Angular permet de créer la partie front end des applications web de type SPA basé sur les composants web

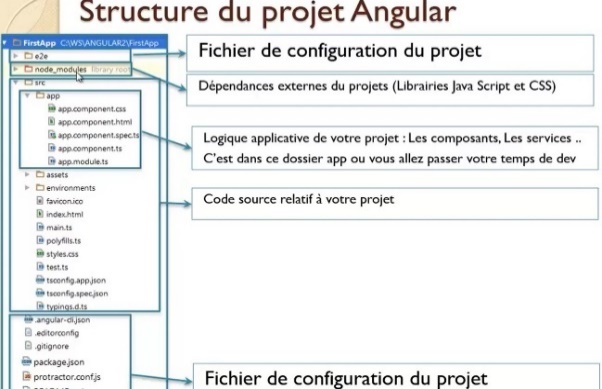
Définition du SPA : Simple Page Application (toute application angular contient une seule page index.html, pas de rechargement dans les pages routées), mise en pratique dans la formation 1 au niveau des routerlink

**Préparation de l’environnement de développement**

Cli : command line interface (ng new, ng serve …) permet de créer, compiler, tester et le déployer un projet angular : installation de @angular/cli contient la logique ng « mot » selon ce que nous voulons faire. npm : node package manager permet de télécharger ou d’inclure des dépendances dans le projet



Création du projet : **ng new nouveau\_projet**



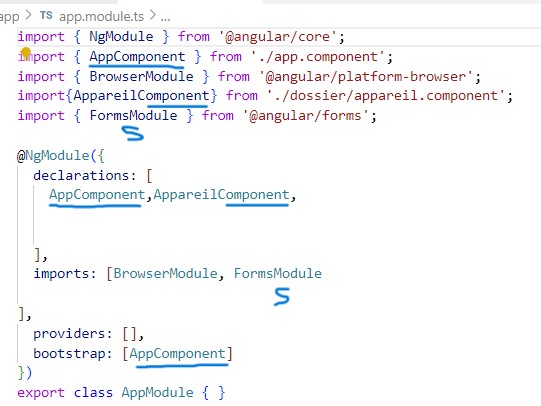
Intallation de style bootstrap dans nouveau\_projet : npm install bootstrap@version –save: Flag(save) permet de l’ajouter au fichier json du projet. Dans angular.json : **“node\_modules/bootstrap/dist/css/bootstrap.css “,**

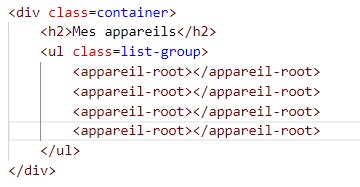
Création d’un component : **ng g c nom\_component**

Décorateur :

**Component** importer depuis @angular/core, **@component** contient trois attributs, le sélecteur est une balise unique liée à un component, l’identifiant de la logique du component

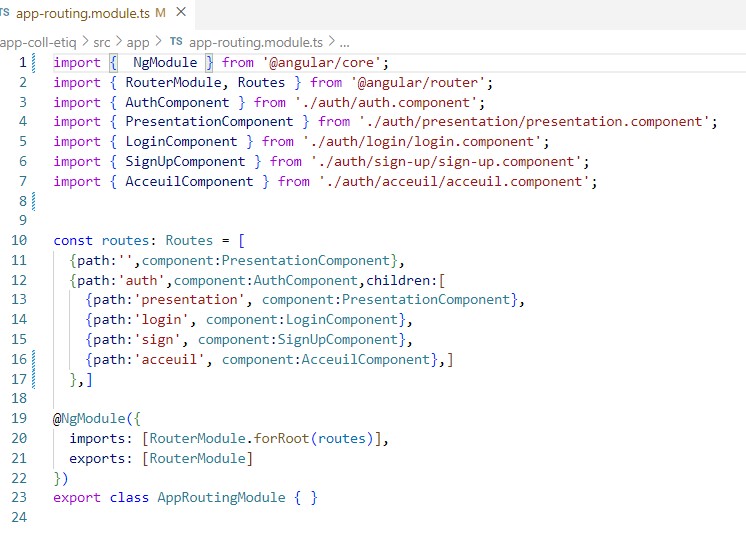


**ngModule** importer depuis @angular/core, **@ngModule** contient 04 attributs : la déclaration : toute composant d’un module doit être déclarer dans le module du composant, Imports : se sont des librairies connues dans angular qu’on import pour un cas d’usage, providers : les services de chaque composant seront déclarés dans ce décorateur, Bootstrap : indique la logique qui doit être afficher 

la balise **appareil-root** geree dans **App.component.html** affiche la logique **AppareilComponent** 

la balise **app-root** affiche la logique AppComponent au navigateur



Le module a été créé dans le dossier **app** qui contient **AppModule**, le module **AuthModule** est un sous module. Création de module : **ng g m nom\_module**

AppRoutingModule qui gère le chemin de l’ensemble des modules, doit être importer

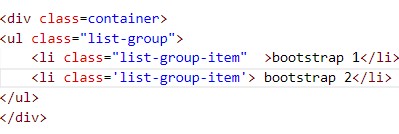
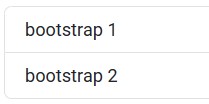


Nous affichons AppModule dans **main.ts** comme le module qui va gerer les autres modules



**Liaison de données :** permet la communication des informations de TS vers le Template

**Code bootstrap** :

**String interpolation** :

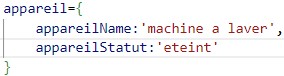
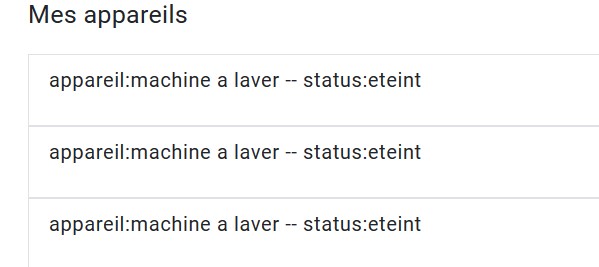
App.component.html

Appareil.component.html

Appareil.component.ts

Objet Rendu html

**Code Bootstrap :**

**Property binding :** on affecte à une propriété une condition



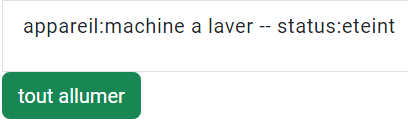
isAuth=false;

<button class="btn btn-primary" [disabled]='!isAuth'>tout allumer </button>

 constructor(){

  setTimeout(()=>{this.isAuth=true}, 5000)

 }



**Event binding :** évènement click, ….

App.component

results='tout alumer !';

<button class="btn btn-primary" (click)=”onToutAllumer()”> tout allumer </button>

  onToutAllumer(){

    console.log('tout est allumer')

  }

onAllumer(){

console.log(this.results);

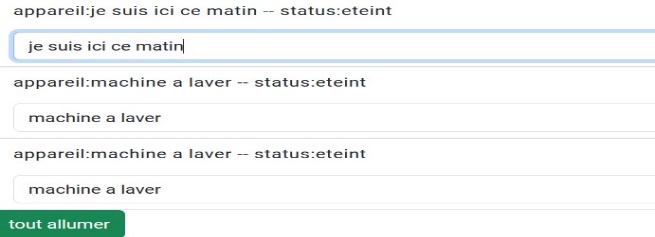
}

**Tow way data binding**

import {FormsModule} from '@angular/forms';

<input[(ngModel)]='Appareils[0].appareilName'>

Appareil.component.html



**Poperty binding personalyzed**

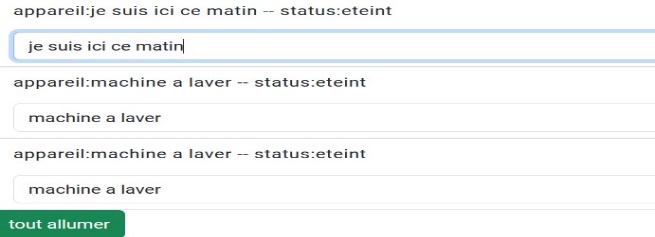
import {Input} from "@angular/core";

App.component

Appareil.component

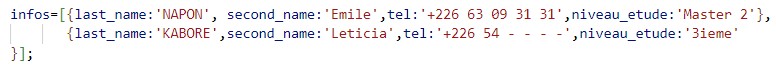
**result**



**Les directives structurelles**

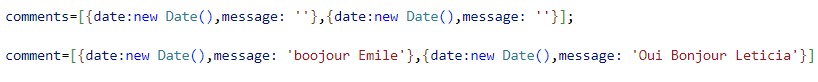
**Contact.component**

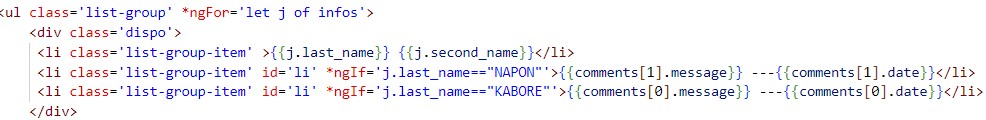
**Objects**

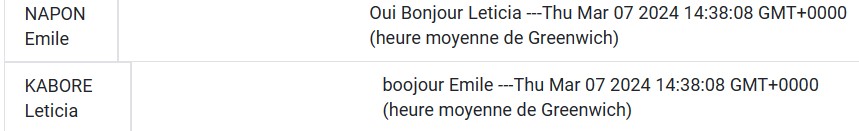


Comments objects

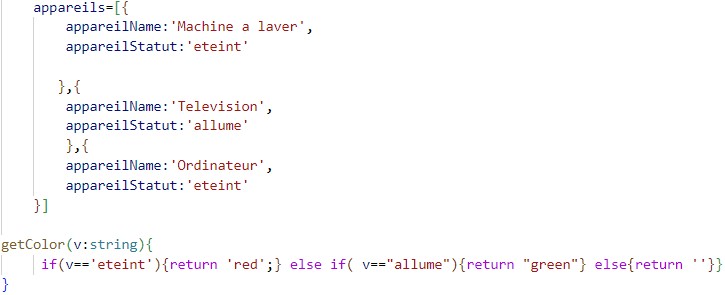






**Directives par attribut : [ngStyle]**

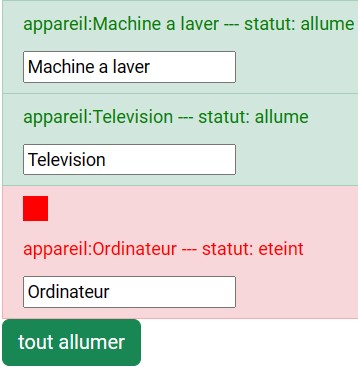
***Vue***

*dom*



**Directives par attribut : [ngClass]**

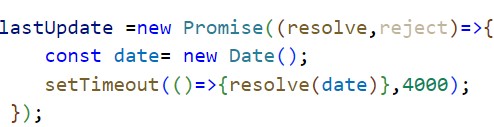
 

**Promesses, pipes**

**Date Pipe**

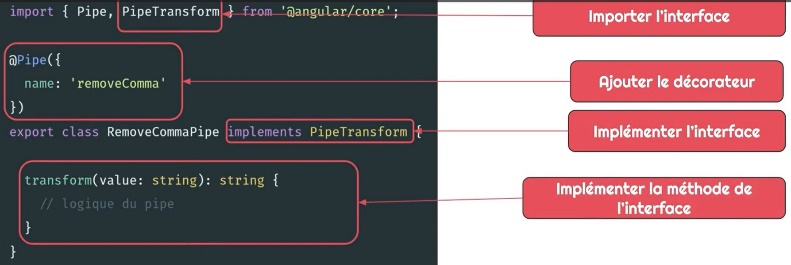
<span [ngStyle]='{color:"red" }'>{{comments[1].date | date}}</span> --->



**Pipe async**

**Personalized Pipe**



Remplace d’une virgule par un point, dans l’affichage du prix dans une application de e-commerce

List of pipes: ngx-pipes sur github

**Services**

Creation du service

Injection du service dans app.modul

L’utilisation du service dans un composant

Les objets sont dans le service

L’implémentation de ngOnInit 🡪 import OnInit dans angular core, qui est une interface

Déclarer un tableau vide dans le composant et utiliser la méthode ngOnInit pour dire que les données dans le service sous forme de tableau correspondant au tableau déclaré dans le component

Ajout des méthodes dans le service

Les méthodes interpolées dans le dom sont les méthodes qui implémentent les méthodes du service