Introduction

Angular est :

Un Framework de design d'applications –

Une plateforme de développement permettant la création d'applications single page et sophistiquées.

La plateforme de développement Angular est construite en TypeScript.

Elle inclut:

– Un Framework basé-composant pour la construction d'application web évolutives,

– Une collection de bibliothèques bien intégrées qui couvrent une large variété de fonctionnalités (routage, gestion de formulaires, communication client-serveur, etc.),

– Une combinaison d'outils de développement permettant d'aider aux : développement, build, test, et mise à jour du code.

Programmation orientée composant : Consiste à utiliser une approche modulaire de l'architecture d'un projet informatique permet d'assurer au logiciel une meilleure lisibilité et une meilleure maintenance.

Structure et architecture d'un projet Angular…Le module Un module peut être partagé à d'autres modules Angular possède son propre système de modularité appelé modules angulaires ou NgModules. Chaque application Angular possède au moins une classe de module angulaire: le module racine, appelé classiquement AppModule. Pour Angular un module est une classe avec un décorateur @NgModule.Les décorateurs sont des fonctions qui modifient les classes JavaScript.Angular possède de nombreux décorateurs qui attachent des métadonnées aux classes pour configurer et donner le sens à ces classes.

1- **Routage** (Routing)

Le **routing dans Angular** est un système qui permet de naviguer entre plusieurs pages d'une application sans avoir besoin de recharger complètement le site. En réalité, il ne s'agit pas de vraies pages séparées, mais de composants que l'on affiche ou cache selon l'URL choisie. Par exemple, lorsqu'un utilisateur clique sur "Accueil" ou "Profil", Angular utilise le routing pour afficher le bon contenu de manière rapide et fluide. Cela rend l'application plus agréable à utiliser, comme si c'était une vraie application installée sur l'ordinateur ou le téléphone. Le routing permet aussi d'organiser l'application en différentes sections et d'ajouter des fonctionnalités utiles comme la gestion d'accès (empêcher un utilisateur non connecté d'accéder à certaines pages) ou le chargement progressif des parties de l'application pour gagner en performance.

Contenu de `app-routing.module.ts`

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

const routes: Routes = [];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Afin de permettre aux utilisateurs de garder leurs habitudes de navigation en visitant une Single Page Application, il est nécessaire d'utiliser un système de Routage. Grâce à ce système, les utilisateurs peuvent : utiliser l'historique de leur navigateur (par exemple les boutons back et next), partager des liens, ajouter une vue à leurs

favoris, ouvrir une vue dans une nouvelle fenêtre via le menu contextuel, …

2- **un composant**

Les composants sont au cœur du développement d'applications avec Angular.
Ils constituent les blocs fondamentaux de construction des interfaces utilisateur et définissent la façon dont les données sont présentées et manipulées dans l'application.
c'est une **partie réutilisable de l'interface utilisateur**, qui regroupe **le code (TypeScript)**, **l'affichage (HTML)** et **le style (CSS)** dans une seule unité.

3- **un service**
Un service est une classe TS composée d'attributs et de méthodes, dont l'instanciation est gérée par Angular.
Ils permettent de :

- réutiliser du code entre différents composants
- faciliter l'échange des données
- centraliser les appels de service
- séparer les responsabilités visuelles (component) et fonctionnelles/techniques (service)

Cas d’utilisation

- **Communication avec une API**
- **Communication avec une base de données**
- **Implémentation d’un cache d’objets**
- **Gestion de la session utilisateur côté client**

Voici un exemple de service, qui est capable d'ajouter une tâche ToDo, à une liste de tâches stockée dans le service lui-même.

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

@Injectable({
    providedIn: 'root'
})
export class TodoService {
    todos: Todo[];

    constructor() {}

    create(todo: Todo) {
        this.todos.push(todo);
    }
}
```

4- directives

En Angular, une directive est une instruction qu’on met dans le code HTML pour changer l’apparence ou le comportement des éléments de DOM

Il existe deux sortes de directives :

- **Les directives structurelles : elles ont pour but de modifier le DOM en ajoutant, enlevant ou replaçant un élément du DOM.**

- **Les attribute directives : elles ont pour but de modifier l'apparence ou le comportement d'un élément.**

5-formulaire:

En Angular, un *formulaire* est un ensemble de champs, permettant à l'utilisateur de saisir des données, qui sont ensuite gérées par Angular via un Form API.

Il sert à collecter, valider et envoyer des informations.

La première étape consiste à importer `ReactiveFormsModule`, cela permet d'avoir accès aux directives `formGroup` et `formControlName` et également d'ajouter l'event `ngSubmit` sur le `<form>`.

Ensuite, il faut créer un objet `FormGroup` qui représente le formulaire. Cet objet est composé de plusieurs `FormControl` qui représentent les champs du formulaire avec leurs valeurs initiales et leurs validateurs.

Enfin, il faut lier le `FormGroup` au `<form>` avec la directive `formGroup` et chaque `FormControl` aux champs du formulaire avec la directive `formControlName`.

6-**pipe**

Les pipes sont des fonctions de transformation de données utilisables directement depuis le template afin de transformer les données dans l'affichage . c'est un moyen simple de transformer, formater ou filtrer une valeur dans votre template.

Lorsque vous utilisez un pipe dans un template, vous le faites suivre d'une barre verticale (|) et du nom du pipe. Vous pouvez également passer des arguments aux pipes.

exemple:

{{ "bonjour" | uppercase }}

il met le bonjour en majuscul à l'affichage

conclusion

Angular est aujourd'hui l'un des frameworks les plus puissants et complets pour le développement d'applications web modernes. Grâce à son architecture basée sur les composants, il favorise la modularité, la réutilisation du code et une meilleure maintenance des projets.

Cependant, les services jouent un rôle essentiel dans la gestion de la logique métier et la communication avec les API, séparant ainsi la partie fonctionnelle de la partie visuelle.

En résumé, Angular est bien plus qu'un simple framework : c'est une véritable plateforme de développement complète, permettant de créer des applications single page robustes, performantes et évolutives.

https://blog.codewise.fr/angular-directives

https://apprendre.bonjour-angular.com/cest-quoi/formulaire/

https://apprendre.bonjour-angular.com/cest-quoi/pipe/