



CFITECH

Web 2025-2026 / angular



Vitor (Prof) 22-05-26 13:59
@Aimable (Prof) @Web Avancé



ismadmh 22-05-26 15:01
ng serve -o



NisRin 22-05-26 15:03

```
ng new angularTest-app
// créer un nouveau dossier angular
ng g c header
// générer un component ex. Header
```

(edited)



Aimable (Prof) 28-05-26 08:55



[Angular_chp1-4.pdf](#)
5.49 MB



1



Aimable (Prof) 28-05-26 11:54

```
<h1>{{titre}}</h1>
<div class="x"><img [src]="imageUrl" alt=""></div>
<a [href]="linkSite" target="_blank">{{linkText}}</a>
<button (click)="changeTitre()" [disabled]="isDisabled">Change</button>
<input type="checkbox" [checked]="isChecked">
```

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

@Component({
  selector: 'app-header',
  imports: [],
  templateUrl: './header.html',
  styleUrls: ['./header.css'],
})
export class Header {
  titre="cfitech";
  changeTitre(){
    this.titre="KOEKELTECH";
  }
  imageUrl="https://png.pngtree.com/png-clipart/20191027/ourmid/pngtree-view-icon-png-image_1869829.jpg "
```

```
linkSite="https://cfitech.be/"
linkText="site CFITECH"

isDisabled=false;
isChecked=true;
}
```



Chapitre_5-une_partie.pdf

561.1 KB



Aimable (Prof) 29-05-26 15:30



Two_way_binding.pdf

372.43 KB



1



Aimable (Prof) 04-06-26 08:55

Exercice 5 — Event binding (click)

Créez un bouton qui déclenche la méthode saluer() du composant. Cette méthode doit afficher console.log("Bonjour !").



Aimable (Prof) 04-06-26 09:13

Exercice 1 — Interpolation simple

Dans le composant, une propriété message = "Bonjour Jean" est définie. Affichez ce message dans une balise <h2>

Exercice 2 — Interpolation avec expression

Une variable age = 25 est déclarée dans le composant. Affichez dans une balise <p> : Vous avez 25 ans. (le nombre doit venir de la variable)

Exercice 3 — Property binding (src)

Une variable

logo = "<https://angular.io/assets/images/logos/angular/angular.png>" est définie. Affichez l'image à l'aide de property binding.



Aimable (Prof) 04-06-26 09:22

Exercice 4 — Property binding (disabled)

Une variable isDisabled = true est définie. Utilisez [disabled] pour désactiver un bouton quand cette valeur est vraie.



Aimable (Prof) 04-06-26 09:47

Exercice 6 — Event binding + Interpolation

Une variable message = "". En cliquant sur un bouton, changez-la en "Bienvenue !", et affichez-la.

Exercice 7 — Two-Way Binding (ngModel)

Une variable nom = ". Faites un champ texte lié à nom, et affichez en direct : Bonjour [nom].

Exercice 8 — Two-Way Binding avec checkbox

Une variable accept = false. Liez-la à une case à cocher, et affichez "Accepté" ou "Refusé"

selon l'état.



Aimable (Prof) 04-06-26 10:32

Exercice 9 — Application des 4 types ensemble

Une variable nom = ". Affichez un bouton "Continuer" uniquement si le champ nom est rempli.

Créez un mini-formulaire avec :

- un champ texte lié à prenom
- un bouton pour valider (désactivé si le champ est vide)
- un message Bonjour [prenom] après clic

Exercice 10 — Interpolation + expression conditionnelle

Déclarer une variable score représentant la note obtenue par l'apprenant.

Si le score est supérieur ou égal à 10, le programme doit afficher :

"Réussi"

Sinon, le programme doit afficher :

"Échoué"

Exercice 11 — Property Binding avec input

Déclarez une variable JavaScript readOnlyMode qui peut être soit true, soit false.

Utilisez cette variable pour contrôler dynamiquement l'attribut readonly d'un champ <input> HTML.

Exercice 12 — Two-way Binding avec transformation

Liez un input à ville, et affichez-la en majuscules à côté.

Exercice 13 — Bouton activé uniquement si case cochée

Une case à cocher liée à conditionsAcceptees. Le bouton "Continuer" doit être activé seulement si true



ismadmh 04-06-26 10:50

```
nom3 = 'Ali';  
changerNom() {  
  this.nom3 = 'Sara';  
}  
}
```

```
<button (click)="changerNom()" >Changer Nom</button>  
<p>Nom: {{nom3}} </p>
```

 **ismadmh** | nom3 = 'Ali'; changerNom() { this.nom3 = 'Sara'; } }

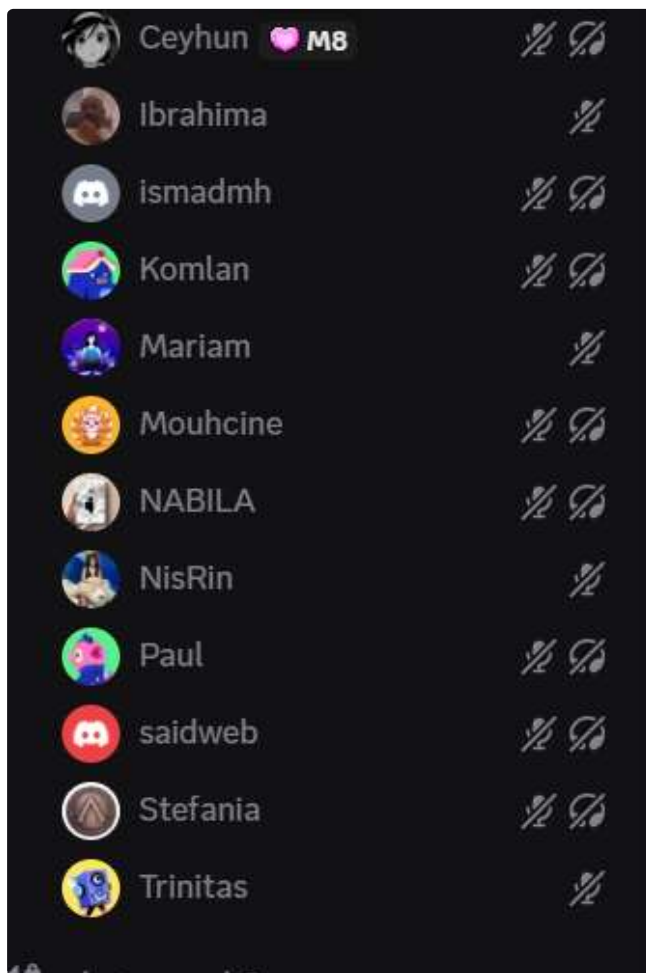


Komlan 04-06-26 10:50

choukrane



Aimable (Prof) 04-06-26 10:52



Aimable (Prof) 05-06-26 09:20



[Les_Signals_dans_Angular.pdf](#)

3.89 MB



1



Aimable (Prof) 12-06-26 16:10



[Chapitre_6_Control_Flow__Directives.pdf](#)

4.73 MB



2



Aimable (Prof) 18-06-26 11:33



[Cours_dangular-debutant_jusquau_18_juin_2026.pdf](#)

6.01 MB



2



Aimable (Prof) 18-06-26 11:52



[Projet_1.pdf](#)

65.66 KB

à presenter jeudi le 25 juin 2026 à la 1ere heure

👍 2



ismadmh 23-06-26 15:38

```
<label for="">Nom:</label>
<input type="text" placeholder="Nom" [(ngModel)]="service.nom" />
<label for="">Prenom:</label>
<input type="text" placeholder="Prenom" [(ngModel)]="service.prenom" />
<label for="">Email:</label>
<input type="email" placeholder="Nom" [(ngModel)]="service.email" />
<label for="">Formation:</label>
<input type="text" placeholder="Formation" [(ngModel)]="service.formation" />
```

(edited)



ismadmh 23-06-26 15:45

```
<p>Nom: {{service.nom}}</p>
<p>Premom: {{service.prenom}}</p>
<p>Email: {{service.email}}</p>
<p>Formation: {{service.formation}}</p>
```

(edited)



Aimable (Prof) 26-06-26 11:02



Services_et_Signal.pdf

186.58 KB



Aimable (Prof) 02-07-26 11:06

Exercice :

Une clinique médicale souhaite développer une application Angular permettant de gérer le dossier d'un patient.

Lorsqu'un patient arrive à l'accueil, ses informations sont enregistrées. Le médecin et le secrétariat doivent ensuite consulter ces informations sans avoir à les ressaisir.

Pour cela, l'application utilisera un service afin de partager les données entre plusieurs composants.

Travail demandé

Créer une application contenant trois composants :

- AccueilPatient
- DossierPatient
- Dashboard

Créer un service :

ng generate service services/patient

Le service devra contenir :

- nom ;
- prénom ;
- âge ;

- maladie.

Le composant AccueilPatient permettra de saisir les informations.

Le composant DossierPatient affichera le dossier médical.

Le composant Dashboard affichera un résumé du patient.



Aimable (Prof) 02-07-26 11:45

```
header.css
.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;

  padding: 20px;
  background: #1e293b;
}

.logo {
  color: white;
  font-size: 24px;
  font-weight: bold;
}

.menu {
  display: flex;
  gap: 20px;
}

.menu a {
  color: white;
  text-decoration: none;
}

button {
  padding: 10px 15px;
  cursor: pointer;
}
```



Aimable (Prof) 03-07-26 13:20

```
. app.css
.light {
  background-color: white;
  color: black;
  min-height: 100vh;

  transition: 0.3s;
}

.dark {
  background-color: #121212;
  color: white;
  min-height: 100vh;
}
```

```
transition: 0.3s;
}
```



Aimable (Prof) 03-07-26 14:08



[Services_et_Signal.pdf](#)

547.79 KB



Aimable (Prof) 06-07-26 10:34



[Modifier__Rechercher.pdf](#)

269.94 KB



Aimable (Prof) 10-07-26 09:30



[Cours_dangular-debutant.pdf](#)

7.09 MB



Aimable (Prof) 10-07-26 14:59



[Template_driven_forms.pdf](#)

181.63 KB



Aimable (Prof) 14-07-26 13:57

exercice:

Ex concret : formulaire d'inscription

Imagine que tu développes une page d'inscription.

L'utilisateur doit remplir :

- Nom
- Email
- Mot de passe

Visuellement :

Nom : _

Email : _

Mot de passe: _

[S'inscrire]

Ici :

- Nom → un FormControl
- Email → un FormControl
- Mot de passe → un FormControl

L'ensemble constitue un seul FormGroup.

On peut le représenter ainsi :

FormGroup (inscription)

|

|--- FormControl (nom)

- └─ FormControl (email)
- └─ FormControl (motDePasse)



Aimable (Prof) 14-07-26 14:30



Aimable (Prof) 14-07-26 15:20



Reactive_Forms.pdf

234.36 KB



ismadmh 16-07-26 13:40

<https://angular.dev/guide/templates/pipes>

Pipes • Angular

The web development framework for building modern apps.



Cédric 16-07-26 13:56

Nom : Angular 17 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS & Flex Layout

ID : Mikael.Angular-BeastCode

Description : 258 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ngRx, RxJS, PWA & Testing)

Version : 17.0.4

Serveur de publication : Mikael Morlund

Lien de la marketplace pour VS : <https://marketplace.visualstudio.com/items?itemName=Mikael.Angular-BeastCode>

Angular 17 Snippets - TypeScript, Html, Angular Material, ngRx, RxJ...

Extension for Visual Studio Code - 258 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ngRx, RxJS, PWA & Testing)





Aimable (Prof) 16-07-26 14:18



message.txt

6.43 KB



Aimable (Prof) 17-07-26 12:17



Pipes.pdf

664.79 KB



Aimable (Prof) 17-07-26 13:13

TP 1 – Application de gestion des étudiants

Vous développez une application de gestion d'une école.

Pour rendre l'interface plus agréable, le directeur souhaite que chaque nom d'étudiant soit précédé du texte Étudiant :.

Au lieu d'écrire ce texte partout dans le HTML, vous décidez de créer un Pipe personnalisé.

Travail demandé

Créer un Pipe nommé :

etudiant

Ce Pipe devra transformer :

Ahmed

en

Étudiant : Ahmed

Données

nom1 = "Ahmed";

nom2 = "Sara";

nom3 = "Ali";

Affichage attendu :

Étudiant : Ahmed

Étudiant : Sara

Étudiant : Ali



Aimable (Prof) 17-07-26 13:35

TP 2 – Application de gestion des employés

Une entreprise souhaite afficher le poste de chaque employé avant son nom.

Le responsable informatique vous demande de créer un Pipe réutilisable.

Travail demandé

Créer un Pipe nommé :

fonction

Le Pipe recevra un paramètre représentant la fonction.

Exemple

```
{{ nom | fonction:'Développeur' }}
```

Affichage attendu:

Développeur : Ahmed

Tester également avec :

Comptable

Chef de projet

Designer

Technicien

TP 3 – Plateforme E-commerce

Vous développez un site de vente en ligne.

Le client souhaite afficher tous les prix suivis du symbole €.

Pour éviter de répéter ce traitement dans tous les composants, vous décidez de créer un Pipe.

Travail demandé

Créer un Pipe nommé :
prix

Données

ordinateur = 1200;

telephone = 850;

tablette = 499;

Affichage attendu

1200 €

850 €

499 €

TP 4 – Gestion des notes

Une école souhaite afficher automatiquement la mention correspondant à la note obtenue par chaque étudiant.

Travail demandé

Créer un Pipe nommé :
mention

Règles

Note Mention

≥16 Très Bien

≥14 Bien

≥12 Assez Bien

≥10 Passable

<10 Échec

Données

18

15

13

11

8

Affichage attendu :

Très Bien

Bien

Assez Bien

Passable

Échec



Aimable (Prof) 17-07-26 14:00

TP 5 – Banque

Une banque souhaite masquer une partie du numéro de compte affiché à l'écran afin de protéger les données des clients.

Seuls les quatre derniers chiffres doivent rester visibles.

Travail demandé

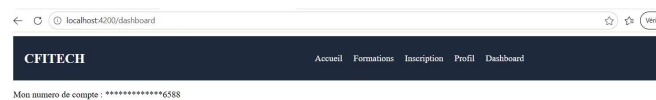
Créer un Pipe nommé :
compte

Données

BE68539007547034

Affichage attendu :

****7034**



Aimable (Prof) 17-07-26 14:51

TP 6 – Application RH :

Le service des ressources humaines souhaite afficher le statut des employés en fonction de leur âge.

Travail demandé
Créer un Pipe nommé :
age

Règles
Âge Résultat
moins de 18 Mineur
de 18 à 59 Adulte
60 et plus Senior

Affichage attendu

16 → Mineur

28 → Adulte

65 → Senior



Aimable (Prof) 17-07-26 14:59

TP 7 – Gestion de stock

Une entreprise vend du matériel informatique.

Le responsable souhaite afficher automatiquement l'état du stock.

Travail demandé
Créer un Pipe nommé :
stock

Règles
Quantité Affichage
0 Rupture de stock
1 à 5 Stock faible
>5 Disponible

Données
20

4

0

Affichage attendu
Disponible

Stock faible

Rupture de stock

TP 8 – Gestion des commandes

Une plateforme de commerce électronique souhaite informer le client de l'état de sa commande.

Travail demandé
Créer un Pipe nommé :
commande

Règles
Valeur Résultat
0 En préparation
1 Expédiée
2 Livrée
3 Annulée

Affichage attendu

0

↓

En préparation



Aimable (Prof) 17-07-26 15:13

TP 9 – Gestion des abonnements

Une plateforme de streaming souhaite afficher le type d'abonnement de ses utilisateurs.

Travail demandé
Créer un Pipe nommé :
abonnement

Règles
Valeur Résultat
B Basic
S Standard
P Premium

Affichage attendu

B

↓

Basic



Aimable (Prof) 17-07-26 15:21

TP 10 – Hôpital

Le service des urgences souhaite afficher le niveau de priorité d'un patient selon un score de gravité.

Travail demandé
Créer un Pipe nommé :
priorite

Règles

Score Priorité

≥90 Urgence absolue

≥70 Urgence élevée

≥40 Urgence normale

<40 Faible priorité

TP 11 – Gestion des salaires

Le directeur souhaite savoir quels employés recevront une prime annuelle.

Travail demandé

Créer un Pipe nommé :
prime

Règles

Salaire Résultat

≥4000 € Prime accordée

<4000 € Pas de prime

Affichage attendu

3500 €

↓

Pas de prime

5200 €

↓

Prime accordée

TP 12 – Plateforme de formation

Une plateforme d'apprentissage souhaite afficher le niveau d'un étudiant selon son pourcentage de progression.

Travail demandé

Créer un Pipe nommé :
niveau

Règles

Progression Résultat

100 % Formation terminée

≥75 % Niveau avancé

≥50 % Niveau intermédiaire

<50 % Niveau débutant

Projet Final :

Gestion des employés d'une entreprise

Vous êtes développeur Angular dans une entreprise spécialisée dans les ressources humaines.

Votre mission consiste à développer une page permettant d'afficher les informations des employés de manière claire et professionnelle.

Pour éviter de répéter les mêmes traitements dans plusieurs composants, votre responsable vous demande de créer plusieurs Pipes personnalisés.

Données

employees = [

```
{  
  nom:"Ahmed",  
  age:35,  
  salaire:4500,  
  telephone:"0499123456",  
  email:"ahmed@gmail.com",  
  note:18,  
  stockConge:24  
},
```

```
{  
  nom:"Sara",  
  age:22,  
  salaire:2800,  
  telephone:"0488456789",  
  email:"sara@hotmail.com",  
  note:13,  
  stockConge:5  
},
```

```
{  
  nom:"Ali",  
  age:17,  
  salaire:1800,  
  telephone:"0477123456",  
  email:"ali@yahoo.com",  
  note:8,  
  stockConge:0  
}
```

```
];
```

Travail demandé

Développez les Pipes suivants :

1. mentionPipe : transforme une note en mention (Très Bien, Bien, Assez Bien, Échec).
2. telephonePipe : masque tous les chiffres sauf les quatre derniers.
3. emailPipe : masque une partie de l'adresse e-mail.
4. primePipe : indique si l'employé bénéficie d'une prime.
5. agePipe : affiche Mineur, Adulte ou Senior.

6. congePipe : affiche :

- o Aucun congé restant si le solde est de 0.
- o Peu de jours restants si le solde est inférieur ou égal à 5.
- o Solde de congés suffisant au-delà de 5 jours.



Cédric 23-07-26 09:52

<https://gugu42.alwaysdata.net/angutech/>



Aimable (Prof) 23-07-26 11:28



Module_3_partie_input.pdf

259.08 KB



Aimable (Prof) 24-07-26 13:54



Partie_2_-_Output.pdf

262.07 KB



Aimable (Prof) 24-07-26 15:57



Cours_angular_niveau_intermediaire.pdf

5.24 MB



Projet_type.pdf

81.07 KB



Aimable (Prof) 25-08-26 10:33



Module_4_httpClient.pdf

4.09 MB



ismadmh 25-08-26 11:32

L'UAA FRONT END SE DÉROULERA LE MARDI 15 SEPTEMBRE 2026



ismadmh 27-08-26 10:42

```
npm install -g json-server
```

```
{
  "formations": [
    {
      "id": "1",
      "title": "Angular",
      "body": "Formation Angular niveau débutant"
    },
    {
      "id": "2",
      "title": "React",
```

```

    "body": "Formation React avec hooks"
  },
  {
    "id": "3",
    "title": "Vue.js",
    "body": "Formation Vue.js 3 et Composition API"
  }
]
}

```

```
json-server --watch db.json --port 3000
```

(edited)



ismadmh 27-08-26 11:40

```

import { Component, inject, signal } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { Formations } from '../services/formations';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, FormsModule],
  templateUrl: './app.html',
  styleUrls: ['./app.css'],
})
export class App {
  service = inject(Formations);
  listeFormations = signal<any[]>([]);
  error = '';
  chargerFormations() {
    this.service.getFormations().subscribe({
      next: (data) => this.listeFormations.set(data),
      error: (err) => (this.error = 'Erreur serveur'),
    });
  }
  ngOnInit() {
    this.chargerFormations();
  }

  titre = '';
  description = '';

  ajouterFormations() {
    const formation = {title: this.titre, body:this.description};
    this.service.addFormations(formation).subscribe({
      next: data => console.log("Formation ajouter"),
      error: err => console.log("Erreur")

    });
  }
}

```

```
import { HttpClient } from '@angular/common/http';
import { inject, Injectable } from '@angular/core';
import { form } from '@angular/forms/signals';

@Injectable({
  providedIn: 'root',
})
export class Formations {
  private http = inject(HttpClient);
  private apiUrl = 'http://localhost:3000/ formations';
  getFormations() {
    return this.http.get<any[]>(this.apiUrl);
  }
  addFormations(formation: any) {
    return this.http.post(this.apiUrl, formation);
  }
}
```

```
@if (error) {
  <p>{{ error }}</p>
} @else {
  @for (formation of listeFormations(); track formation.id) {
    <h1>{{ formation.title }}</h1>
    <p>{{ formation.body }}</p>
  }
}

<input type="text" placeholder="Introduire titre" [(ngModel)]="titre">
<input type="text" placeholder="Introduire description"
[(ngModel)]="description">

<button (click)="ajouterFormations()" >Ajouter</button>
```



ismadmh 28-08-26 13:50

```
#[ORM\Table(name: "users")]
```

au dessus de class User



Cédric 28-08-26 13:53

registerdAt



ismadmh 28-08-26 13:58

```
symfony console doctrine:migrations:diff
```

ensuite recuperer la version et faire

```
symfony console doctrine:migrations:execute version
```

(edited)



Cédric 28-08-26 14:10

symfony console doctrine:migrations:execute --up
"DoctrineMigrations\VERSION_GENERE.php"
symfony console make:controller Api/UserController



Aimable (Prof) 28-08-26 14:47



message.txt

2.97 KB



ismadmh 28-08-26 15:19



text.txt

2.97 KB



Aimable (Prof) 28-08-26 15:55



Creation_dune_application_Web_Angular__Symfony__MySQL....

2.95 MB



2



ismadmh 01-09-26 09:48

<https://github.com/isma-dmh/full-stack>

GitHub - isma-dmh/full-stack

Contribute to isma-dmh/full-stack development by creating an account on GitHub.



Aimable (Prof) 01-09-26 11:01

[

```
{ id: 1, name: 'iPhone 16', price: 999, category: 'Téléphone' },  
{ id: 2, name: 'Samsung Galaxy S25', price: 899, category: 'Téléphone' },  
{ id: 3, name: 'MacBook Air', price: 1299, category: 'Ordinateur' },  
{ id: 4, name: 'Dell XPS', price: 1199, category: 'Ordinateur' },  
{ id: 5, name: 'iPad Air', price: 699, category: 'Tablette' },  
{ id: 6, name: 'Galaxy Tab', price: 599, category: 'Tablette' }  
];
```



Aimable (Prof) 01-09-26 11:26



Cours_angular_niveau_intermediaireAmeliore.pdf

5.79 MB



ismadmh 01-09-26 11:41

Service

```

import { HttpClient } from '@angular/common/http';
import { inject, Injectable } from '@angular/core';
import { Product } from '../models/product';
import { Observable, of } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  private http = inject(HttpClient);

  private products: Product[] = [
    { id: 1, name: 'iPhone 16', price: 999, category: 'Téléphone' },
    { id: 2, name: 'Samsung Galaxy S25', price: 899, category: 'Téléphone' },
    { id: 3, name: 'MacBook Air', price: 1299, category: 'Ordinateur' },
    { id: 4, name: 'Dell XPS', price: 1199, category: 'Ordinateur' },
    { id: 5, name: 'iPad Air', price: 699, category: 'Tablette' },
    { id: 6, name: 'Galaxy Tab', price: 599, category: 'Tablette' },
  ];

  getProducts(): Observable<Product[]> {
    return of(this.products);
  }

  searchProducts(search: string): Observable<Product[]> {
    const result = this.products.filter((product) =>
      product.name.toLowerCase().includes(search.toLowerCase()),
    );

    return of(result);
  }
}

```

Composants

```

import { Component, inject, signal, Signal } from '@angular/core';
import { ProductService } from '../../services/product.service';
import { Product } from '../../models/product';

@Component({
  selector: 'app-produits',
  imports: [],
  templateUrl: './produits.html',
  styleUrls: ['./produits.css'],
})
export class Produits {
  private productService = inject(ProductService);
  products = signal<Product[]>([]);
  loading = signal(true);

  ngOnInit() {
    this.productService.getProducts().subscribe((products) => {
      next: {

```

```

        this.products.set(products);
        this.loading.set(false);
    }
    error: {
        console.log(`Erreur`);
        this.loading.set(false);
    }
  });
}
}

```

Html

```

<h1>Mes produits</h1>

@if (loading()) {
  <p>Chargement</p>
} @else {
  @if (products().length === 0) {
    <p>Aucun produit trouver</p>
  } @else {
    @for (product of products(); track product.id) {
      <div>
        <p>{{ product.name }}</p>
        <p>{{ product.price }}</p>
      </div>
    }
  }
}

```

Copier coller intelligemment on a peut etre pas les meme nom de variable et de fichier



ismadmh 02-09-26 09:24

Voici la db du projet full-stack



school.sql

2.78 KB



1

Exported 94 message(s)

Timezone: UTC+1