.test();});
$scope.$watch('lName', function() {$scope.test();});
```

```
$scope.test = function() {
  if ($scope.passw1 !== $scope.passw2) {
    $scope.error = true;
  } else {
    $scope.error = false;
  }
  $scope.incomplete = false;
  if ($scope.edit && (!$scope.fName.length ||
    !$scope.lName.length ||
    !$scope.passw1.length || !$scope.passw2.length)) {
    $scope.incomplete = true;
  }
};

});
```

AngularJS Includes

```
<body ng-app="">
```

```
<div ng-include=""myFile.htm"></div>
```

```
</body>
```

AngularJS Animations

```
<style>
```

```
div {
```

```
  transition: all linear 0.5s;
```

```
  background-color: lightblue;
```

```
  height: 100px;
```

```
  width: 100%;
```

```
  position: relative;
```

```
  top: 0;
```

```
  left: 0;
```

```
}
```

```
.ng-hide {
```

```
  height: 0;
```

```
  width: 0;
```

```
  background-color: transparent;
```

```
  top: -200px;
```

```
  left: 200px;
```

```
}
```

```
</style>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-  
animate.js"></script>
```

```
<body ng-app="myApp">
```

```
<h1>Hide the DIV: <input type="checkbox" ng-model="myCheck"></h1>
```

```
<div ng-hide="myCheck"></div>
```

```
<script>
```

```
var app = angular.module('myApp', ['ngAnimate']);
```

```
</script>
```

```
*****
```

```
*****
```

AngularJS Routing

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-route.js"></script>
```

```
<body ng-app="myApp">
```

```
<p><a href="#/!">Main</a></p>
```

```
<a href="#!red">Red</a>
```

```
<a href="#!green">Green</a>
```

```
<a href="#!blue">Blue</a>
```

```
<div ng-view></div>
```

```
<script>
```

```
var app = angular.module("myApp", ["ngRoute"]);
```

```
app.config(function($routeProvider) {
```

```
    $routeProvider
```

```
    .when("/", {
```

```
        templateUrl : "main.htm"
```

```
    })
```

```
    .when("/red", {
```

```
        templateUrl : "red.htm"
```

```

    })
    .when("/green", {
        templateUrl : "green.htm"
    })
    .when("/blue", {
        templateUrl : "blue.htm"
    });
});
</script>
*****

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-route.js"></script>

<body ng-app="myApp">

<p><a href="#/!">Main</a></p>

<a href="#!london">City 1</a>
<a href="#!paris">City 2</a>

<p>Click on the links to read about London and Paris.</p>

<div ng-view></div>

<script>
var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {
    $routeProvider
    .when("/", {
        templateUrl : "main.htm"
    })

```

```

.when("/london", {
    templateUrl : "london.htm"
})
.when("/paris", {
    templateUrl : "paris.htm"
});
});
</script>
*****
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-route.js"></script>

<body ng-app="myApp">

<p><a href="#/!">Main</a></p>

<a href="#!london">City 1</a>
<a href="#!paris">City 2</a>

<p>Click on the links.</p>

<p>Note that each "view" has its own controller which each gives the "msg" variable a
value.</p>

<div ng-view></div>

<script>
var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {
    $routeProvider
    .when("/", {

```



```

        templateUrl : "main.htm",
    })
    .when("/london", {
        templateUrl : "london.htm",
        controller : "londonCtrl"
    })
    .when("/paris", {
        templateUrl : "paris.htm",
        controller : "parisCtrl"
    });
});
app.controller("londonCtrl", function ($scope) {
    $scope.msg = "I love London";
});
app.controller("parisCtrl", function ($scope) {
    $scope.msg = "I love Paris";
});

```

</script>

Make a Shopping List

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<script>

var app = angular.module("myShoppingList", []);

app.controller("myCtrl", function(\$scope) {

\$scope.products = ["Milk", "Bread", "Cheese"];

```

$scope.addItem = function () {
    $scope.errortext = "";
    if (!$scope.addMe) {return;}
    if ($scope.products.indexOf($scope.addMe) == -1) {
        $scope.products.push($scope.addMe);
    } else {
        $scope.errortext = "The item is already in your shopping list.";
    }
}
$scope.removeItem = function (x) {
    $scope.errortext = "";
    $scope.products.splice(x, 1);
}
});
</script>

<div ng-app="myShoppingList" ng-controller="myCtrl">
    <ul>
        <li ng-repeat="x in products">{{x}}<span ng-click="removeItem($index)">x</span></li>
    </ul>
    <input ng-model="addMe">
    <button ng-click="addItem()">Add</button>
    <p>{{errortext}}</p>
</div>

*****

*****

<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

```

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
```

```
<body>
```

```
<script>
```

```
var app = angular.module("myShoppingList", []);
```

```
app.controller("myCtrl", function($scope) {
```

```
    $scope.products = ["Milk", "Bread", "Cheese"];
```

```
    $scope.addItem = function () {
```

```
        $scope.errortext = "";
```

```
        if (!$scope.addMe) {return;} 
```

```
        if ($scope.products.indexOf($scope.addMe) == -1) {
```

```
            $scope.products.push($scope.addMe);
```

```
        } else {
```

```
            $scope.errortext = "The item is already in your shopping list.";
```

```
        }
```

```
    }
```

```
    $scope.removeItem = function (x) {
```

```
        $scope.errortext = "";
```

```
        $scope.products.splice(x, 1);
```

```
    }
```

```
});
```

```
</script>
```

```
<div ng-app="myShoppingList" ng-cloak ng-controller="myCtrl" class="w3-card-2 w3-margin" style="max-width:400px;">
```

```
    <header class="w3-container w3-light-grey w3-padding-16">
```

```
        <h3>My Shopping List</h3>
```

```
    </header>
```

```
    <ul class="w3-ul">
```

```
        <li ng-repeat="x in products" class="w3-padding-16">{{x}}<span ng-click="removeItem($index)" style="cursor:pointer;" class="w3-right w3-margin-right">x</span></li>
```

```

</ul>
<div class="w3-container w3-light-grey w3-padding-16">
  <div class="w3-row w3-margin-top">
    <div class="w3-col s10">
      <input placeholder="Add shopping items here" ng-model="addMe" class="w3-input w3-
border w3-padding">
    </div>
    <div class="w3-col s2">
      <button ng-click="addItem()" class="w3-btn w3-padding w3-green">Add</button>
    </div>
  </div>
  <p class="w3-text-red">{{errortext}}</p>
</div>
</div>

</body>
</html>

```

```
*****
```

```
SELECT - DROPDOWBOX
```

```
*****
```

```

<div ng-app="myApp" ng-controller="myCtrl">

  <select ng-model="selectedName" ng-options="x for x in names">

  </select>

</div>

<script>

```

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.names = ["Emil", "Tobias", "Linus"];
});
</script>
```

TABELA

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
  $http.get("customers.php")
```

```
  .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

HTTP BUSCAR DADOS

```
<div ng-app="myApp" ng-controller="customersCtrl">
```

```
<table>
```

```
<tr ng-repeat="x in names">
```

```
<td>{{ x.Name }}</td>
```

```
<td>{{ x.Country }}</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('customersCtrl', function($scope, $http) {
```

```
  $http.get("customers_sql.aspx")
```

```
  .then(function (response) {$scope.names = response.data.records;});
```

```
});
```

```
</script>
```

FORM

```
<form>
```

Check to show a header:

```
<input type="checkbox" ng-model="myVar">
```

```
</form>
```

```
<h1 ng-show="myVar">My Header</h1>
```

```
*****
```

```
FORM 2
```

```
*****
```

```
<div ng-app="myApp" ng-controller="formCtrl">  
  <form novalidate>  
    First Name:<br>  
    <input type="text" ng-model="user.firstName"><br>  
    Last Name:<br>  
    <input type="text" ng-model="user.lastName">  
    <br><br>  
    <button ng-click="reset()">RESET</button>  
  </form>  
  <p>form = {{user}}</p>  
  <p>master = {{master}}</p>  
</div>
```

```
<script>  
var app = angular.module('myApp', []);  
app.controller('formCtrl', function($scope) {  
  $scope.master = {firstName: "John", lastName: "Doe"};  
  $scope.reset = function() {  
    $scope.user = angular.copy($scope.master);  
  };  
  $scope.reset();  
});
```

```
</script>
```

```
*****
```

```
UL LI
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
<ul>
```

```
<li ng-repeat="x in names">{{x}}</li>
```

```
</ul>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('myCtrl', function($scope) {
```

```
  $scope.names = ["Emil", "Tobias", "Linus"];
```

```
});
```

```
</script>
```

```
*****
```

```
ARQUIVO CONTROLER
```

```
*****
```

```
<div ng-app="myApp" ng-controller="personCtrl">
```

```
First Name: <input type="text" ng-model="firstName"><br>
```



```
Last Name: <input type="text" ng-model="lastName"><br>
```

```
<br>
```

```
Full Name: {{fullName()}}
```

```
</div>
```

```
<script src="personController.js"></script>
```

```
angular.module('myApp', []).controller('namesCtrl', function($scope) {
```

```
  $scope.names = [
```

```
    {name:'Jani',country:'Norway'},
```

```
    {name:'Hege',country:'Sweden'},
```

```
    {name:'Kai',country:'Denmark'}
```

```
  ];
```

```
});
```

```
*****
```

```
ON CLIQUE
```

```
*****
```

```
<div ng-app="myApp" ng-controller="myCtrl">
```

```
  <h1 ng-click="changeName()">{{firstname}}</h1>
```

```
</div>
```

```
<script>
```

```
var app = angular.module('myApp', []);
```

```
app.controller('myCtrl', function($scope) {
```

```
$scope.firstname = "John";
$scope.changeName = function() {
  $scope.firstname = "Nelly";
}
});
</script>
```

TABELA ORDEM

```
<div ng-app="myApp" ng-controller="namesCtrl">
```

```
<table border="1" width="100%">
  <tr>
    <th ng-click="orderByMe('name')">Name</th>
    <th ng-click="orderByMe('country')">Country</th>
  </tr>
  <tr ng-repeat="x in names | orderBy:myOrderBy">
    <td>{{x.name}}</td>
    <td>{{x.country}}</td>
  </tr>
</table>
```

```
</div>
```

```
<script>
angular.module('myApp', []).controller('namesCtrl', function($scope) {
  $scope.names = [
    {name:'Jani',country:'Norway'},
```

```
{name:'Carl',country:'Sweden'},
{name:'Margareth',country:'England'},
{name:'Hege',country:'Norway'},
{name:'Joe',country:'Denmark'},
{name:'Gustav',country:'Sweden'},
{name:'Birgit',country:'Denmark'},
{name:'Mary',country:'England'},
{name:'Kai',country:'Norway'}
];
$scope.orderByMe = function(x) {
  $scope.myOrderBy = x;
}
});
</script>
```

FOR

```
<ul ng-app="myApp" ng-controller="namesCtrl">
  <li ng-repeat="x in names">
    {{x | myFormat}}
  </li>
</ul>
```

```
<script>
var app = angular.module('myApp', []);
app.filter('myFormat', function() {
  return function(x) {
    var i, c, txt = "";
```

```
for (i = 0; i < x.length; i++) {  
    c = x[i];  
    if (i % 2 == 0) {  
        c = c.toUpperCase();  
    }  
    txt += c;  
}  
return txt;  
};  
});  
app.controller('namesCtrl', function($scope) {  
    $scope.names = ['Jani', 'Carl', 'Margareth', 'Hege', 'Joe', 'Gustav', 'Birgit', 'Mary', 'Kai'];  
});  
</script>
```

CALCULO

```
<div ng-app="" ng-init="quantity=1;cost=5">  
<p>Total in dollar: {{ quantity * cost }}</p>  
</div>
```