



Visual Studio Code



Firebase



TypeScript

# Cours de Développement Web Avancé



Développement d'applications web  
avec le framework front-end Angular et  
le langage TypeScript



Prof. Ousmane SALL

Professeur Titulaire des Universités CAMES

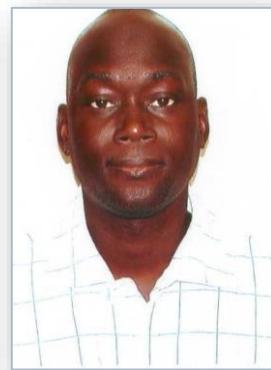
Université de THIES - UFR Sciences et Technologies

Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)



# A propos de moi

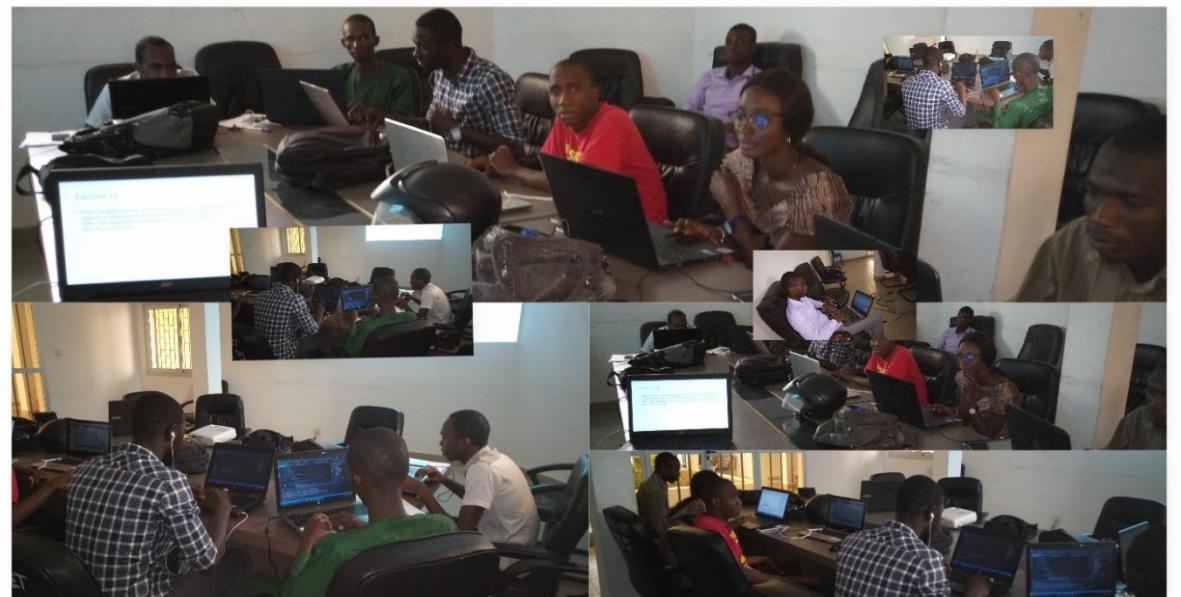


- Enseignant-Chercheur à l'UFR SET- Université de THIES  
<https://sites.google.com/a/univ-thies.sn/osall751/>
- Enseignements:
  - Algorithmique et Programmation(C, Java, PHP)
  - Programmation WEB dynamique(HTML 5 CSS, PHP, MySQL, CMS,...)
  - Programmation Java, Dart, TypeScript
  - Programmation Java, Jakarta EE, JSF, Spring, SpringBoot, Angular
  - Technologies Mobiles Android, Xamarin, Ionic, Flutter
  - Programmation .Net, C#
  - Gestion de Projet Informatique
  - Génie Logiciel, Qualité et Métrique du Logiciel
- Contact:
  - [osall@univ-thies.sn](mailto:osall@univ-thies.sn)
  - UFR SET, Université de THIES -Dpt Informatique, BP 967 THIES.



# Une sagesse chinoise...

*« J'écoute et j'oublie; je lis et je comprends; je fais et j'apprends »*  
[Proverbe chinois]



# Contenu

1. Introduction au Framework Angular et le langage TypeScript
2. Initiation au langage de programmation TypeScript
3. Développement d'applications web avec le Framework Angular
4. Persistance de données et Interaction avec le Cloud
5. Déploiement sur Firebase Hosting



# Description de l'ECUE

- Ce cours porte sur le **développement d'applications web avec le Framework front-end Angular et le langage TypeScript**
- Ce cours s'est inspiré des contenus suivants :
  - La documentation disponible sur TypeScriptLang et Angular.io  
<https://www.typescriptlang.org/docs> et <https://angular.io/docs>
  - Le support de cours de Achref El Mouelhi <http://www.lsis.org/elmouelhia/> sur Angular et TypeScript avec son autorisation
  - Développement Web Front End Avec Angular 4 de Mohamed Youssfi  
<https://www.youtube.com/watch?v=KJEJexcnEQ&list=PLxr551TUsmApkhhzmpn6lVyRGjAC1URrO>
  - Supports disponibles sur le web et Tutoriaux Youtube  
[https://www.youtube.com/results?search\\_query=TypeScript](https://www.youtube.com/results?search_query=TypeScript) et  
[https://www.youtube.com/results?search\\_query=Angular](https://www.youtube.com/results?search_query=Angular)

# Objectifs/Compétences visé(e)s

- **Objectif général** : l'objectif du cours est de maîtriser les concepts, enjeux et les techniques de base de la programmation d'applications web front-end avec le framework Angular.
- **A la fin de ce cours, l'étudiant devra être capable de :**
  - Mettre en œuvre les méthodologies pour la conception et le développement d'applications web front-end ;
  - Exploiter Angular, HTML, Node Package Manager et TypeScript pour le développement d'applications web.
  - Utiliser les concepts avancés de TypeScript dans Angular.
  - Créer des Composants Angular, Modules, Directives et Services.
  - Lier les données aux IU et enregistrer les données dans les bases de données ;
  - Déployer des applications dans Firebase Hosting.

# Evaluation du cours

- Les apprentissages sont évalués par un examen sur table et une évaluation pratique (projet) individuelle à rendre à la fin du cours.
- Le projet consistera au développement d'une solution web avec Angular mettant en œuvre les concepts, normes et méthodologies vus dans le cours relevant des couches présentation, métier et données.
- Ce projet à rendre devra faire l'objet d'un petit rapport dont le gabarit sera donné et une séance de présentation diapositive.
- Chaque projet doit être déployé dans Firebase Hosting

# Prérequis

- Connaissances des notions de base de la Programmation Orientée Objet
- Expérience avec Visual Studio ou un autre EDI
- Connaissance du HTML et des styles CSS, Bootstrap, JavaScript
- Dans la mesure du possible une machine avec au moins **04 Go de RAM** et une connexion internet stable avec un bon débit



# Déroulement prévisionnel du cours(40h)

Séances	Sujets	Durées
Séance n°01	<b>Présentation du cours et Introduction au Framework Angular et le langage TypeScript</b>	02h
Séance n°02	<b>Initiation au langage de programmation TypeScript:</b> Variables, Types de base et Opérateurs, Structures de Contrôle, Fonctions, Tableaux et Collections	03h
Séance n°03	<b>Initiation au langage de programmation TypeScript:</b> Programmation Orientée Objet, Programmation Asynchrone, Espaces de nom et Modules, Décorateurs	01h
Séance n°04	<b>Initiation au langage de programmation TypeScript:</b> TP n°1 sur TypeScript	02h
Séance n°05	<b>Initiation au langage de programmation TypeScript:</b> TP n°1 sur TypeScript	03h
Séance n°06	<b>Développement d'applications web avec le Framework front-end Angular:</b> Composants	02h
Séance n°07	<b>Développement d'applications web avec le Framework front-end Angular:</b> La liaison de données ou Data Binding	
Séance n°08		
Séance n°09		
Séance n°10		
Séance n°11		
Séance n°12		
Séance n°13		

# Webographie

- <https://angular.io/>
- <http://typescriptlang.org>
- <https://www.youtube.com/watch?v=KJEJejxcnEQ&list=PLxr551TUsmAphhzmpn6IVyRGjAC1URrO>
- [https://www.youtube.com/results?search\\_query=TypeScript](https://www.youtube.com/results?search_query=TypeScript)
- [https://www.youtube.com/results?search\\_query=Angular](https://www.youtube.com/results?search_query=Angular)
- Tutos sur Google et Youtube entre autres.



Visual Studio Code



Firebase



TypeScript



# Introduction au Framework Angular et le langage TypeScript



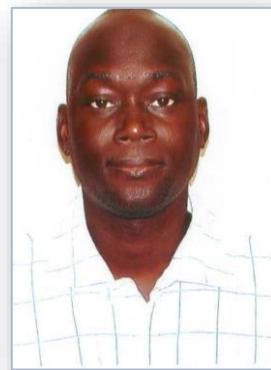
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)



# A propos de moi

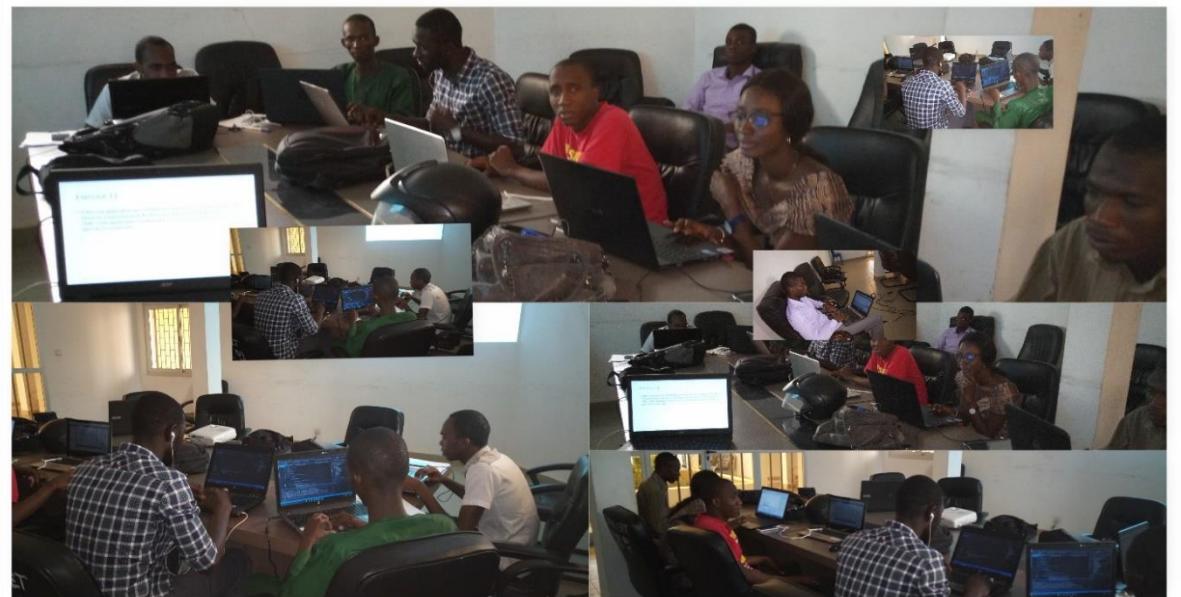


- Enseignant-Chercheur à l'UFR SET- Université de THIES  
<https://sites.google.com/a/univ-thies.sn/osall751/>
- Enseignements:
  - Algorithmique et Programmation(C, Java, PHP)
  - Programmation WEB dynamique(HTML 5 CSS, PHP, MySQL, CMS,...)
  - Programmation Java, Dart, TypeScript
  - Programmation Java, Jakarta EE, JSF, Spring, SpringBoot, Angular
  - Technologies Mobiles Android, Xamarin, Ionic, Flutter
  - Programmation .Net, C#
  - Gestion de Projet Informatique
  - Génie Logiciel, Qualité et Métrique du Logiciel
- Contact:
  - [osall@univ-thies.sn](mailto:osall@univ-thies.sn)
  - UFR SET, Université de THIES -Dpt Informatique, BP 967 THIES.



# Une sagesse chinoise...

*« J'écoute et j'oublie; je lis et je comprends; je fais et j'apprends »*  
[Proverbe chinois]



# Contenu

- 1. Introduction au Framework Angular et le langage TypeScript**
2. Initiation au langage de programmation TypeScript
3. Développement d'applications web avec le Framework Angular
4. Persistance de données et Interaction avec le Cloud
5. Déploiement sur Firebase Hosting



# Framework Logiciel

- Un **Framework** est une boite à outils pour un développeur web.  
Frame signifie **cadre** et **work** se traduit par travail.  
Un **Framework** contient des composants autonomes qui permettent de faciliter le développement d'un site web ou d'une application.
- Un ensemble cohérent de composants logiciels structurels. Sert à créer les **fondations** ainsi que les **grandes lignes** de tout ou d'une partie d'un logiciel(**Architecture**)
- Par exemple, un framework peut inclure des classes et des fonctions prédéfinies qui peuvent être utilisées pour traiter les entrées, gérer les périphériques matériels et interagir avec le logiciel système.

# Front End vs. Back End



# Front End vs. Back End

- **Front End**

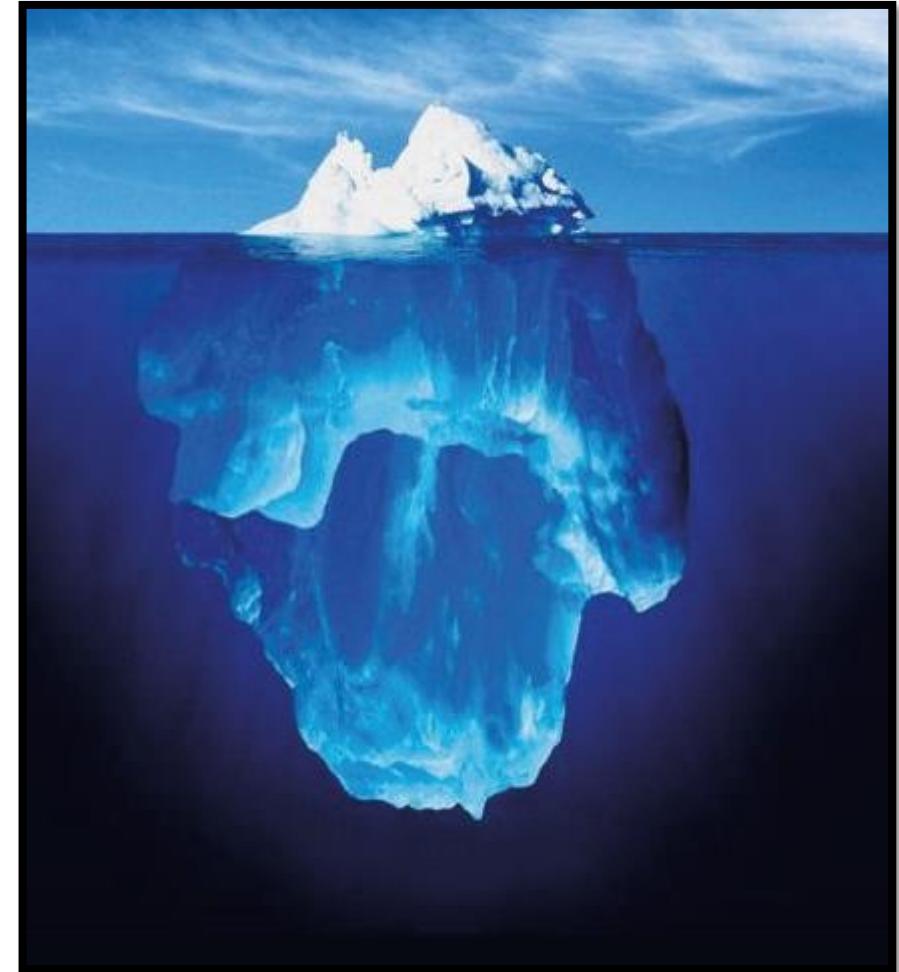
- Codes exécutés sur la machine client (votre) (ordinateur, tablette, téléphone portable, etc.)
- Tous les navigateurs (Firefox, Chrome, ...) exécutent du code front-end
- Ont de nombreuses restrictions de sécurité (heureusement)
- Traite généralement la conception de pages, les graphiques, l'interaction utilisateur et les programmes côté client
- **HTML, CSS, JavaScript, Angular, JQuery, FLASH,**

- **Back End**

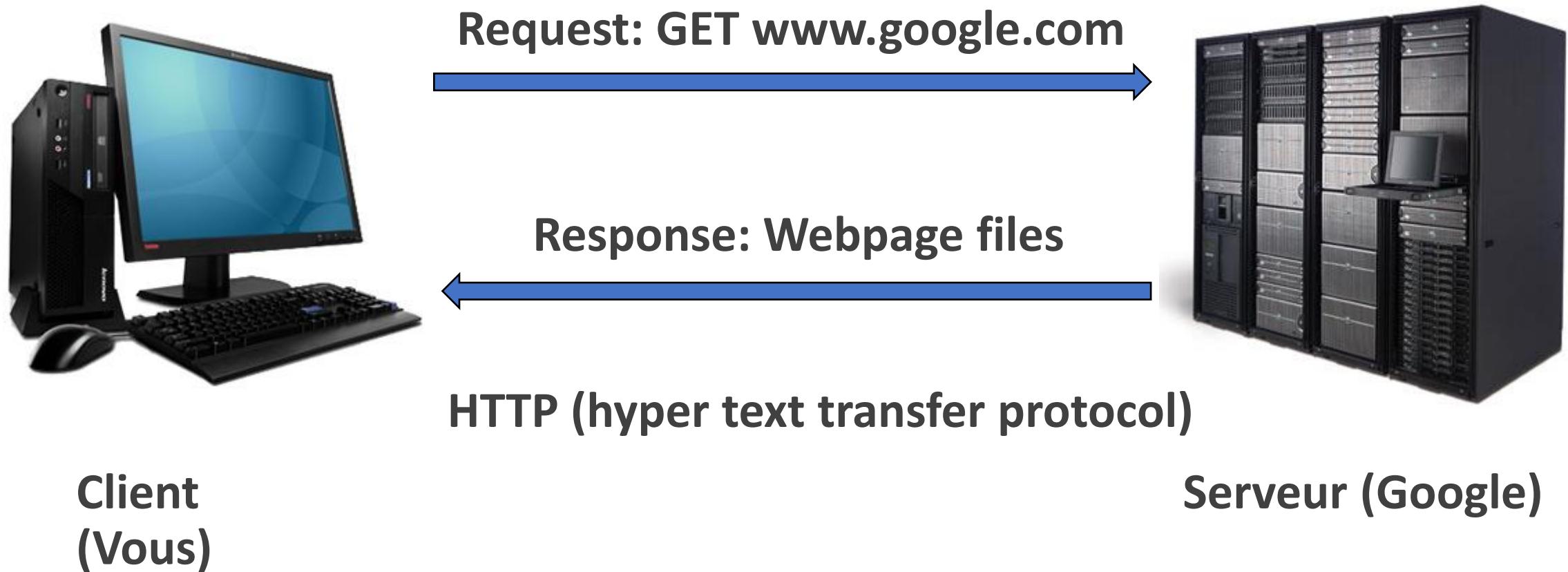
- Codes exécutés sur la machine serveur
- Génère dynamiquement le contenu des pages Web
- Traite les bases de données, les fichiers,...
- **C#, ASP.NET, PHP, JSP, Web Services, Node.js, SQL, JAVA, PYTHON**

# Front End vs. Back End

- **Pont entre le Front End et le Back End**
  - JSON
  - XML
  - MVC
  - Web Forms
  - AJAX



# Accéder à un site Web



# Qu'est ce que Angular ?

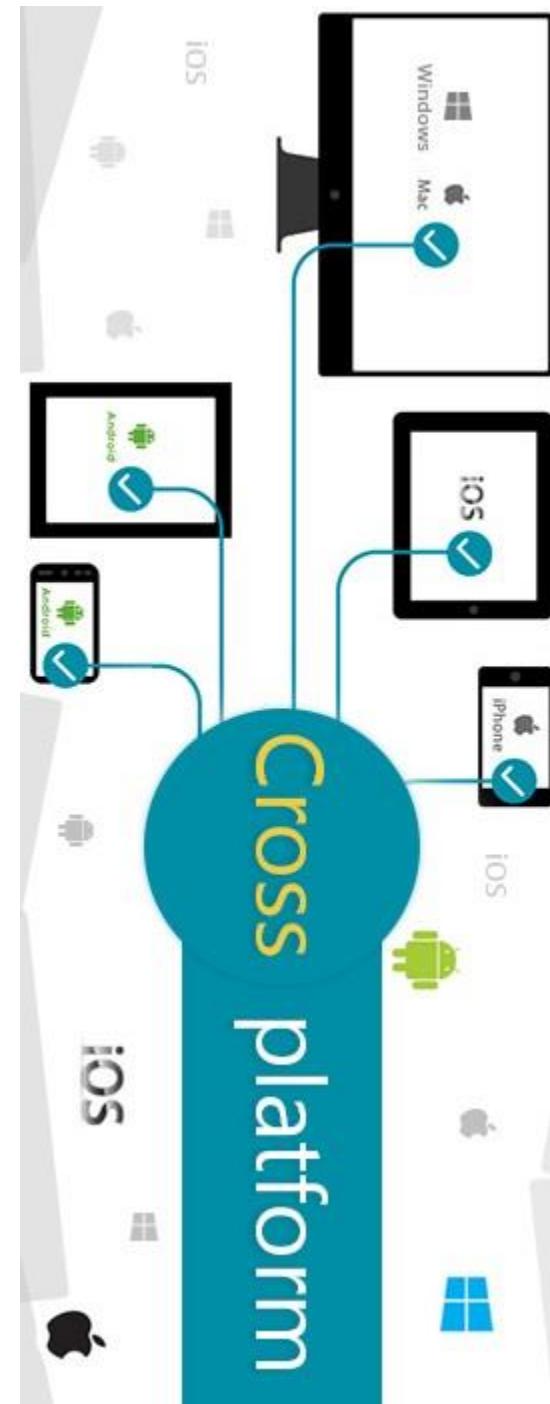
- Framework côté client open-source basé sur TypeScript pour le développement d'applications web. Angular est une réécriture d'AngularJS
- Permettant de créer des applications **Front-End**, mono-page(**SPA** Single Page Application) - Une application qui contient une seule page HTML (index.html) récupérée du serveur utilisant les **composants web**
- Basé sur un concept connu en programmation orientée objet **Injection de dépendance** : consistant à utiliser des classes sans faire de l'instanciation statique
- Langages utilisés par **Angular**: **HTML** pour les vues, **CSS** pour les styles, **TypeScript** pour le code métier(scripts) depuis la version 2



<b>Developer(s)</b>	Google
<b>Initial release</b>	2.0 / 14 September 2016; 4 years ago <sup>[1]</sup>
<b>Stable release</b>	10.1.5 / 7 October 2020; 4 days ago <sup>[2]</sup>
<b>Preview release</b>	11.0.0-next.5 / 7 October 2020; 4 days ago <sup>[2]</sup>
<b>Repository</b>	<a href="#">Angular Repository</a> ↗
<b>Written in</b>	TypeScript
<b>Platform</b>	Web platform
<b>Type</b>	Web framework
<b>License</b>	MIT License
<b>Website</b>	<a href="#">angular.io</a> ↗

# Qu'est ce que Angular ?

- Multiplateforme(Mobile, Desktop, Web)
- Maintenu par Google
- Communauté Active:
  - <https://gitter.im/angular/angular>
  - <https://angular-2-training-book.rangle.io/>
  - <https://www.youtube.com/channel/UC3cEGKhg3OERn-ihVsJcb7A>
  - <https://angular.io/docs>
  - <http://www.tutorialspoint.com/angularjs/index.htm>
  - <http://stackoverflow.com/questions/tagged/angular2>
  - <https://www.reddit.com/r/angular2/>
  - <https://devchat.tv/adventures-in-angular>
- Typescript



# Versions d'Angular

- Angular 1 (ou AngularJS) présenté en 2009 : utilisant le JavaScript
- Angular 2 ou Angular présenté en octobre 2014 : remplacement du JavaScript par TypeScript
- Angular 4 présenté en décembre 2016
- Angular 5 présenté en novembre 2017
- Angular 6 présenté en mai 2018
- Angular 7 présenté en octobre 2018
- Angular 8 présenté en mai 2019
- Angular 9 présenté en février 2020
- Angular 10 présenté en juin 2020



# Versions d'Angular

- Mise à jour majeure tous les 06 mois
- 01 à 03 mises à jour majeures pour chaque release majeures
- Correctifs chaque semaine



# TypeScript

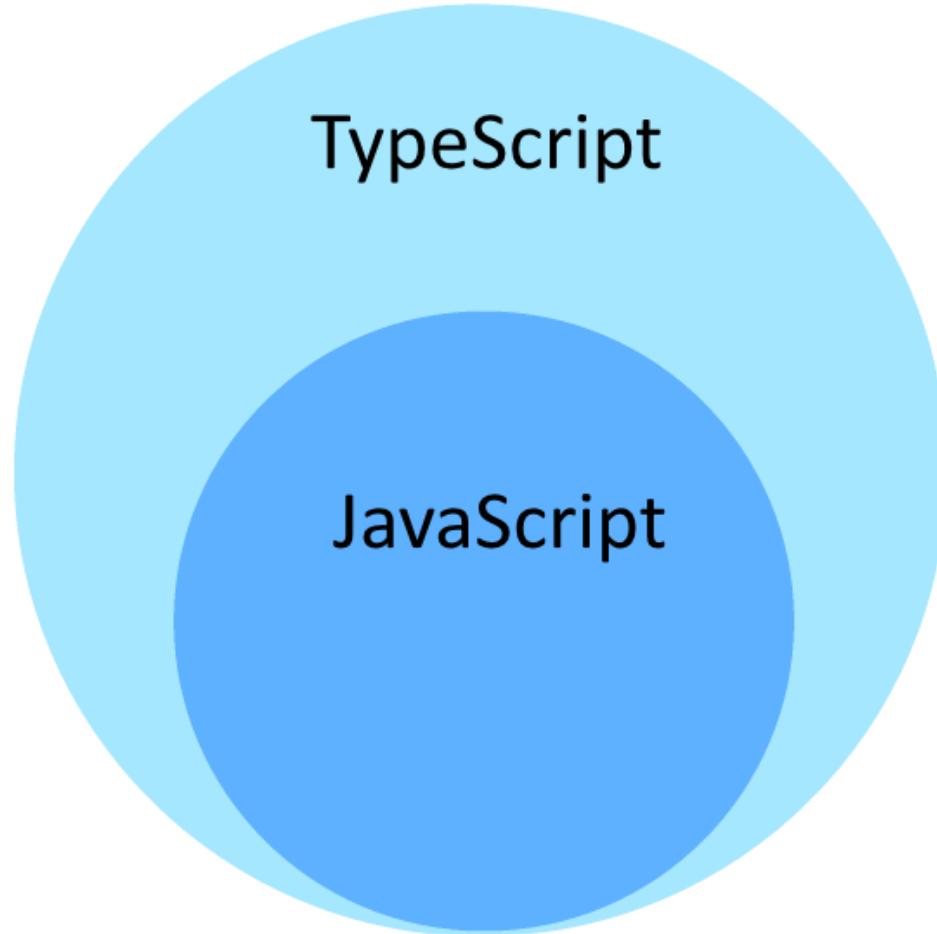


Paradigm	Multi-paradigm: functional, generic, imperative, object-oriented
Designed by	Microsoft
Developer	Microsoft
First appeared	1 October 2012; 8 years ago <sup>[1]</sup>
Stable release	4.0.3 / 18 September 2020; 25 days ago <sup>[2]</sup>
Preview release	4.0 Beta / 7 July 2020; 3 months ago <sup>[2]</sup>
Typing discipline	Duck, gradual, structural <sup>[3]</sup>
License	Apache License 2.0
Filename extensions	.ts, .tsx
Website	<a href="http://www.typescriptlang.org">www.typescriptlang.org</a>
Influenced by	C#, Java, JavaScript
Influenced	AtScript, AssemblyScript

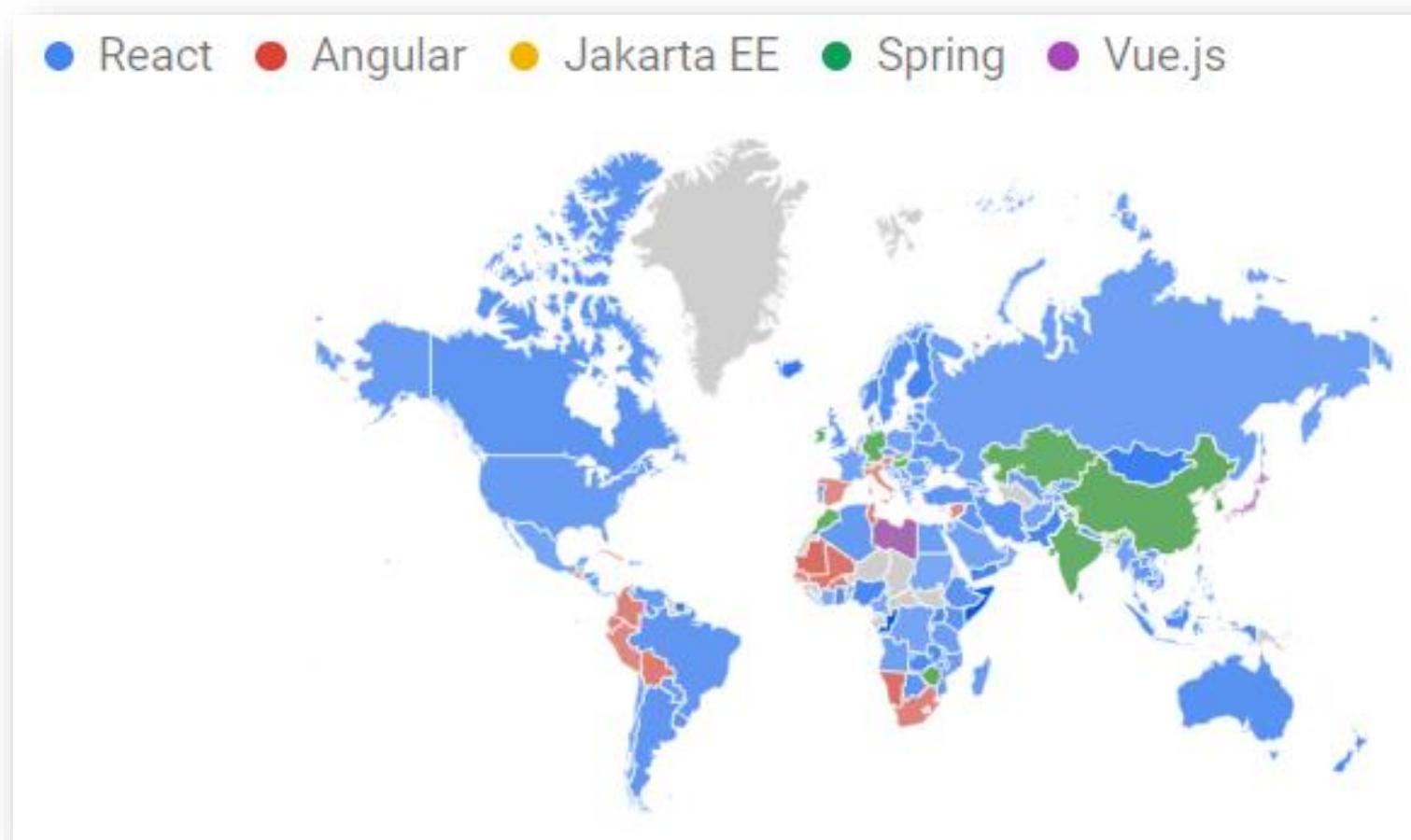
- **Langage de Programmation Orientée Objet, open source influencé par C# et JavaScript**
- Développé et présenté par **MicroSoft** en 2012
- Par rapport au **JavaScript**, il supporte l'**ES6** et il permet de :
  - **typer les variables**
  - définir des **classes** et des **interfaces**
  - utiliser les **annotations (les décorateurs)**
  - exporter et importer des **modules**

# TypeScript

## Sur-ensemble du JavaScript

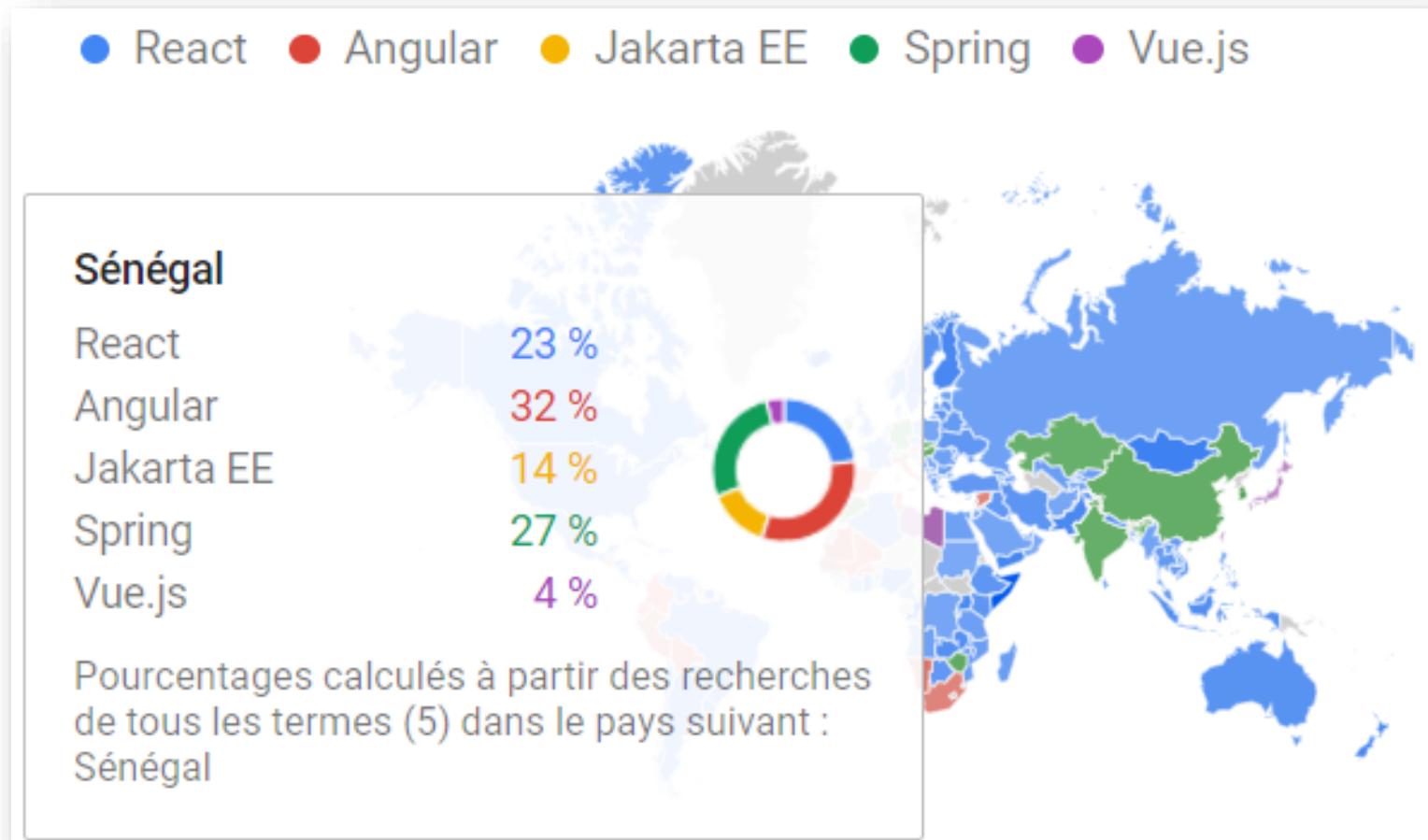


# Google Trends React vs. Angular vs. Jakarta EE vs. Vue.js vs. Spring de janvier à Octobre 2020



<https://trends.google.com/trends/explore?cat=5&date=2020-01-01%202020-12-31&q=%2Fm%2F012l1vxv,%2Fg%2F11c6w0ddw9,%2Fm%2F0bs6x,%2Fm%2F0dhx5b,%2Fg%2F11c0vmgx5d>

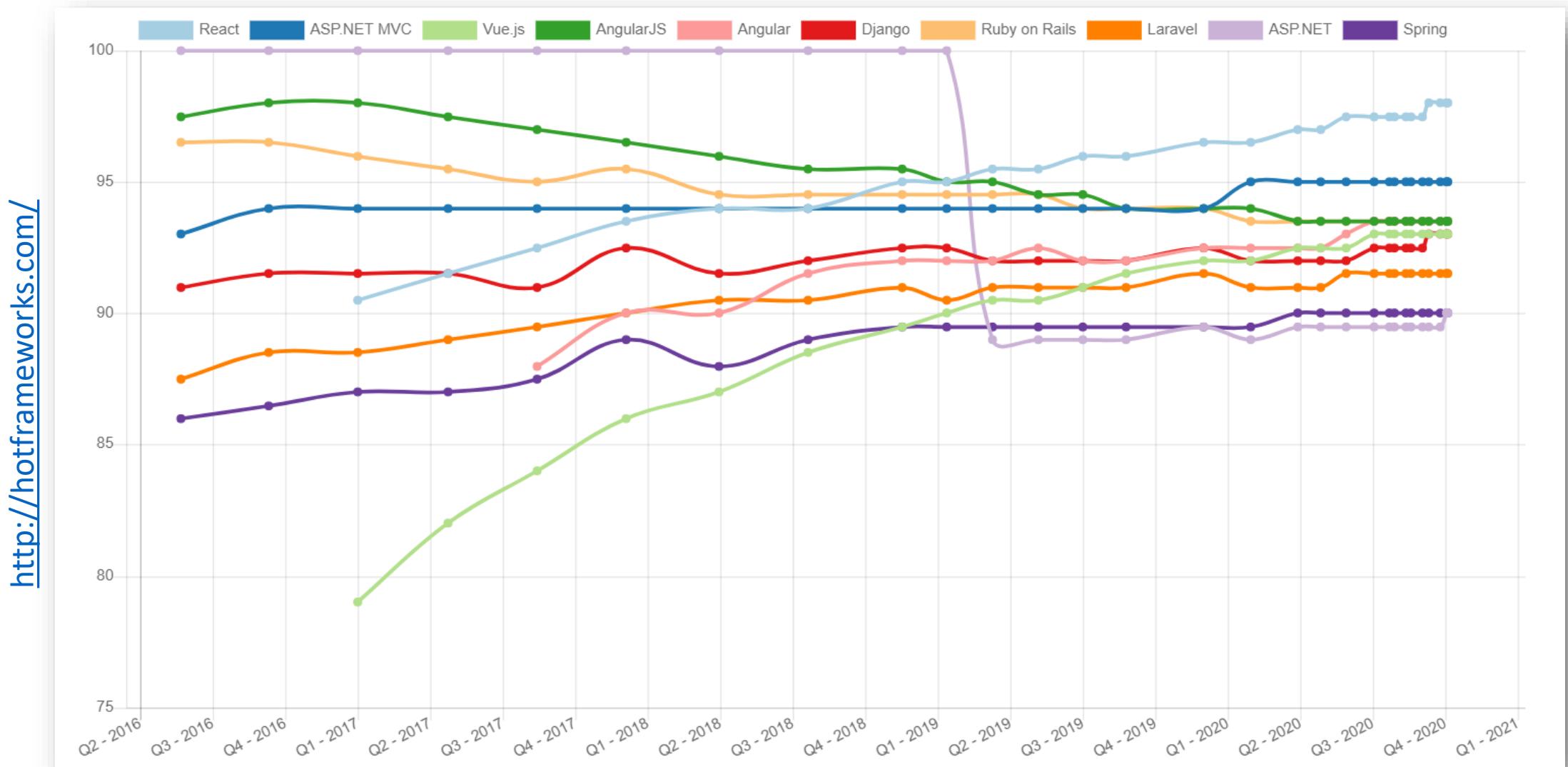
# Google Trends React vs. Angular vs. Jakarta EE vs. Vue.js vs. Spring au **Sénégal** Janvier à Octobre 2020



<https://trends.google.com/trends/explore?cat=5&date=2020-01-01%202020-12-31&q=%2Fm%2F012l1vxv,%2Fg%2F11c6w0ddw9,%2Fm%2F0bs6x,%2Fm%2F0dhx5b,%2Fg%2F11c0vmgx5d>

# Hotframeworks.com

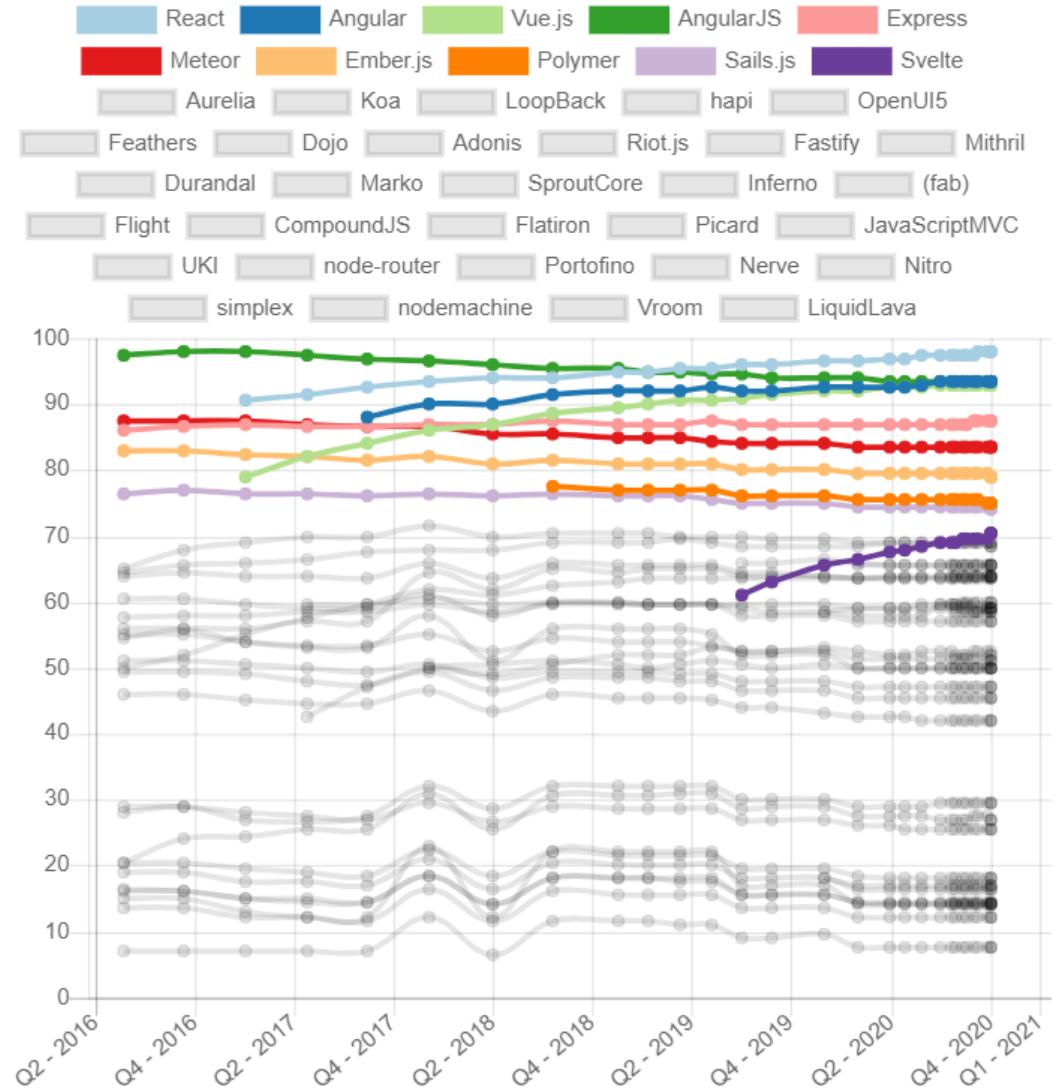
Popularité du framework web pour trouver les frameworks intéressants à regarder.



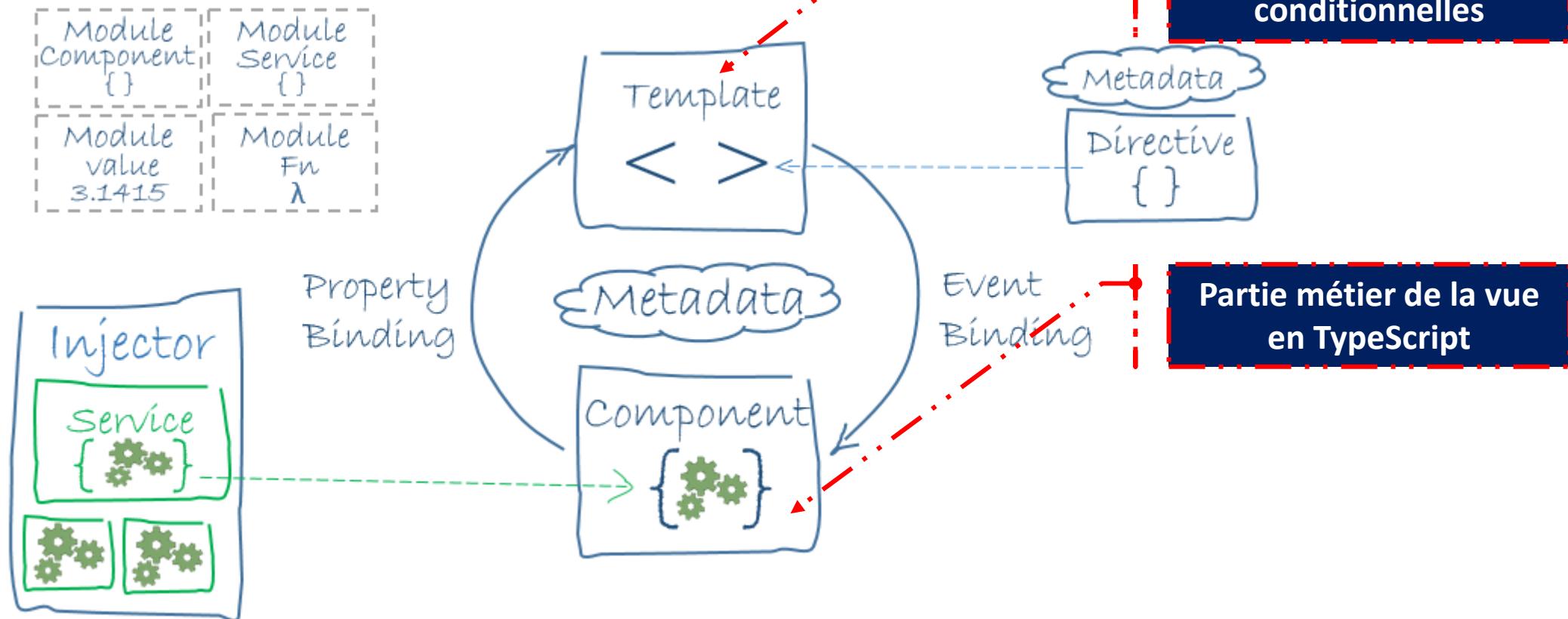
# Frameworks JavaScript

<http://hotframeworks.com/languages/javascript>

Framework	Score
React	98
Angular	93
Vue.js	93
AngularJS	93
Express	87
Meteor	83
Ember.js	79
Polymer	75
Sails.js	74
Svelte	70
Aurelia	69
Koa	68
LoopBack	65
hapi	65
OpenUI5	64



# Architecture du Framework Angular



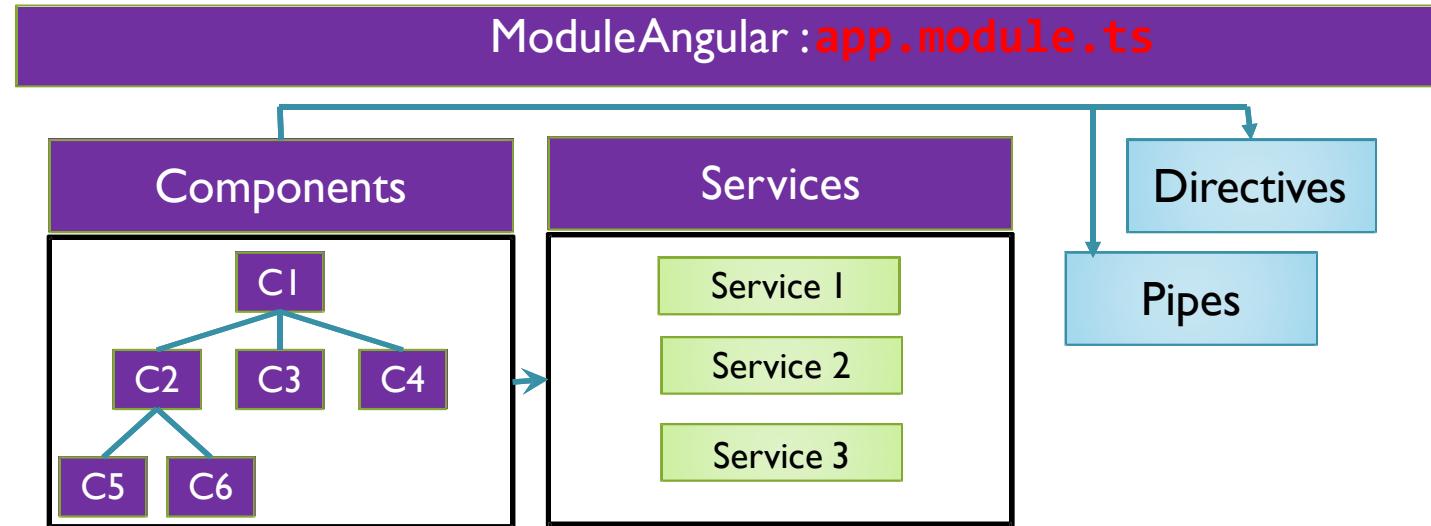
Une application Angular se compose de un à plusieurs modules dont un est principal. Chaque module peut inclure :  
Des composant web : La partie visible de l'application Web (IHM)  
Des services pour la logique applicative.

# Architecture du Framework Angular

- Une application **Angular** se compose d'un à plusieurs **modules** dont un est principal **app.module.ts**

## Chaque module peut inclure :

- Des **composant web** : La partie visible de la 'application Web (IHM)
- Des **services** pour la logique applicative. Les composants peuvent utiliser les services via le principe de l'**injection des dépendances**.
- Les **directives** : un composant peut utiliser des directives
- Les **pipes** : utilisés pour formater l'affichage des données dans els composants.



# Angular : une Architecture d'une Application basée sur les composants



- Exemple simple avec 05 composants fils du composant racine **app-root**
- **AppComponent** est le composant principal : tous les autres composants seront emboîtés ou "nested" dans celui-ci.
- **Un Module est ici un groupe de composants** par exemple **app.module.ts**

# Composants web

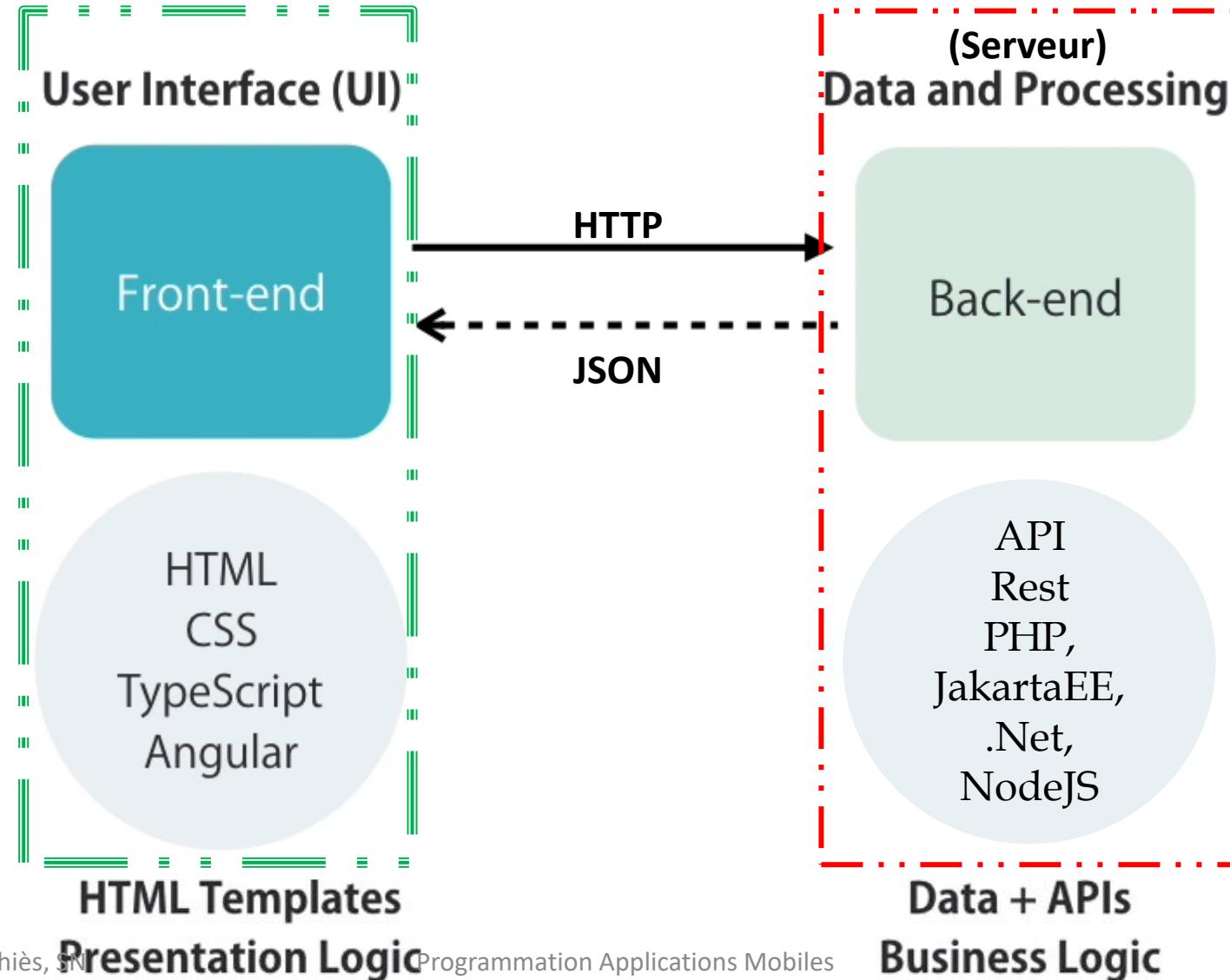
- Constitués de plusieurs technologies web (HTML, CSS, TypeScript...)
- Permettent de définir des balises personnalisées et réutilisables



# Composants web

- Chaque composant se compose principalement des éléments suivants:
  - **HTML Template** : représentant sa **vue**(personne.component.html)
  - Une **classe typescript** représentant sa **logique métier** (personne.component.ts)
  - Une **feuille de style CSS** pour le **design**(personne.component.css)
  - Un **fichier spec** sont des **tests unitaires**(personne.component.spec.ts)
- Les composants sont faciles à mettre à jour et à échanger entre les différentes parties des applications.

# Angular : une Architecture Applicative basée sur les composants



Stack Overflow

<https://stackoverflow.com/tags/angular>





# Angular University

Learn And Keep Up With The Angular Ecosystem

[VIEW ALL LESSONS](#)

## Courses updated to the latest Angular (v10)

Learn Angular in a fast and effective way: via a series of short lessons (about 5 min) that cover a single well-contained topic, lessons updated ASAP with releases.

## Courses for All Levels

Premium Quality Weekly screencasts for all levels: Beginner, Intermediate or Expert, covering everything from in-depth lessons to What's New, or taking a look under the hood.

## Videos & E-Books

Our video lessons are complemented with E-Books in PDF format. Each E-Book drills down into one specific Angular topic in detail: for example the Router, Forms and much more.



# Installations et configurations

1. Installer **NodeJS** et NodeJS <https://nodejs.org/>
2. Installer **GIT** <https://git-scm.com/downloads>
3. Installer **Angular CLI**(Command-Line Interface)

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

3. Installer **Visual Studio Code** <https://code.visualstudio.com/>



One framework.  
Mobile & desktop.

[GET STARTED](#)

#### DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

A Angular - Introduction to the An x

+ - □ X

angular.io/docs

Search Twitter  GitHub 

ANGULAR FEATURES DOCS RESOURCES EVENTS BLOG

INTRODUCTION

GETTING STARTED > Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

MAIN CONCEPTS > These Angular docs help you learn and use the Angular framework and development platform, from your first application to optimizing complex single-page apps for enterprises. Tutorials and guides include downloadable examples to accelerate your projects.

BUILT-IN FEATURES >

BEST PRACTICES >

ANGULAR TOOLS >

TUTORIALS >

RELEASE INFORMATION >

REFERENCE >

stable (v10.1.5)

**Introduction to the Angular Docs** Edit

**Get Started**  
Set up your local environment for development with the Angular CLI.  
[Local setup](#)

**Learn and Explore**  
Learn about the fundamental design concepts and architecture of Angular apps.  
[Introduction to Angular concepts](#)

**Take a Look**  
Examine and work with a small ready-made Angular app, without any setup.  
[Try it now](#)

**Hello World**  
Work through a full tutorial to create your first app.  
[Tour of Heroes tutorial](#)

**Assumptions**

INTRODUCTION

GETTING STARTED

- Try it
- Setup

MAIN CONCEPTS

BUILT-IN FEATURES

BEST PRACTICES

ANGULAR TOOLS

TUTORIALS

RELEASE INFORMATION

REFERENCE

stable (v10.1.5)

## Setting up the local environment and workspace

This guide explains how to set up your environment for Angular development using the [Angular CLI tool](#). It includes information about prerequisites, installing the CLI, creating an initial workspace and starter app, and running that app locally to verify your setup.

### TRY ANGULAR WITHOUT LOCAL SETUP

If you are new to Angular, you might want to start with [Try it now!](#), which introduces the essentials of Angular in the context of a ready-made basic online store app that you can examine and modify. This standalone tutorial takes advantage of the interactive [StackBlitz](#) environment for online development. You don't need to set up your local environment until you're ready.

## Prerequisites

To use the Angular framework, you should be familiar with the following:

- [JavaScript](#)
- [HTML](#)
- [CSS](#)

Knowledge of [TypeScript](#) is helpful, but not required.

To install Angular on your local system, you need the following:

- [Node.js](#)

Angular requires a [current, active LTS, or maintenance LTS](#) version of Node.js.

For information about specific version requirements, see the engines key in the [package.json](#) file.

Setting up the local environment and workspace

Prerequisites

Install the Angular CLI

Create a workspace and initial application

Run the application

Next steps

<https://angular.io/guide/setup-local>

[HOME](#) | [ABOUT](#) | [DOWNLOADS](#) | [DOCS](#) | [GET INVOLVED](#) | [SECURITY](#) | [CERTIFICATION](#) | [NEWS](#)

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

# Installer NodeJS

<https://nodejs.org/>

#BlackLivesMatter

Download for Windows (x64)

12.19.0 LTS

Recommended For Most Users

14.13.1 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

- C'est un runtime Javascript écrit en C++.
- Il permet d'interpréter et exécuter du code Javascript.
- Il fournit par défaut un ensemble de bibliothèques pour interagir avec le système d'exploitation et concevoir des serveurs Web.



© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

[Report Node.js issue](#) | [Report website issue](#) | [Get Help](#)

Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the [Trademark List](#) and [Trademark Guidelines](#) of the OpenJS Foundation.

[Node.js Project Licensing Information](#).



Thank you larissayvette for being a



# NPM (Node Package Manager)

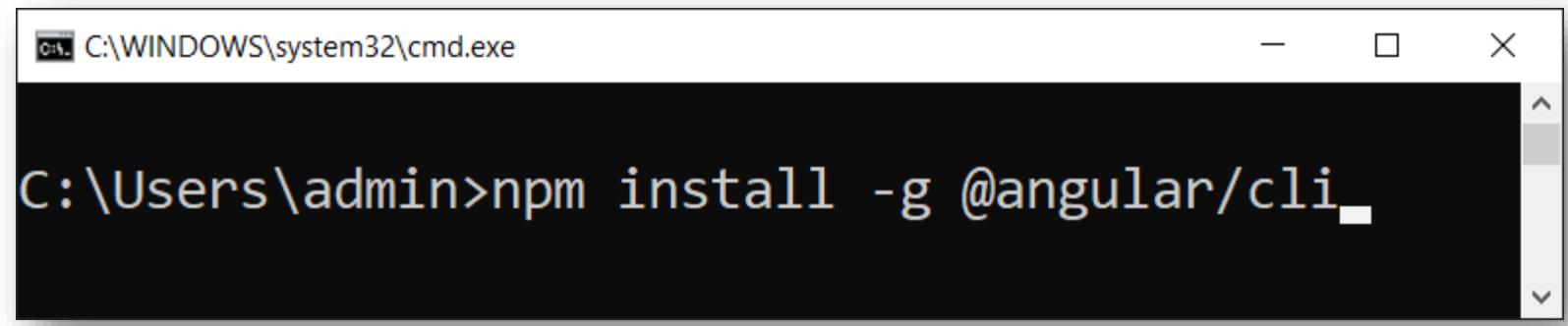
- Lorsque vous installez node, vous installez également npm.
- C'est un outil en ligne de commande qui vous permet de facilement installer des packages écrits en Javascript et compatibles avec NodeJS.
- Pour rechercher des packages rendez-vous sur <https://npmjs.com>



# Installer Angular CLI(Command-Line Interface)

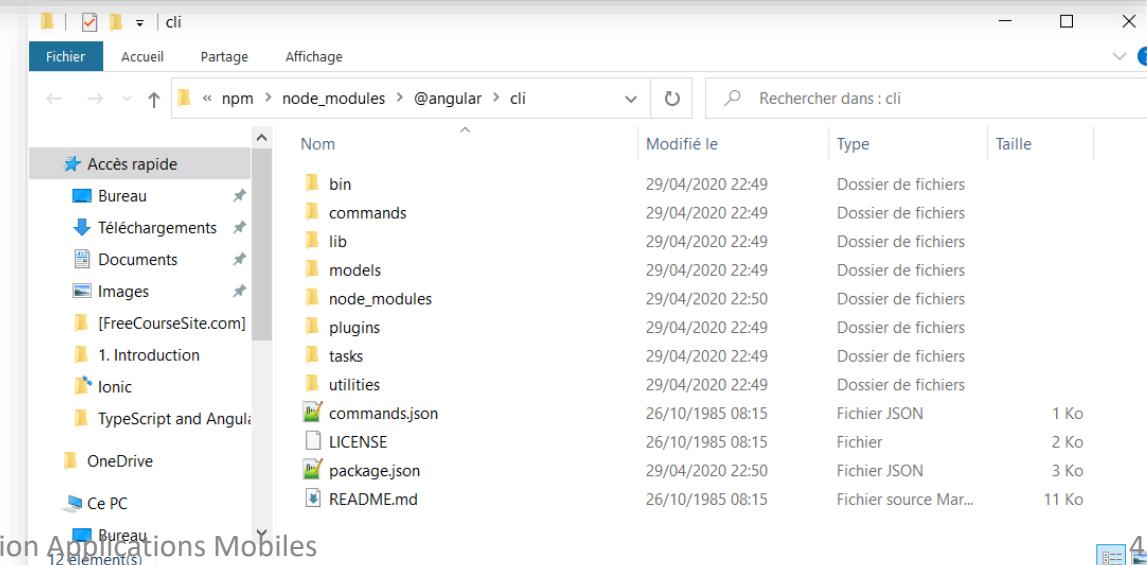
- Sur windows

Permet de générer,  
compiler, tester et  
déployer des projets  
angular



```
C:\WINDOWS\system32\cmd.exe
C:\Users\admin>npm install -g @angular/cli
```

- Sous MacOs ajouter « sudo »



```
C:\WINDOWS\system32\cmd.exe
C:\Users\admin>ng --version

Angular CLI: 9.1.3
Node: 12.16.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package          Version
-----
@angular-devkit/architect    0.901.3
@angular-devkit/core         9.1.3
@angular-devkit/schematics   9.1.3
@schematics/angular          9.1.3
@schematics/update           0.901.3
rxjs                  6.5.4
```

Pour désinstaller

```
npm uninstall -g  
@angular/cli npm cache  
clean --force
```

# Editeur de Développement Intégré



**IntelliJ IDEA**



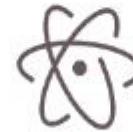
Sublime Text



Emacs



Visual Studio Code



Atom



WebStorm



Visual Studio



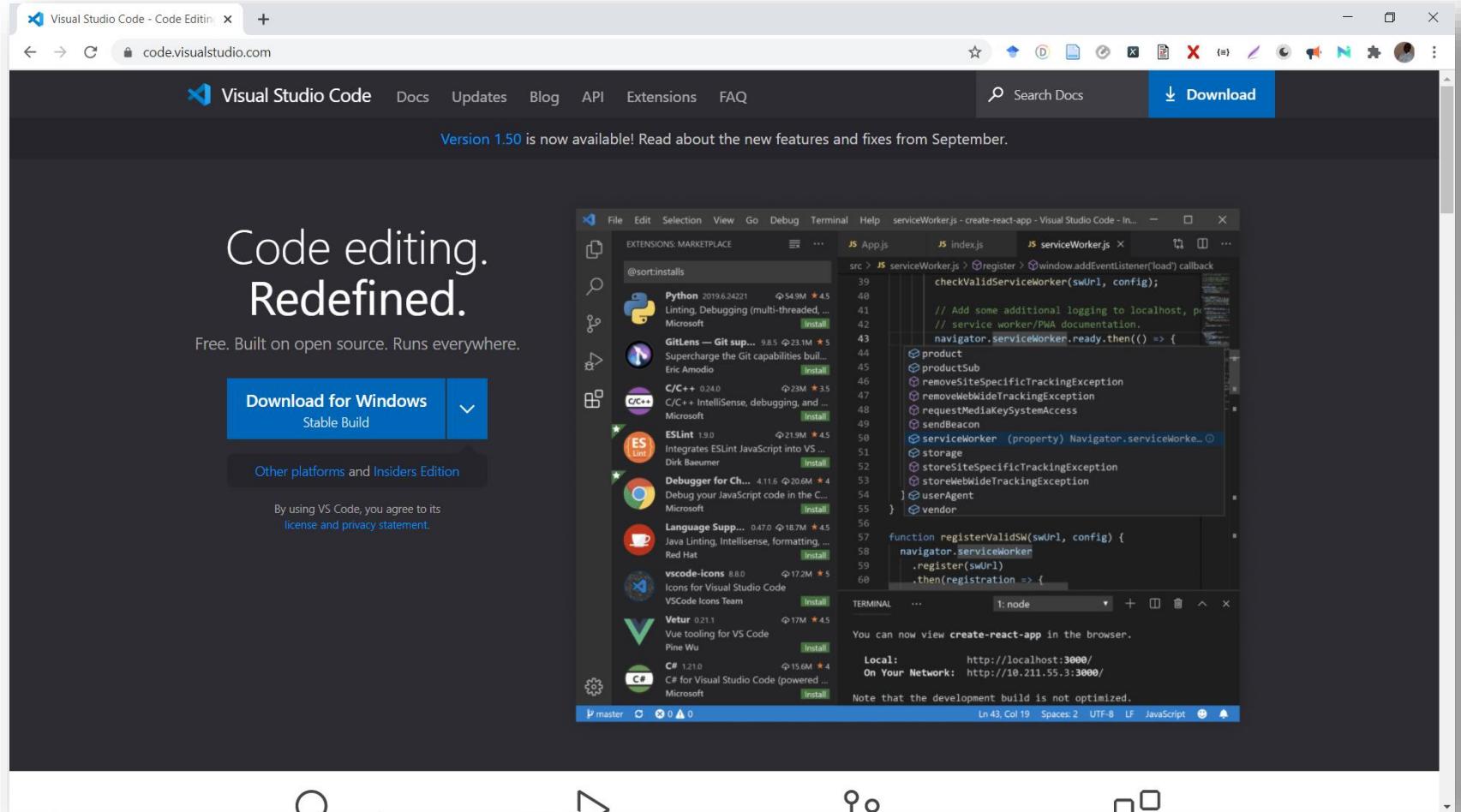
Eclipse



Vim

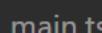
# Installer Visual Studio Code

## <https://code.visualstudio.com/>





EXTENSIONS



Search Extensions in Marketplace

INSTALLED

**Angular 10 Snippets - TypeScript...** 10.0.3

242 Angular Snippets (TypeScript, Ht...)

Mikael Morlund

**Angular Language Service** 0.1001.0

Editor services for Angular templates

Angular

**Awesome Flutter Snippets** 2.0.4

Awesome Flutter Snippets is a collecti...

Neevash Ramdial

**Dart** 3.15.0

Dart language support and debugger ...

Dart Code

**Dart Data Class Generator** 0.4.2

Create dart data classes easily, fast an...

BendixMa



RECOMMENDED

**Debugger for Firefox** 2.9.1

Debug your web application or brows...

Firefox DevTools

**GitLens — Git supercharged** 10.2.2

Supercharge the Git capabilities built i...

Eric Amodio



main.ts

Extension: Angular 10 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS &amp; Flex Layout



# Angular 10 Snippets - TypeScript, Html, Angular |

Mikael Morlund | 1520166 | ★★★★★ | Repository | License | v10.

242 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ngRx, RxJS, PWA &amp; T...)

[Disable](#) [Uninstall](#)

This extension is enabled globally.

[Details](#) [Feature Contributions](#) [Changelog](#)

Visual Studio Marketplace v10.0.3 installs 1.52M

## VSCode Angular TypeScript & Html Snippets

Visual Studio Code TypeScript and Html snippets and code examples for Angular 2,4,5,6,7,8,9 &amp; 10.

All code snippets are based on and follow the Angular style guide <https://angular.io/docs/ts/latest/style-guide/style-guide.html>

The source for the snippets come from:

- <https://angular.io/>
- <https://material.angular.io/>
- <https://github.com/angular/flex-layout>
- <https://github.com/ngrx/platform>



# Une première application Angular: Créer un nouveau projet avec la commande ng

```
C:\Users\admin>ng
Available Commands:
  add Adds support for an external library to your project.
  analytics Configures the gathering of Angular CLI usage metrics. See https://v8.angular.io/cli/usage-analytics-gathering.
  build (b) Compiles an Angular app into an output directory named dist/ at the given output path. Must be executed from within a workspace directory.
  deploy Invokes the deploy builder for a specified project or for the default project in the workspace.
  config Retrieves or sets Angular configuration values in the angular.json file for the workspace.
  doc (d) Opens the official Angular documentation (angular.io) in a browser, and searches for a given keyword.
  e2e (e) Builds and serves an Angular app, then runs end-to-end tests using Protractor.
  generate (g) Generates and/or modifies files based on a schematic.
  help Lists available commands and their short descriptions.
  lint (l) Runs linting tools on Angular app code in a given project folder.
  new (n) Creates a new workspace and an initial Angular app.
  run Runs an Architect target with an optional custom builder configuration defined in your project.
  serve (s) Builds and serves your app, rebuilding on file changes.
  test (t) Runs unit tests in a project.
  update Updates your application and its dependencies. See https://update.angular.io/
  version (v) Outputs Angular CLI version.
  xi18n (i18n-extract) Extracts i18n messages from source code.
```

Plusieurs options

For more detailed help run "ng [command name] --help"

```
> npm install -g @angular/cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

## Angular CLI

A command line interface for Angular

[GET STARTED](#)

## ng new

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

## ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.

# Une première application Angular: Créer un nouveau projet avec la commande ng

```
C:\WINDOWS\system32\cmd.exe
C:\Users\ousma>npm install -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\ousma\AppData\Roaming\npm\ng -> C:\Users\ousma\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@10.1.6 postinstall C:\Users\ousma\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

+ @angular/cli@10.1.6
added 276 packages from 206 contributors in 97.04s

C:\Users\ousma>cd Desktop

C:\Users\ousma\Desktop>cd workspace-angular

C:\Users\ousma\Desktop\workspace-angular>ng new premiereAppAngular
```

# Une première application Angular: Créer un nouveau projet avec la commande **ng new nomDeVotreProjet**

```
cmd npm
+ @angular/cli@10.1.6
added 276 packages from 206 contributors in 97.04s

C:\Users\ousma>cd Desktop

C:\Users\ousma\Desktop>cd workspace-angular

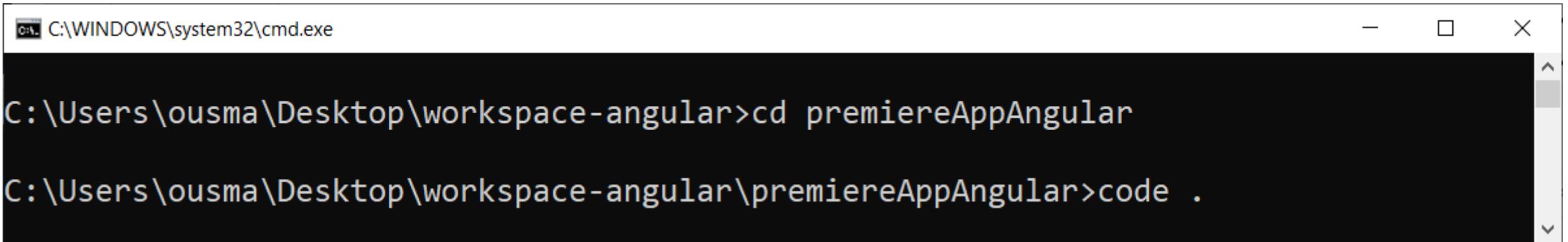
C:\Users\ousma\Desktop\workspace-angular>ng new premiereAppAngular
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
Stylus [ http://stylus-lang.com ]
```

# Unexpected end of JSON input while parsing near '...iUFJEQMjb7HL1sZAGG9x8'

- <https://stackoverflow.com/questions/39566257/how-to-uninstall-upgrade-angular-cli>
- Pensez à désinstaller tout et surtout réinstaller nodejs avec une autre version que la dernière.

# Ouvrir le projet dans Visual Studio Code

- Commande code . Dans le dossier du projet



A screenshot of a Windows Command Prompt window. The title bar says "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
C:\Users\ousma\Desktop\workspace-angular>cd premiereAppAngular
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>code .
```



EXPLORER

...



...

## &gt; OPEN EDITORS

## PREMIEREAPPANGULAR

- > e2e
- > node\_modules
- > src
  - .browserslistrc
  - .editorconfig
  - .gitignore
  - angular.json
  - karma.conf.js
- package-lock.json
- package.json
- README.md
- {...} tsconfig.app.json
- {...} tsconfig.json
- {...} tsconfig.spec.json
- {...} tslint.json

...

La commande `ng new` génère la structure d'un projet Angular avec les différents fichiers requis par une application basique et installe aussi toutes les dépendances requises par le projet.

PHP, Azure,...

of Vim, Sub...

first\_app C:\Users\ousma\Desktop\flutter\_exemples

More... (Ctrl+R)

Color theme

Make the editor and your code look the way you love

## Learn

Find and run all commands

Rapidly access and search commands from the Comm...

## Interface overview

Get a visual overlay highlighting the major component...

Interactive playground



## Help

[Printable keyboard cheatsheet](#)[Introductory videos](#)[Tips and Tricks](#)[Product documentation](#)[GitHub repository](#)[Stack Overflow](#)[Join our Newsletter](#)

&gt; OUTLINE

&gt; TIMELINE

# Une première application Angular: déployer votre application dans le serveur

- Compiler l'application et générer les fichiers nécessaires **ng serve -o**

```
--open ou -o pour ouverture automatique du navigateur
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>ng serve -o
```

```
Windows PowerShell

chunk {main} main.js, main.js.map (main) 59.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 12.5 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 2.62 MB [initial] [rendered]
Date: 2020-10-13T22:36:53.885Z - Hash: 6bfe7e17caeeebafa21f - Time: 16601ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Dépendances javascript  
incluses par AngularCLI



Welcome



premiereAppAngular app is running!

## Resources

Here are some links to help you get started:

 Learn Angular

<> CII Documentation

Angular Blog >

## Next Steps

What do you want to do next with your app?

+ New Component

+ Angular Materi

+ Add PWA Support

+ Add Dependency

## + Run and Watch Test

## + Build for Production

10

ng generate component xyz

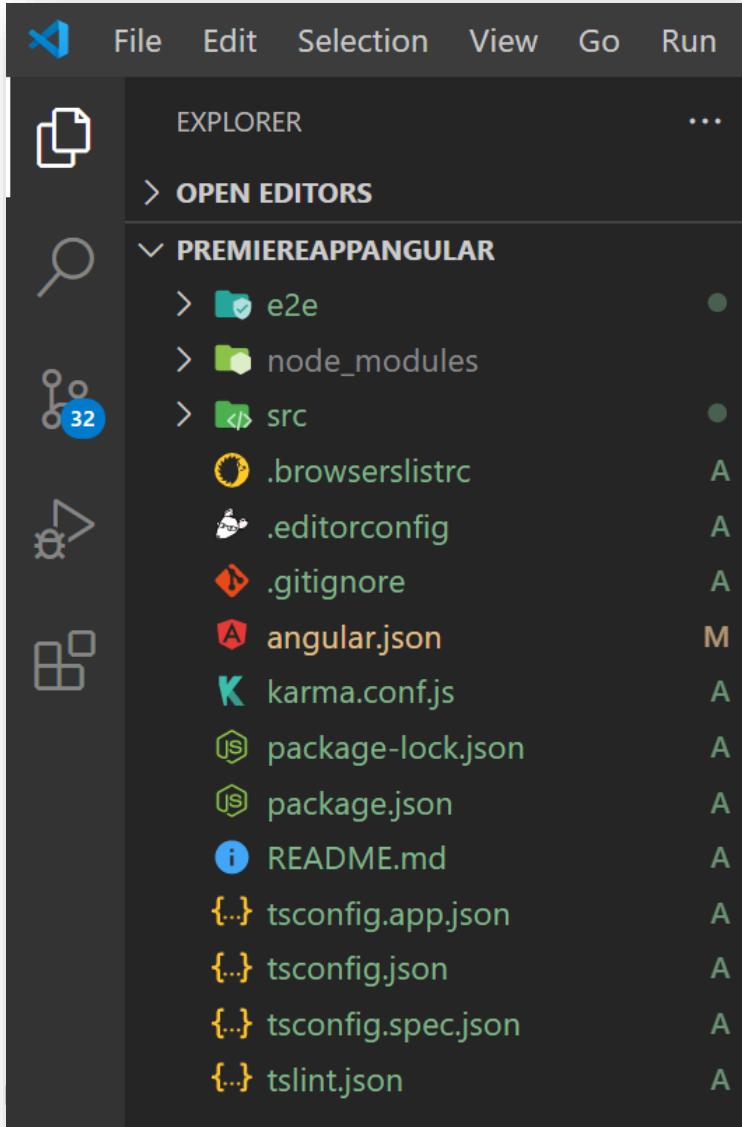


Love Angular? Give our repo a star. [Star](#)

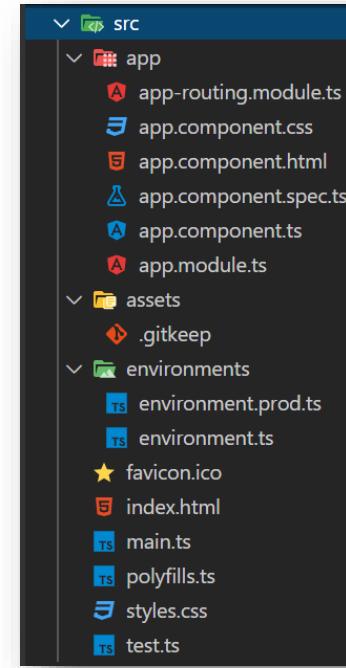


Cette commande compile le code source du projet pour transpiler le code TypeScript en JavaScript et en même temps démarre un serveur Web local basé sur NodeJS pour déployer l'application localement.

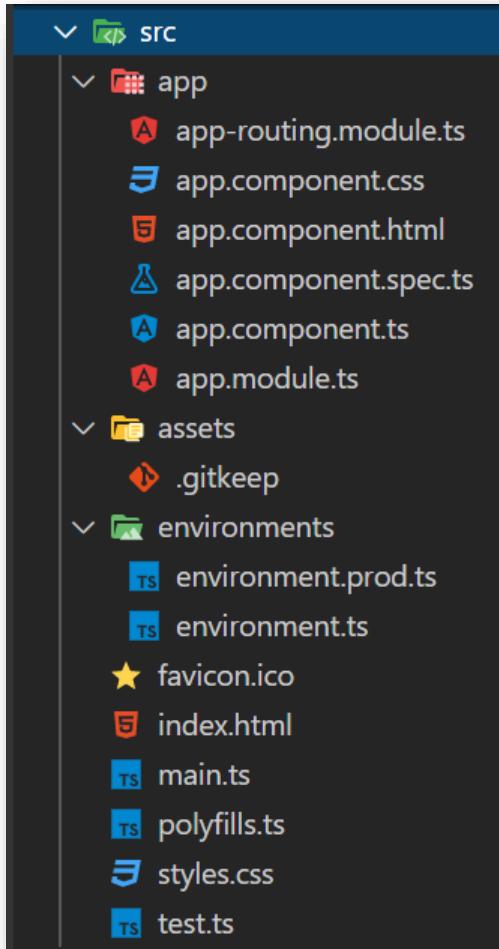
# Structure d'un projet Angular



- **e2e** contenant les fichiers permettant de tester l'application
- **node\_modules** contenant les fichiers nécessaires de la librairie nodeJS
- **src** dossier pour le code source de l'application
- **package.json** : contenant l'ensemble des dépendances (genre *pom.xml* sous java) de l'application
- **angular.json** : contenant les données concernant la configuration du projet (les commandes raccourcies, l'emplacement des fichiers de démarrage...)
- **tslint.json** : contenant les données sur les règles à respecter par le développeur (nombre de caractères max par ligne, l'emplacement des espaces...)



# Structure d'un projet Angular



- **assets** : l'unique dossier accessible aux visiteurs et contenant les images, les sons...
- **index.html** : l'unique fichier HTML d'une application **Angular**
- **styles.css** : la feuille de style commune de tous les composants web de l'application
- **favicon.ico** : le logo d'**Angular**
- **app** : contient initialement les 06 fichiers du module principal
  - **app.module.ts** : la classe correspondante au module principal
  - **app.component.ts** : la classe associée au composant web
  - **app.component.html** : le fichier contenant le code HTML associé au composant web
  - **app.component.css** : le fichier contenant le code CSS associé au composant web
  - **app.component.spec.ts** : le fichier de test du composant web

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure:
  - src folder contains app, assets, environments, favicon.ico, and index.html (which is selected).
  - assets folder contains .gitkeep.
  - environments folder contains environment.ts.
  - main.ts, polyfills.ts, styles.css, and test.ts are also listed.
  - Configuration files like .editorconfig, .gitignore, angular.json, karma.conf.js, package-lock.json, and package.json are present.
- Editor (Center):** The file index.html is open, displaying the following code:

```
src > index.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <title>PremiereAppAngularUvs</title>
6  <base href="/">
7  <meta name="viewport" content="width=device-width, initial-scale=1">
8  <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
```
- Bottom Status Bar:** Shows the current branch (master\*), commit count (0), pull request count (0), and file status (0).
- Bottom Right:** Includes navigation icons for back, forward, search, and refresh, along with status information: Ln 1, Col 1, Spaces: 2, UTF-8, LF, HTML, and a magnifying glass icon.

**Text Overlay (Bottom Right):**

index.html : l'unique fichier HTML d'une application  
Angular  
Le corps de la page contient un seul composant dont le  
seleteur est **app-root**

# Structure d'un projet Angular

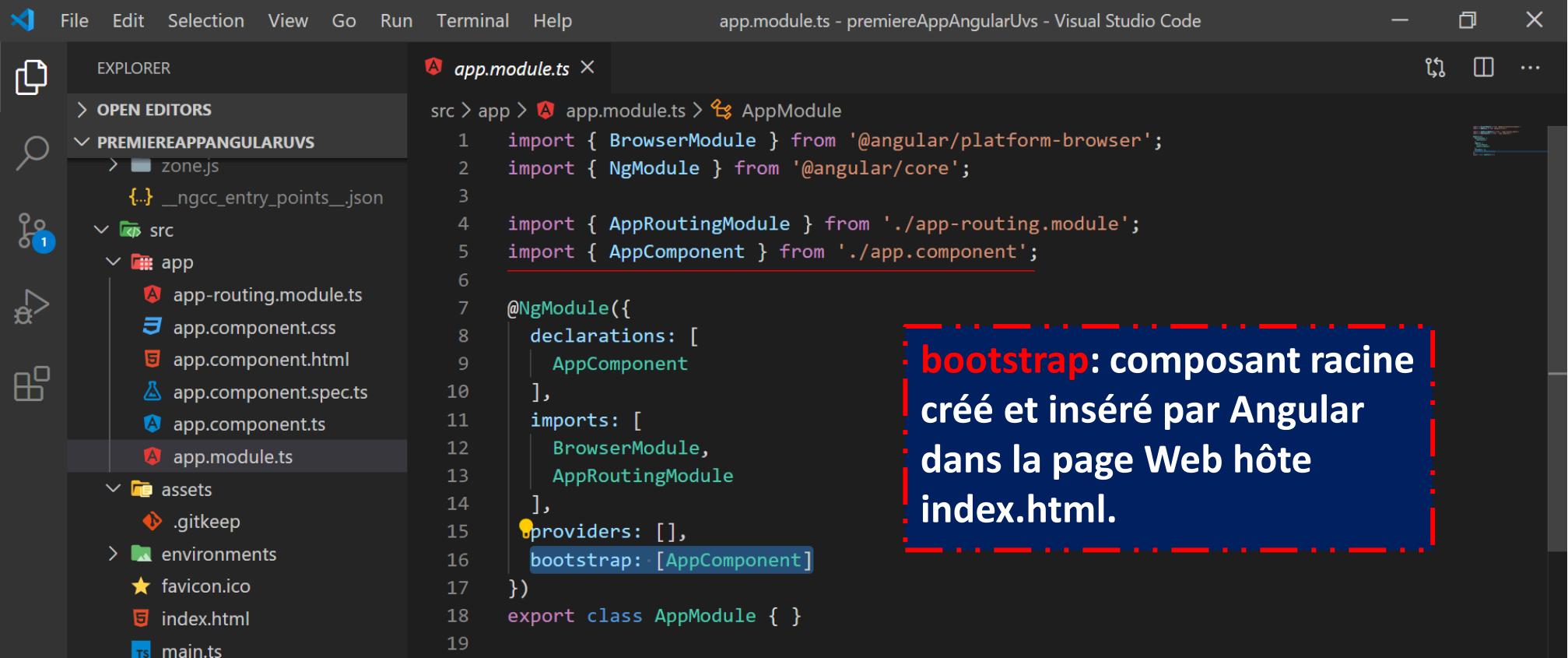
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - premiereAppAngularUvs - Visual Studio Code.
- Explorer Panel:** Shows the project structure:
  - > OPEN EDITORS
  - < PREMIEREAPPANGULARUVS
    - < src
      - > app
      - < assets
        - .gitkeep
      - > environments
      - favicon.ico
      - index.html
    - main.ts
    - polyfills.ts
    - styles.css
    - test.ts
  - .editorconfig
  - .gitignore
  - angular.json
  - browserslist
  - karma.conf.js
  - package-lock.json
  - package.json
- Editor Panel:** Displays the main.ts file content:

```
src > main.ts > ...
1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.error(err));
13
```
- Bottom Status Bar:** Prof. Oscarman SAKI Univ. Thiès, SN, Programmation Applications Mobiles, Ln 1, Col 1, Spaces: 2, UTF-8, LF, TypeScript 3.8.3, 63.

**Point d'entrée de l'application**  
 AppModule sera celui qui va démarrer en premier parmi les modules. Il s'agit de la classe AppModule dans le fichier app.module.ts du dossier App

# Structure d'un projet Angular



```
File Edit Selection View Go Run Terminal Help
app.module.ts - premiereAppAngularUvs - Visual Studio Code
EXPLORER app.module.ts
src > app > app.module.ts > AppModule
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6
7 @NgModule({
8   declarations: [
9     AppComponent
10   ],
11   imports: [
12     BrowserModule,
13     AppRoutingModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
```

bootstrap: composant racine  
créé et inséré par Angular  
dans la page Web hôte  
index.html.

- **@NgModule** : pour déclarer cette classe comme module
- **declarations** : dans cette section, on déclare les **composants** de ce module
- **imports** : dans cette section, on déclare les modules nécessaires pour le module
- **providers** : dans cette section, on déclare les services qui seront utilisés dans le module
- **bootstrap** permet à Angular de connaître le composant qui sera le composant de départ de l'application

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PREMIEREAPPANGULARUVS". The "src" folder contains "app" which has "app-routing.module.ts", "app.component.css", "app.component.html", "app.component.spec.ts", and "app.component.ts".
- Code Editor:** The file "app.component.ts" is open. The code defines an "AppComponent" class with the following properties:

```
src > app > A app.component.ts > AppComponent
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'premiereAppAngularUvs';
10 }
```
- Annotations:** A red dashed box highlights the annotations for the **@Component** decorator and its properties (**selector**, **templateUrl**, **styleUrls**).
- Text:** A callout box with a red border contains the following text:

**@Component** : pour déclarer cette classe comme composant web

  - **selector** : pour définir le nom de balise correspondant à ce composant web pour l'inclure dans la vue par exemple
  - **templateUrl** : pour indiquer le fichier contenant le code HTML correspondant au composant web
  - **styleUrls** : pour indiquer le(s) fichier(s) contenant le code CSS correspondant au composant web
- Bottom Status Bar:** Shows the current file is "master\*", 0 changes, and the code editor is in "TypeScript 3.8.3". Other status includes "Ln 5, Col 1 (38 selected)", "Spaces: 2", "UTF-8", "LF", and "Programmation Applications Mobiles".

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the Angular project structure. The Explorer sidebar on the left lists files and folders like .editorconfig, .gitignore, angular.json, browserslist, karma.conf.js, package-lock.json, package.json, README.md, and various ts and html files. The app.component.html file is open in the main editor area, showing the following code:

```
<div class="content" role="main">
  <!-- Highlight Card -->
  <div class="card highlight-card card-small">
    
    <span>{{ title }} app is running!</span>
    
  </div>
  <!-- Resources -->
  <h2>Resources</h2>
  <p>Here are some links to help you get started:</p>
  <div class="card-container">
    <a class="card" target=" blank" rel="noopener" href="https://angular.io/tutorial">
```

A red box highlights the interpolation expression `{{ title }}` in the `span` element. To the right of the code editor, a large blue box contains the following bullet points:

- `{{ Interpolation }}`
- `{{ title }}`: **title** sera remplacé par sa valeur dans `app.component.ts`

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** styles.css - premiereAppAngularUvs - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - PREMIEREAPPANGULARUVS (selected)
  - zone.js
  - {...} \_\_ngcc\_entry\_points\_\_.json
  - src
    - app
      - app-routing.module.ts
      - app.component.css
      - app.component.html
      - app.component.spec.ts
      - app.component.ts
      - app.module.ts
    - assets
      - .gitkeep
    - environments
      - favicon.ico
      - index.html
      - main.ts
      - polyfills.ts
    - styles.css (selected)
    - test.ts
    - .editorconfig
    - .gitignore
    - angular.json
    - browserslist
    - karma.conf.js
  - OUTLINE
  - TIMELINE
  - NPM SCRIPTS
- Editor View:** The styles.css file content is displayed:

```
1 /* You can add global styles to this file, and also import other style files */
2
```
- Bottom Status Bar:** master\* 0 △0 Ln 1, Col 1 Spaces: 4 UTF-8 LF CSS ⚡ ⌂

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** package.json - premiereAppAngularUvs - Visual Studio Code.
- Explorer Bar (Left):**
  - OPEN EDITORS:** package.json (highlighted), package-lock.json, README.md, tsconfig.app.json, tsconfig.json, tsconfig.spec.json, tslint.json.
  - PREMIEREAPPANGULARUVS:** yeast, yn, zone.js, \_\_ngcc\_entry\_points\_\_.json, src (highlighted), .editorconfig, .gitignore, angular.json, browserslist, karma.conf.js, package-lock.json, package.json (highlighted).
  - OUTLINE, TIMELINE, NPM SCRIPTS:** These sections are also visible in the sidebar.
- Editor Area (Right):** The package.json file is open, showing its contents. The file includes definitions for the name, version, scripts, private status, dependencies, and devDependencies of the Angular application.
- Bottom Status Bar:** master\*, 0 0 △ 0, Prof. Ousmane SALL, Univ. Thiès, SN, Programmation Applications Mobiles, Ln 1, Col 1, Spaces: 2, UTF-8, LF, JSON, 68.

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** tsconfig.json - premiereAppAngularUvs - Visual Studio Code.
- Explorer View:** Shows the project structure under "PREMIEREAPPANGULARUVS".
  - yeast
  - yn
  - zone.js
  - \_ngcc\_entry\_points\_.json
  - src
    - .editorconfig
    - .gitignore
    - angular.json
    - browserslist
    - karma.conf.js
    - package-lock.json
    - package.json
    - README.md
  - tsconfig.app.json
  - tsconfig.json (highlighted)
  - tsconfig.spec.json
  - tslint.json
- Editor View:** Displays the content of the tsconfig.json file.

```
1  {
2    "compileOnSave": false,
3    "compilerOptions": {
4      "baseUrl": "./",
5      "outDir": "./dist/out-tsc",
6      "sourceMap": true,
7      "declaration": false,
8      "downlevelIteration": true,
9      "experimentalDecorators": true,
10     "module": "esnext",
11     "moduleResolution": "node",
12     "importHelpers": true,
13     "target": "es2015",
14     "lib": [
15       "es2018",
16       "dom"
17     ]
18   },
19   "angularCompilerOptions": {
20     "fullTemplateTypeCheck": true,
21     "strictInjectionParameters": true
22   }
23 }
```
- Bottom Status Bar:** master\* ⑧ 0 △ 0, Ln 1, Col 1, Spaces: 2, UTF-8, LF, JSON with Comments, search icon, notifications icon.
- Annotations:** A blue callout box highlights the "tsconfig.json" file in the Explorer view and its content in the Editor view, with the text: "The tsconfig.json file specifies the root files and the compiler options required to compile the project."
- Page Footer:** Prof. Ousmane SALL, Univ. Thiès, SN, Programmation Applications Mobiles, 69.

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PREMIEREAPPANGULARUVS". Key files visible include: yeast, yn, zone.js, \_\_ngcc\_entry\_points\_\_.json, src, .editorconfig, .gitignore, angular.json, browserslist, karma.conf.js, package-lock.json, package.json, README.md, tsconfig.app.json, tsconfig.json, tsconfig.spec.json, and tslint.json (which is currently selected).
- Code Editor:** Displays the content of the tslint.json file.
- Text in the code editor:**

```
1  {
2    "extends": "tslint:recommended",
3    "rules": {
4      "align": {
5        "options": [
6          "parameters",
7          "statements"
8        ]
9      },
10     "array-type": false,
11     "arrow-return-shorthand": true,
12     "curly": true,
13     "deprecation": {
14       "severity": "warning"
15     },
16     "component-class-suffix": true,
17     "contextual-lifecycle": true,
18     "directive-class-suffix": true,
19     "directive-selector": [
20       true,
21       "attribute",
22       "app",
23       "camelCase"
24     ],
25     "component-selector": [
26       true,
```
- Annotations:** A red dashed box highlights the text "Contenant les données sur les règles à respecter par le développeur (nombre de caractères max par ligne, l'emplacement des espaces...)" (Containing data on rules to be followed by the developer (maximum number of characters per line, placement of spaces...)).
- Text overlay:** A large blue box contains the text "Configuration du compilateur !!!" (Compiler configuration !!!).
- Bottom status bar:** Shows "master\*" and "0 △ 0" on the left, and "Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON" on the right.

# Structure d'un projet Angular

The screenshot shows the Visual Studio Code interface with two tabs open:

- app.component.ts**: The code defines an Angular component named `AppComponent` with a selector of `'app-root'`, a template URL of `'./app.component.html'`, and style URLs of `['./app.component.css']`. The title is set to `'Master CyberSécurité'`.
- app.component.html**: The template contains an `<div>` element with a class of `"content"` and a role of `"main"`. It includes an `<!-- Highlight Card -->` comment and a `<div>` element with a class of `"card highlight-card card-small"`. Inside this card, there is an `<svg>` element with an ID of `"rocket"` representing a rocket ship, and another `<svg>` element with an ID of `"rocket-smoke"` representing smoke. A `<span>` element displays the message `title` followed by the text `c'est la première application Angular !`. The code also includes sections for `<!-- Resources -->` and a `<h2>Resources</h2>` heading.

The status bar at the bottom indicates the file is in `master*`, has 0 changes, and is in `UTF-8` encoding.

# Structure d'un projet Angular

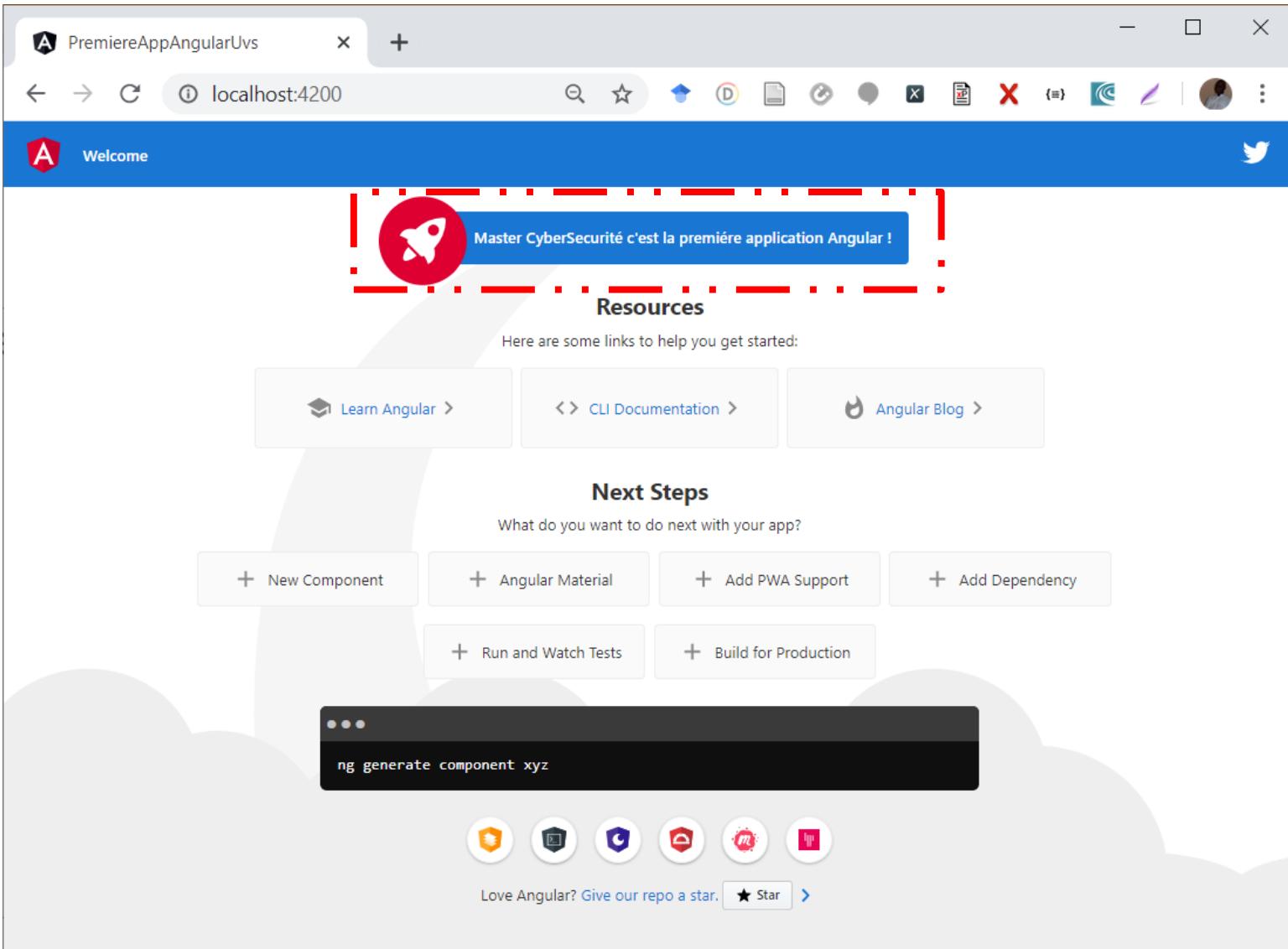
```
ngcc (worker)
Compiling @angular/router : es2015 as esm2015
Compiling @angular/animations/browser/testing : es2015 as esm2015
Compiling @angular/common/http/testing : es2015 as esm2015
Compiling @angular/platform-browser/animations : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
Compiling @angular/router/testing : es2015 as esm2015
Compiling @angular/platform-browser-dynamic/testing : es2015 as esm2015

chunk {main} main.js, main.js.map (main) 60.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 12.4 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3 MB [initial] [rendered]
Date: 2020-04-29T23:30:11.885Z - Hash: 1061a01e2f800e068a4d - Time: 30616ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.

Date: 2020-04-30T00:17:46.234Z - Hash: 374e0c3dcf988ce25735
4 unchanged chunks
chunk {main} main.js, main.js.map (main) 60.7 kB [initial] [rendered]
Time: 1839ms
: Compiled successfully.

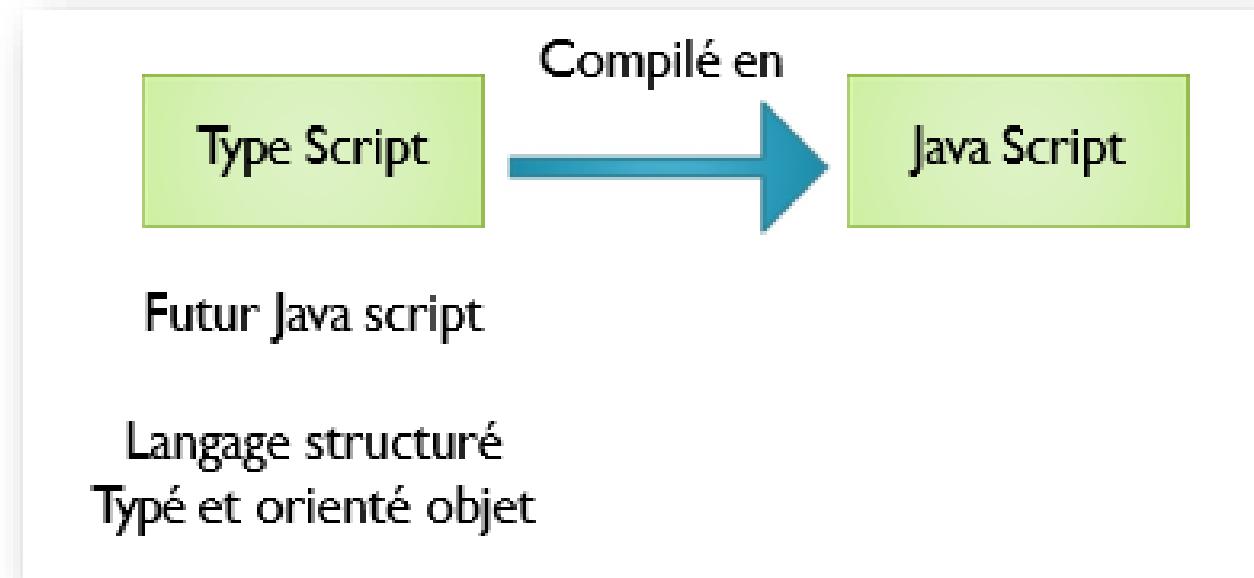
Date: 2020-04-30T00:17:47.718Z - Hash: 76eac7f943721b93d49f
4 unchanged chunks
chunk {main} main.js, main.js.map (main) 60.7 kB [initial] [rendered]
Time: 1098ms
: Compiled successfully.
```

# Structure d'un projet Angular



# Angular avec TypeScript

- Pour développer une application Angular il est recommandé d'utiliser TypeScript qui sera compilé et traduit en JavaScript.
- TypeScript est un langage de script structuré et orienté objet qui permet de simplifier le développement d'applications JavaScript



# Webographie

- <https://angular.io/>
- <http://typescriptlang.org>
- <https://www.youtube.com/watch?v=KJEJejxcnEQ&list=PLxr551TUsmAphhzmpn6lVyRGjAC1URrO>
- [https://www.youtube.com/results?search\\_query=TypeScript](https://www.youtube.com/results?search_query=TypeScript)
- [https://www.youtube.com/results?search\\_query=Angular](https://www.youtube.com/results?search_query=Angular)
- Tutos sur Google et Youtube entre autres.



Visual Studio Code



Firebase



TypeScript



# Introduction au langage de Programmation Orientée Objet TypeScript

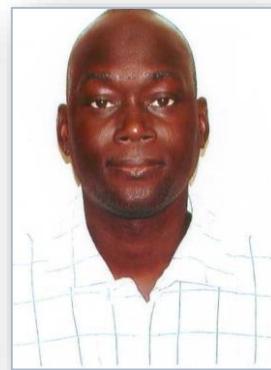
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)



# A propos de moi

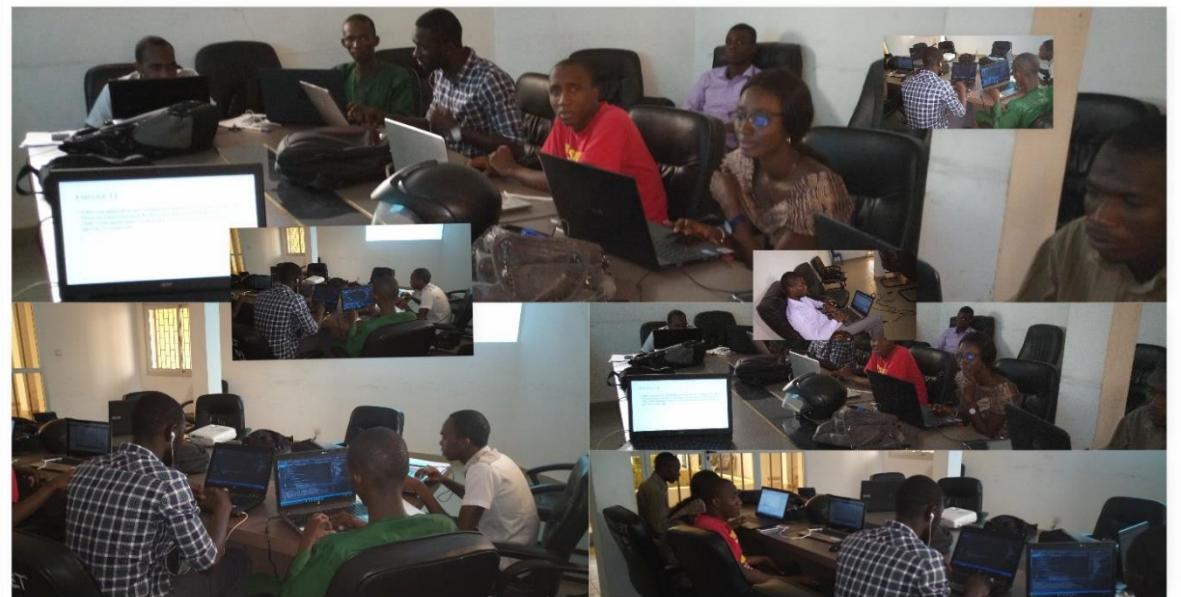


- Enseignant-Chercheur à l'UFR SET- Université de THIES  
<https://sites.google.com/a/univ-thies.sn/osall751/>
- Enseignements:
  - Algorithmique et Programmation(C, Java, PHP)
  - Programmation WEB dynamique(HTML 5 CSS, PHP, MySQL, CMS,...)
  - Programmation Java, Dart, TypeScript
  - Programmation Java, Jakarta EE, JSF, Spring, SpringBoot, Angular
  - Technologies Mobiles Android, Xamarin, Ionic, Flutter
  - Programmation .Net, C#
  - Gestion de Projet Informatique
  - Génie Logiciel, Qualité et Métrique du Logiciel
- Contact:
  - [osall@univ-thies.sn](mailto:osall@univ-thies.sn)
  - UFR SET, Université de THIES -Dpt Informatique, BP 967 THIES.



# Une sagesse chinoise...

*« J'écoute et j'oublie; je lis et je comprends; je fais et j'apprends »*  
[Proverbe chinois]

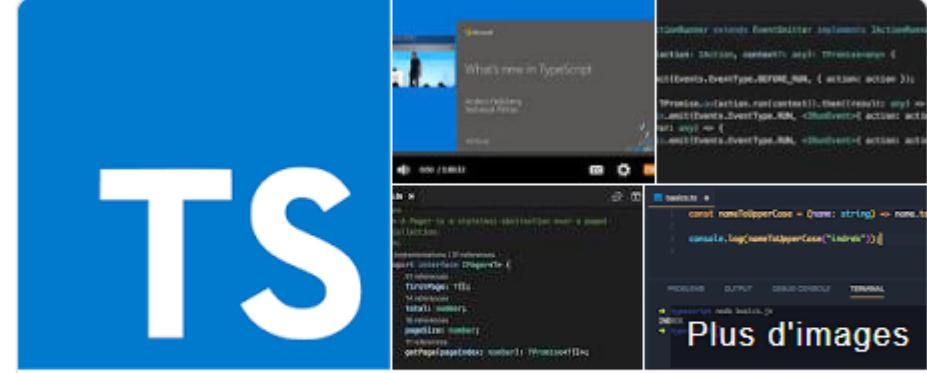


# Contenu

1. Introduction au Framework Angular et le langage TypeScript
- 2. Initiation au langage de programmation TypeScript**
3. Développement d'applications web avec le Framework Angular
4. Persistance de données et Interaction avec le Cloud
5. Déploiement sur Firebase Hosting

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



**TypeScript**  
Langage de programmation

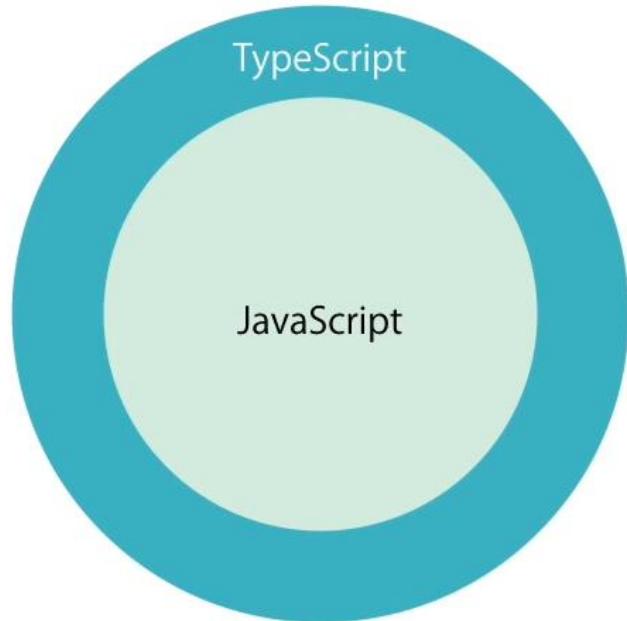
TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript.

[Wikipédia](#)

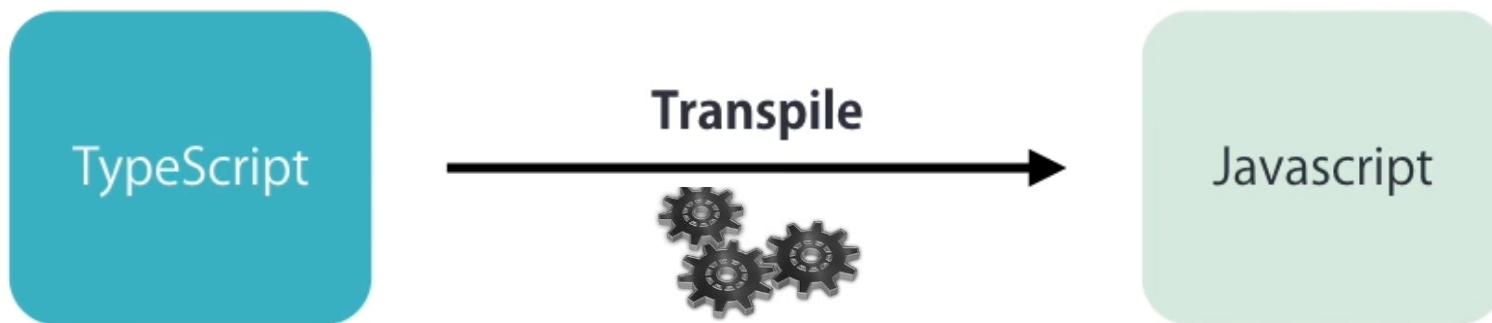
Conçu Par : [Microsoft](#)  
Extension de fichier : `ts`  
Date de première version : [9 février 2012](#)  
Dernière version : [3.2.2 \(29 novembre 2018\)](#)  
Typage : dynamique, faible, fort optionnel, statique optionnel  
Paradigme : Multi-paradigme

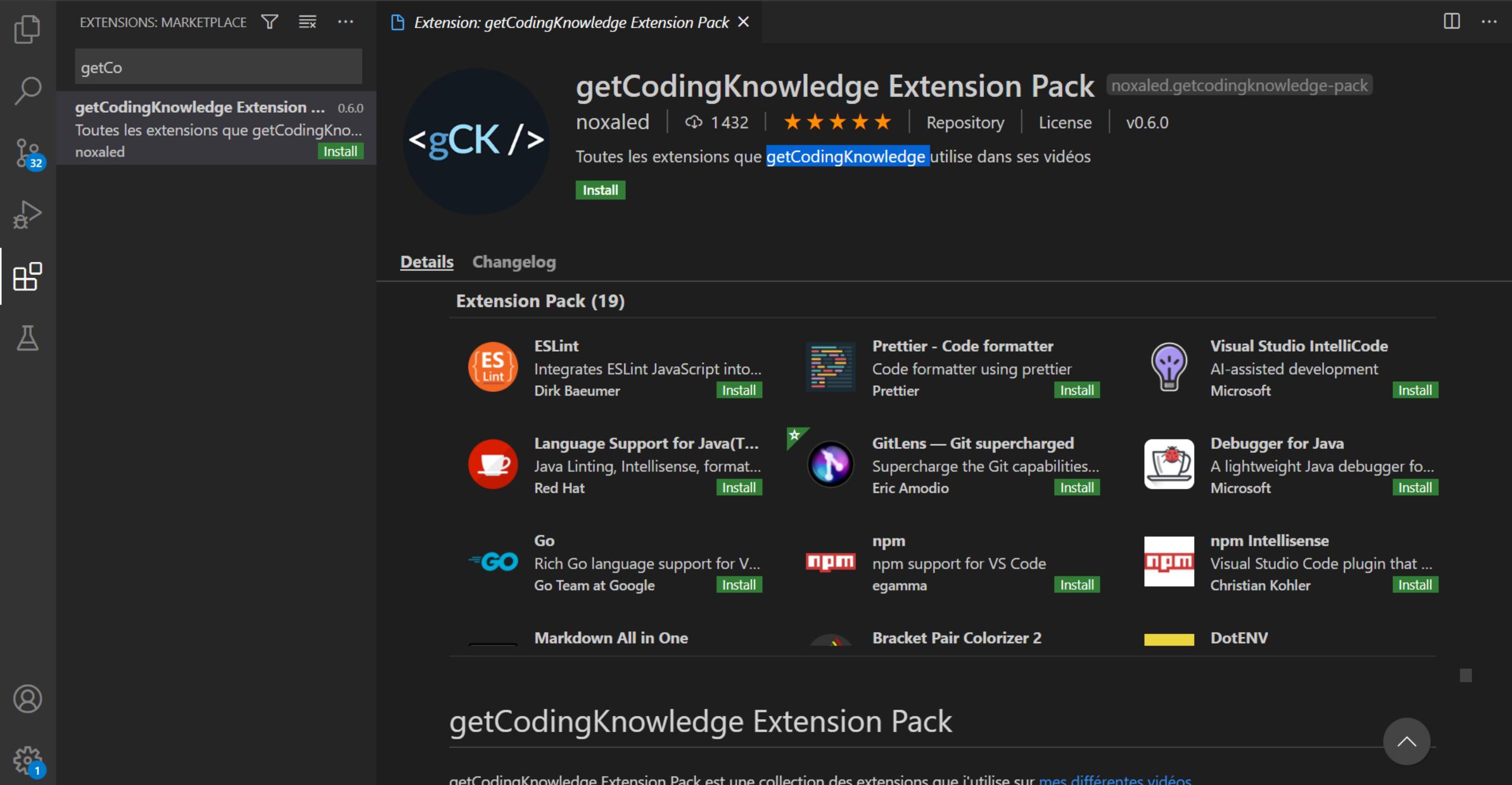
[Plus d'images](#)

# Initiation à TypeScript



- **TypeScript** est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript.
- C'est un **sur-ensemble** de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript).
- Le code TypeScript est **transcomplié** en JavaScript, pouvant ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript.





TypeScript [3.8](#) is now available, [3.9](#) has a release candidate to try.

# TypeScript is Typed JavaScript at Any Scale.

TypeScript extends JavaScript by adding types to the language.

TypeScript speeds up your development experience by catching errors and providing fixes before you even run your code.

Any browser, any OS, anywhere JavaScript runs.  
Entirely Open Source.

The screenshot shows a comparison between plain JavaScript and TypeScript. On the left, under 'Without TypeScript', is the following code:

```
JS Without TypeScript
function addPrices(items) {
  let sum = 0;
  for (const item of items) {
    sum += item;
  }
  return sum;
}
addPrices(3, 4, 6);
```

On the right, under 'TypeScript', is the same code with type annotations and an error message:

```
TS TypeScript
function addPrices(items: number[]) {
  let sum = 0;
  for (const item of items) {
    sum += item;
  }
  return sum;
}
addPrices(3, 4, 6);

Expected 1 argument,
but got 3.
```

A tooltip at the bottom left says 'Validate your function arguments'.

Next ➤

[Try in your browser](#)

[Install locally](#)

## What is TypeScript?



# Installer TypeScript

```
C:\Users\Developpeur>npm install -g typescript
C:\Users\Developpeur\AppData\Roaming\npm\tsserver -> C:\Users\Developpeur\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
C:\Users\Developpeur\AppData\Roaming\npm\tsc -> C:\Users\Developpeur\AppData\Roaming\npm\node_modules\typescript\bin\tsc
+ typescript@3.3.333
added 1 package from 1 contributor in 13.168s

C:\Users\Developpeur>cd Desktop

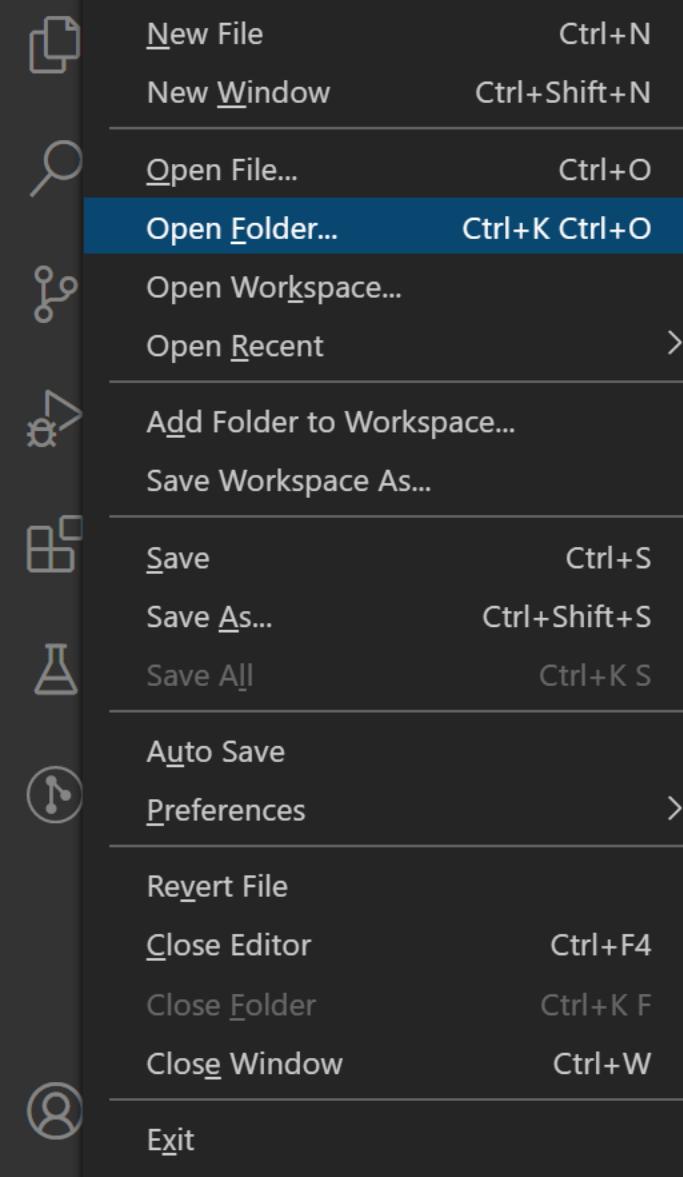
C:\Users\Developpeur\Desktop>cd AppIonic

C:\Users\Developpeur\Desktop\AppIonic>code exemple.ts
```

# Configuration du workSpace

- Créer dans le dossier de travail un sous-dossier nommé **exemple1** ou avec le nom de votre choix et ouvrez le dans visual studio code
- Créer un fichier **main.ts** et un fichier **index.html** où vous importerez le fichier **main.js**
- Lancer la commande de **transcompilation** dans le dossier **exemple1** avec l'option **-w**
- **Autre alternative à préférer: pour la configuration avec déploiement sur serveur et utilisation fichiers de configuration à la section avant les décorateurs**





## Customize

## Tools and languages

Install support for JavaScript, [Python](#), [Java](#), [PHP](#), [Azure](#), [Docker](#) and [more](#)

## Settings and keybindings

Install the settings and keyboard shortcuts of [Vim](#), [Sublime](#), [Atom](#) and [other editors](#)

## Color theme

Make the editor and your code look the way you love

## Learn

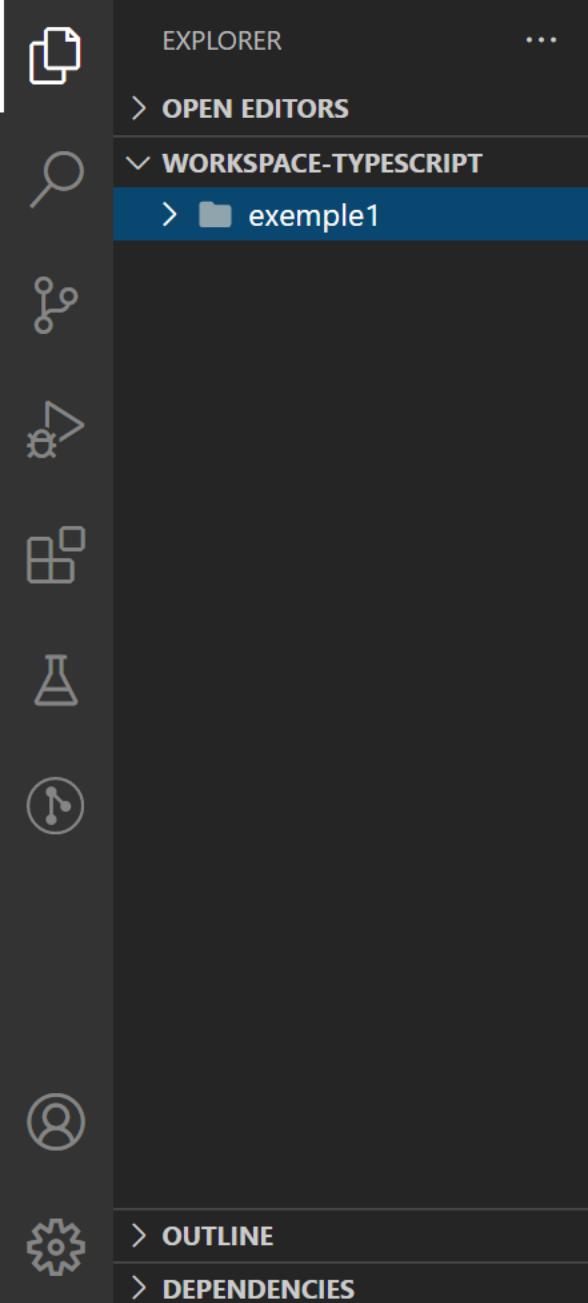
## Find and run all commands

Rapidly access and search commands from the Command Palette ([Ctrl+Shift+P](#))

## Interface overview

Get a visual overlay highlighting the major components of the UI

## Interactive playground



## Welcome

### Start

- New file
- Open folder...
- Add workspace folder...

### Recent

- test (Workspace) C:\Users\ousma\Desktop\workspace-types...
- premiereAppAngular C:\Users\ousma\Desktop\workspace-a...
- projetli C:\Users\ousma\Desktop\workspace-angular
- exemple1 C:\Users\ousma\Desktop\flutter\_exemples
- first\_app C:\Users\ousma\Desktop\flutter\_exemples
- More... (Ctrl+R)

### Help

- Printable keyboard cheatsheet
- Introductory videos
- Tips and Tricks
- Product documentation
- GitHub repository
- Stack Overflow

### Customize

#### Tools and languages

Install support for JavaScript, Python, Java, PHP, Azure, Do...

#### Settings and keybindings

Install the settings and keyboard shortcuts of Vim, Sublim...

#### Color theme

Make the editor and your code look the way you love

### Learn

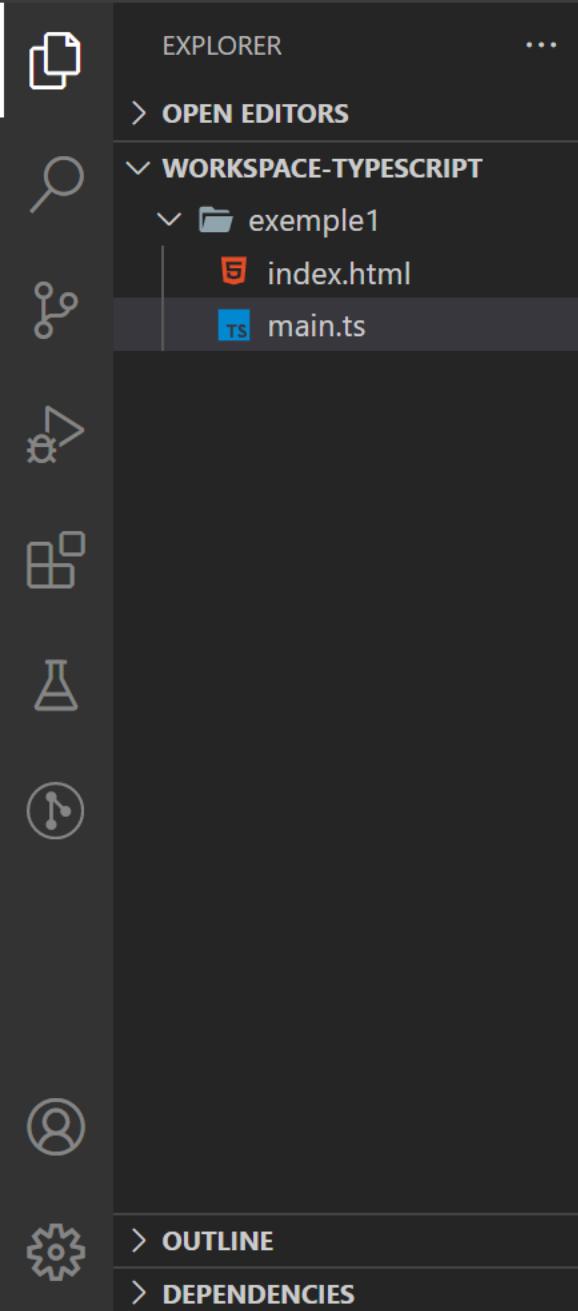
#### Find and run all commands

Rapidly access and search commands from the Command ...

#### Interface overview

Get a visual overlay highlighting the major components of...

#### Interactive playground



Welcome index.html main.ts

exemple1 > main.ts > ...

```
1 let message="Bienvenue, Hello from UT";
2 console.log(message);
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Sidebar (Left):**
  - EXPLORER: Shows WORKSPACES (example1) and OPEN EDITORS (index.html, main.js, main.ts).
  - SEARCH icon.
  - WORKSPACE icon.
  - OPEN EDITORS icon.
  - OUTLINE icon.
  - DEPENDENCIES icon.
- Editor Area (Top):** Welcome, index.html, main.ts (active tab). The main.ts file contains:

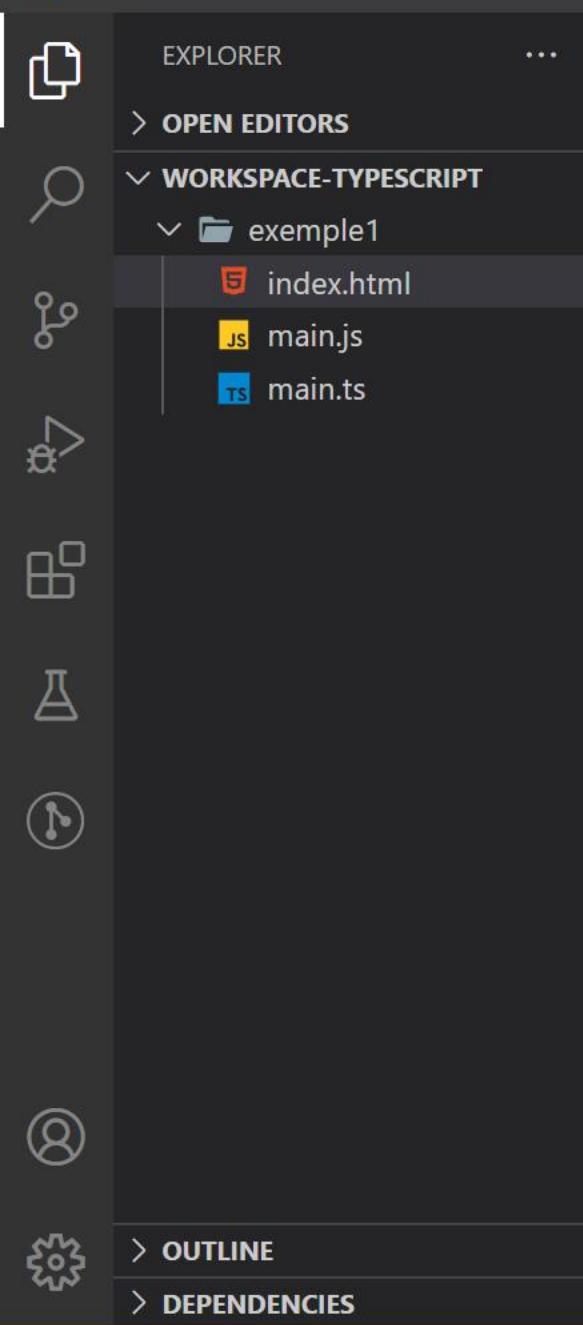
```
1 let message="Bienvenue, Hello from UT";
2 console.log(message);
```
- Bottom Editor Area (Bottom Left):** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (active tab).
- Terminal Area (Bottom Right):** 1: powershell. The terminal shows PowerShell commands:

```
PS C:\Users\ousma\Desktop\workspace-typescript> cd .\exemple1\_
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1>tsc .\main.ts
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1>dir
```

Répertoire : C:\Users\ousma\Desktop\workspace-typescript\exemple1

Mode	LastWriteTime	Length	Name
-a---	10/15/2020 2:40 PM	222	index.html
-a---	10/15/2020 2:43 PM	66	main.js
-a---	10/15/2020 2:42 PM	62	main.ts

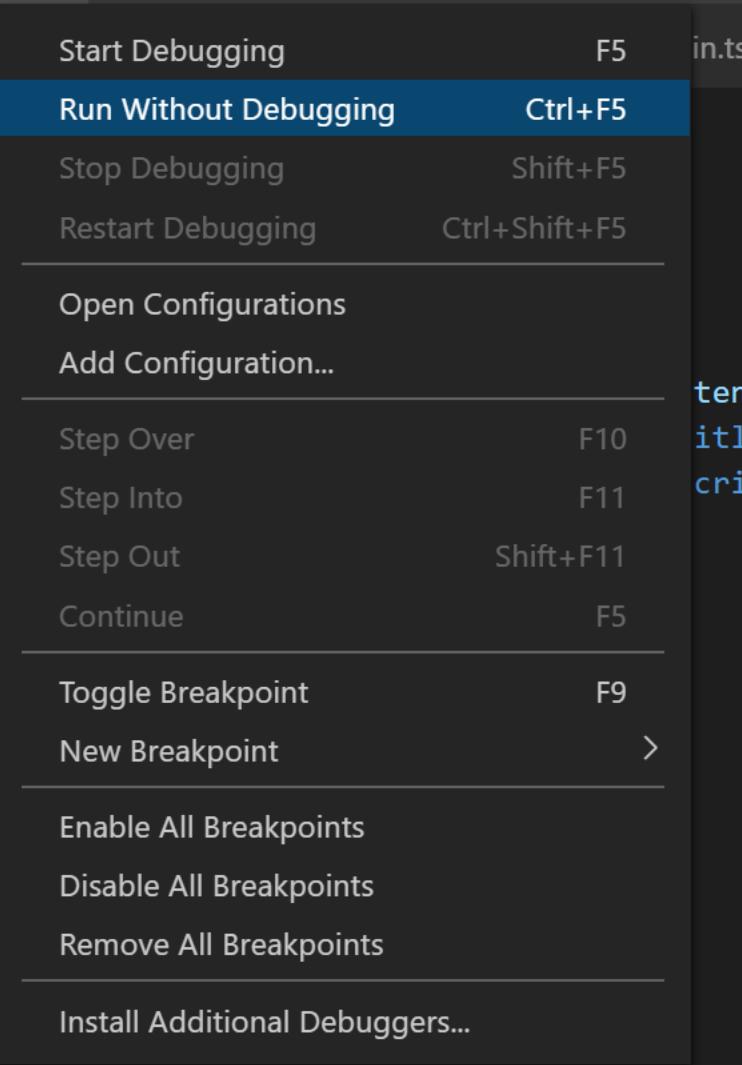
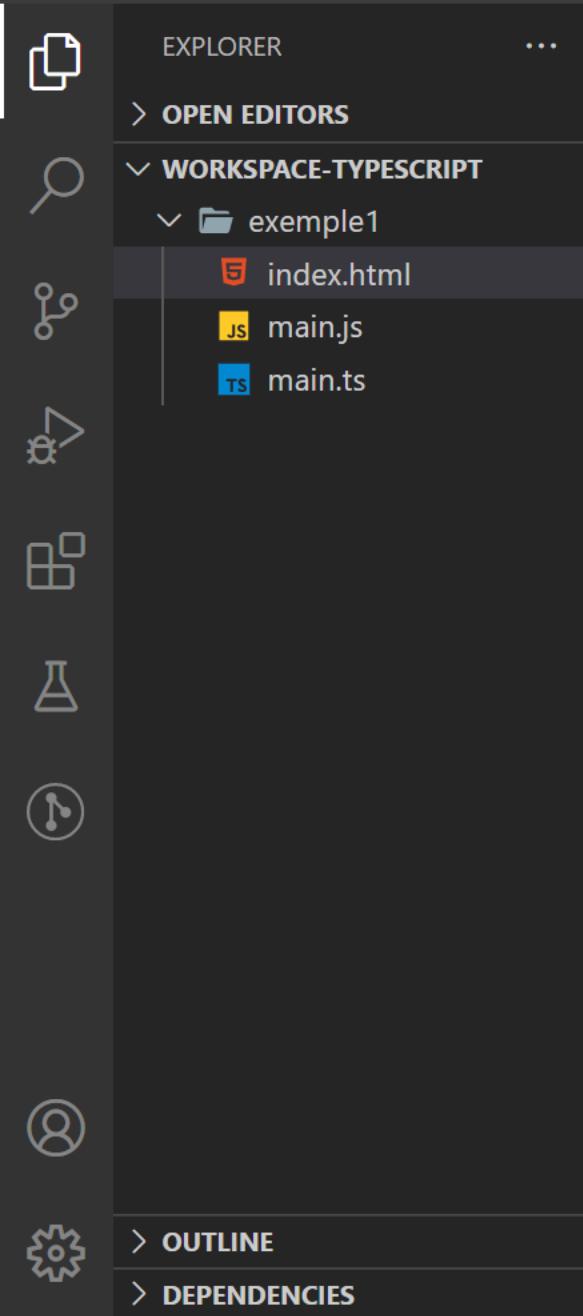
```
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> []
```
- Bottom Status Bar:** Compile Hero: Off, Ln 2, Col 22, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier, a few icons.



Welcome index.html X main.ts

exemple1 > index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Exemple de base</title>
8    <script src="main.js"></script>
9  </head>
10
11 <body>
12
13 </body>
14
15 </html>
```



```
tent="width=device-width, initial-scale=1.0">
title>
script>
```



EXPLORER

...

Select Environment

□ ...



&gt; OPEN EDITORS

WORKSPACE-TYPESCRIPT

exemple1

index.html

main.js

main.ts

Chrome (preview)

Dart &amp; Flutter

Edge: Launch

Java

Node.js

Node.js (preview)

More...

```
8 |     <script src="main.js"></script>
9 | </head>
10 |
11 |<body>
12 |
13 |</body>
14 |
```

Exemple de base



Fichier | C:/Users/ousma/Desktop/workspace-typescript/exemple1/index.html



top



Filter

Default levels ▾



Bienvenue, Hello from UT

main.js:2



&gt; OUTLINE

&gt; DEPENDENCIES





main.ts X

C: &gt; Users &gt; ousma &gt; Desktop &gt; workspace-typescript &gt; main.ts &gt; ...

```
1 var message="Hello from UT";
2 console.log(message);
3
```



PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: powershell



```
PS C:\Users\ousma\Desktop\workspace-typescript> code main.ts
PS C:\Users\ousma\Desktop\workspace-typescript> tsc .\main.ts
PS C:\Users\ousma\Desktop\workspace-typescript> dir
```



Répertoire : C:\Users\ousma\Desktop\workspace-typescript

Mode	LastWriteTime	Length	Name
-a---	10/15/2020 1:41 PM	55	main.js
-a---	10/15/2020 1:39 PM	53	main.ts





main.ts X

C: &gt; Users &gt; ousma &gt; Desktop &gt; workspace-typescript &gt; main.ts &gt; ...

```
1 var message="Hello from UT";
2 console.log(message);
3
```



PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: node



Répertoire : C:\Users\ousma\Desktop\workspace-typescript



Mode	LastWriteTime	Length	Name
---	-----	-----	-----
-a----	10/15/2020 1:41 PM	55	main.js
-a----	10/15/2020 1:39 PM	53	main.ts

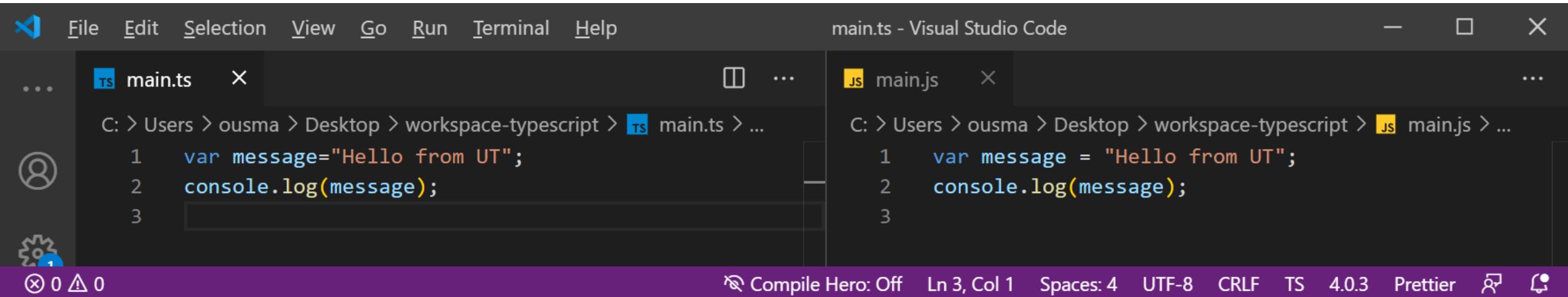
PS C:\Users\ousma\Desktop\workspace-typescript&gt; node .\main.js

Hello from UT

PS C:\Users\ousma\Desktop\workspace-typescript&gt;



# Le fichier JavaScript issu de la transcompilation du fichier TypeScript



The screenshot shows the Visual Studio Code interface with two open files:

- main.ts - Visual Studio Code**: A TypeScript file containing the following code:

```
1 var message="Hello from UT";
2 console.log(message);
3
```
- main.js - Visual Studio Code**: A JavaScript file generated by the transcompiler, containing the same code:

```
1 var message = "Hello from UT";
2 console.log(message);
3
```

The status bar at the bottom provides information about the files: "Compile Hero: Off", "Ln 3, Col 1", "Spaces: 4", "UTF-8", "CRLF", "TS 4.0.3", "Prettier", and icons for search, refresh, and other tools.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar (Explorer):**
  - OPEN EDITORS
  - WORKS...
  - exemple1
    - index.html
    - main.js
    - main.ts** (selected)
  - debug.log
- Center Area (Editor):** Welcome. The file main.ts contains the following code:

```
1 let message="Bienvenue, Hello from UT";
2 console.log(message);
```
- Bottom Navigation:** PROBLEMS, OUTPUT, TERMINAL, ... (TERMINAL is selected).
- Terminal:** 2: powershell. The command `tsc -w .\main.ts` is entered.
- Bottom Status Bar:** Compile Here: Off, Ln 2, Col 22, Spaces: 4, UTE 8, CRLF, TS 4.0.3, Brattain.
- Callout:** A red-bordered callout box highlights the text: "Lancer la commande **tsc** avec l'option **-w** qui fera que le fichier typecript sera automatiquement transcompilé à chaque fois que des changements seront détectés".

The screenshot shows the Visual Studio Code interface with the following components:

- Top Bar:** File, Edit, Selection, View, Go, ... main.ts - workspace-typescript - V...
- Left Sidebar:** Welcome, Recent files, Search, Repository, Issues (1), Folders, Problems, Terminal, Help.
- Code Editor:** A file named "main.ts" containing the following code:

```
1 let message="Bienvenue, Hello depuis UT en 2020";
2 console.log(message);
```
- Terminal:** A terminal window titled "node" showing logs of the TypeScript compiler:

```
[15:07:07] File change detected. Starting incremental compilation...
[15:07:07] Found 0 errors. Watching for file changes.
[15:08:55] File change detected. Starting incremental compilation...
```
- Output Panel:** An "Exemple de base" panel showing the output of the console log:

Console	Sources
Bienvenue, Hello depuis UT en 2020	main.js:2
- Bottom Status Bar:** Shows 0 errors, 0 warnings, Ln 1, Col 38, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier, and a bell icon.

# TypeScript

## Variables, Types de base et Opérateurs

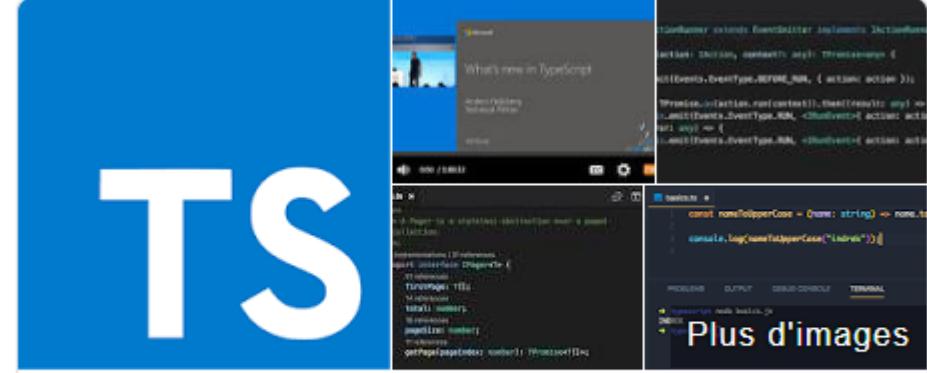
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- **Variables, Types de base et Opérateurs**
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



**TypeScript**  
Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)  
Extension de fichier : `ts`  
Date de première version : [9 février 2012](#)  
Dernière version : [3.2.2 \(29 novembre 2018\)](#)  
Typage : dynamique, faible, fort optionnel, statique optionnel  
Paradigme : Multi-paradigme

[Plus d'images](#)

# Initiation à TypeScript: déclaration des variables

- TypeScript suit les mêmes règles que JavaScript pour les déclarations de variable. Les variables peuvent être déclarées en utilisant: **var**, **let** et **const**.
- Déclaration avec **var** : Les variables dans TypeScript peuvent être déclarées à l'aide du mot clé **var**, comme dans JavaScript. Les règles de détermination de la portée restent les mêmes qu'en JavaScript.

```
function uneFontion():string{
    for (var index = 0; index < 10; index++) {
        console.log(index);
    }
    return "La dernière valeur de index est "+index;
}
```

(local var) index: number

# Initiation à TypeScript: Déclaration des variables

- Déclaration avec **let** et **const** pour résoudre les problèmes liés aux déclarations **var**, ES6 a introduit deux nouveaux types de déclarations de variable en JavaScript, à l'aide des mots-clés **let** et **const**. TypeScript, étant un sur-ensemble de JavaScript, prend également en charge ces nouveaux types de déclarations de variables.

A screenshot of a code editor showing a TypeScript function named 'uneFonction'. The code uses a 'let' declaration inside a for loop. A tooltip appears over the word 'index' with the text 'any' and the message 'Cannot find name 'index''. Below the tooltip, there are links for 'Peek Problem (Alt+F8)' and 'No quick fixes available'.

```
function uneFonction():string{
    for [let index = 0; index < 10; index++) {
        console.log(index);
    }
    return "La dernière valeur de index est "+index;
}
```

Contrairement aux variables déclarées avec **var**, les variables déclarées avec **let** ont une portée de bloc. Cela signifie que la portée des variables **let** est limitée à leur bloc conteneur, par ex. fonction, bloc **if else** ou bloc de la boucle.

# Initiation à TypeScript: les types de base

Pour écrire des programmes, nous devons travailler avec certaines des unités de données les plus simples: nombres, chaînes, structures, valeurs booléennes, etc. En TypeScript, les types de base sont les mêmes que ceux en JavaScript, avec un type d'énumération pratique ajouté pour faciliter les choses.

- Le fait de différencier le type d'une variable Dans un langage dit typé, on ne peut enregistrer qu'un type de valeur dans une variable. Typage en TS :

```
let variable: type;  
const unNombre: number = 12;
```

- **boolean** : Le type de données le plus élémentaire est la valeur simple **true/ false**, que JavaScript et TypeScript appellent une valeur booléenne. Exemple: `let bool: boolean = false;`

# Initiation à TypeScript: les types de base

number	1, 5.3, -10	Tous les nombres, pas de différenciation entre les entiers ou les flottants
string	'Hi', "Hi", `Hi`	Toutes les valeurs de texte
boolean	true, false	Juste ces deux-là, pas de valeurs "vraies" ou "faux"
object	{age: 30}	Tout objet JavaScript, des types plus spécifiques (type d'objet) sont possibles
Array	[1, 2, 3]	Tout tableau JavaScript, le type peut être dynamique ou strict (concernant les types d'éléments)
Tuple	[1, 2]	Tableau de longueur fixe
Enum	enum { NEW, OLD }	identificateurs de constantes globales automatiquement énumérés

# Types de base TypeScript

- Pour que les programmes soient utiles, nous devons être capables de travailler avec certaines des types de données : **boolean, number, string, array, tuple, enum, any, void, null and undefined, never, object**, etc. Dans TypeScript.
- Ce sont les mêmes types qu'en JavaScript.

# Initiation à TypeScript: les types de base

- **number** : Comme en JavaScript, tous les nombres en TypeScript sont de type float. Le type number en TypeScript supporte aussi les valeurs hexadécimales, les binaires et octales :

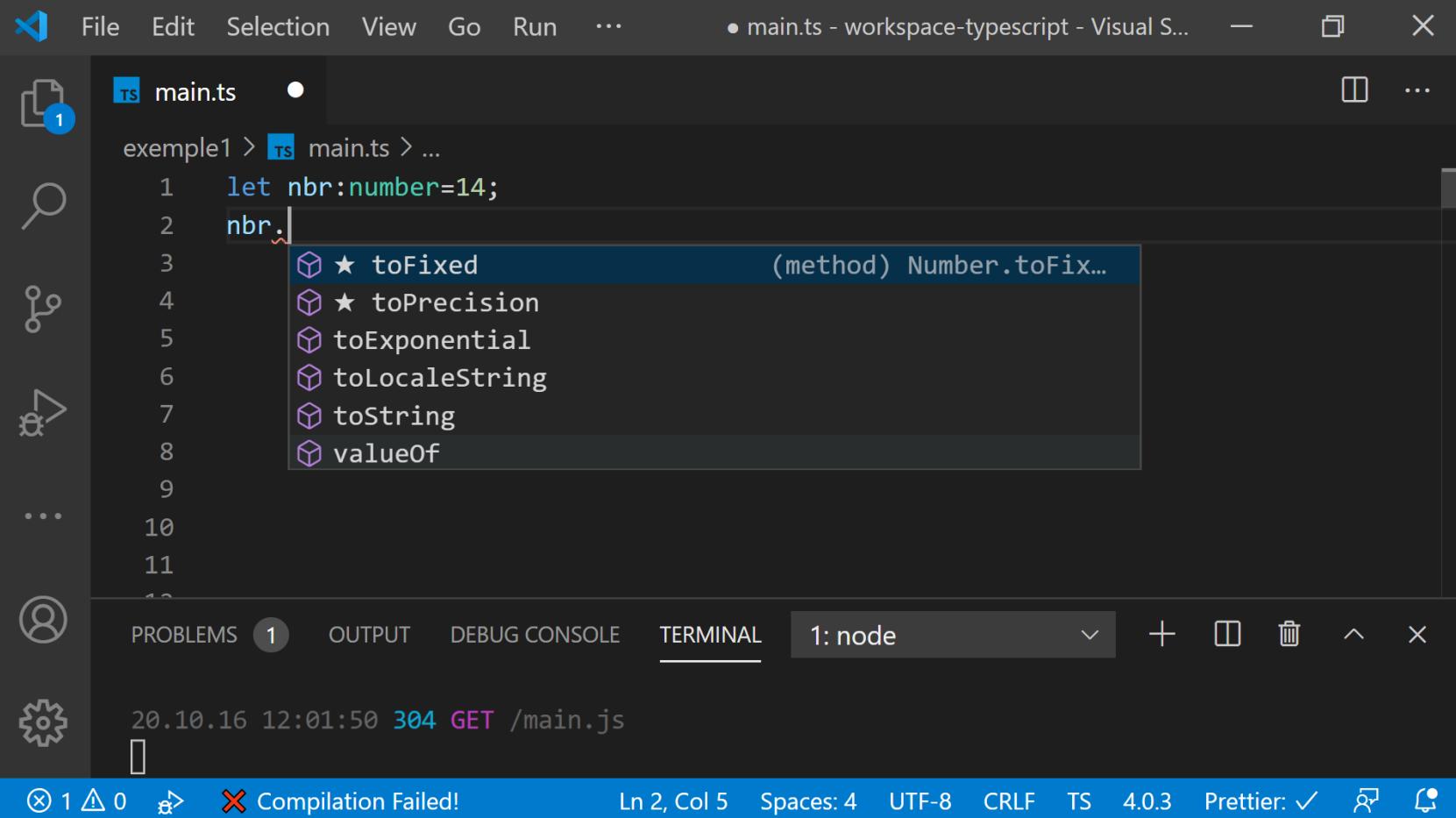
```
let valDecimal: number = 6;
let valHex: number = 0xf00d;
let valBinaire: number = 0b1010;
let valOctale: number = 0o744;
```

- **string** : Un autre type ultra commun, le type String. Il est utilisé dès que vous voulez travailler avec des données textes. En TypeScript, il est possible d'utiliser les double quotes ou les simples quotes pour définir une variable de type String.

```
let couleur: string = "bleue";
couleur = 'rouge';
```

# Initiation à TypeScript: les types de base

## Méthodes associées au type number



The screenshot shows a Visual Studio Code interface with a dark theme. The main editor window displays a file named "main.ts" containing the following code:

```
1  let nbr:number=14;
2  nbr.
```

A code completion dropdown menu is open at the cursor position, showing the following methods:

- toFixed
- toPrecision
- toExponential
- toLocaleString
- toString
- valueOf

The "toFixed" method is highlighted in blue. The status bar at the bottom indicates "Compilation Failed!".

# Initiation à TypeScript: les types de base

## Méthodes associées au type string

The screenshot shows a Visual Studio Code interface with a dark theme. The main editor window displays a file named `main.ts` containing the following code:

```
exemple1 > ts main.ts > ...
1 let str:string="Macky";
2 str.
```

A code completion dropdown menu is open at the cursor position, listing various methods available for the `string` type. The methods listed are:

- replace (method) String.replace
- split
- length
- toLowerCase
- match
- charAt
- charCodeAt
- concat
- indexOf
- lastIndexOf
- localeCompare
- search
- slice
- substr
- substring
- toLocaleLowerCase
- toLocaleUpperCase
- toString
- toUpperCase
- trim
- valueOf

The VS Code interface includes a sidebar with icons for file, search, and other tools. The bottom status bar shows the file path `1: node`, the status bar indicators, and the page number `107`.

# Interpolation de chaînes de caractères

- Pour mettre la valeur d'une expression dans une chaîne, utilisez **`$ {expression}`**. Si l'expression est un identifiant, vous pouvez omettre **`{}`**.
- Ces chaînes sont entourées du caractère backtick / backquote (`'`) AltGr+7 et les expressions incorporées sont de la forme **``${expr}`**.
- Voici quelques exemples d'utilisation de l'interpolation de chaîne:

---

## String

``${3 + 2}``

``${'word'.toUpperCase()}``

``$myObject``

## Résultat

5

WORD

The value of myObject.toString()

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar:** Icons for Welcome, index.html, main.ts (selected), and other workspace files.
- Code Editor:** Content of main.ts:

```
exemple1 > main.ts > ...
1 let nom="sall";
2 let prenom="Macky";
3 let age=58;
4 let message="Bienvenue, Hello depuis UT en 2020\n";
5 message=`Le président se nomme ${prenom} ${nom.toUpperCase()}.`;
6 message+=` Il est agé de ${age} ans`;
7 console.log(message);
8
```
- Terminal:** Shows logs from a node.js process:

```
[15:07:07] File change detected. Starting incremental compilation
[15:07:07] Found 0 errors. Watching for file changes.
[15:36:16] File change detected. Starting incremental compilation
[15:36:16] Found 0 errors. Watching for file changes.
```
- Output Panel:** A floating panel titled "Exemple de base" showing the output of the console log:

```
Fichier | C:/Users/ousma/Desktop/workspace-typescript/exemple...
Le président se nomme Macky SALL. Il est agé de 58 ans
main.js:7
```
- Bottom Status Bar:** Compile Hero: Off, Ln 6, Col 11, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier, and a few small icons.

# Initiation à TypeScript: les types de base

- Un **Tuple** permet de déclarer un tableau avec un nombre fixe d'éléments dont les types sont connus, mais n'ont pas besoin d'être les mêmes. Par exemple, vous souhaiterez peut-être représenter une valeur sous la forme d'une paire de chaîne et d'un nombre:

```
// Déclaration d'un tuple
let monTuple: [string, number];
// Initialisation
monTuple = ["UT", 2020]; // OK
// Initialisation incorrecte
monTuple = [2020, "UT"]; // Erreur
```

# Initiation à TypeScript: les types de base

- **enum** : Comme dans les langages objet comme C#, une enum est un moyen de définir un ensemble de constantes nommées.

```
enum Color {Red, Green, Blue}  
let c: Color = Color.Green;
```

```
enum Color {Red = 1, Green, Blue}  
let colorName: string = Color[2];  
alert(colorName); // Displays 'Green' as it's value is 2 above
```

- **any**: Normalement, vous avez utilisé ce type si vous développez en TypeScript. Il est possible d'utiliser ce type pour remplacer n'importe quel autre type:

```
let notSure: any = 4;  
notSure = "maybe a string instead";  
notSure = false; // okay, definitely a boolean
```

```
let liste: any[] = [1, true, "free"];  
liste[1] = 100;
```

# Initiation à TypeScript: les types de base

- **void** : Le type void est à l'opposé du type Any. Il sert à indiquer à une fonction qu'elle ne doit rien renvoyer:

```
function warnUser(): void {  
    alert("This is my warning message");  
}
```

- Ou à une variable qui ne peut que avoir **undefined** ou **null** comme valeur:

```
let unusable: void = undefined;
```

# Initiation à TypeScript: les types de base

## Types multiples avec Union

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is the sidebar with various icons. The main area has a tab bar with "Welcome", "index.html", and "main.ts". The "main.ts" tab is active, displaying the following code:

```
1 let str: string | number | boolean;
2 str="Macky";
3 console.log(str);
4 str=58;
5 console.log(str);
6 str=false;
7 console.log(str);
```

Below the code editor are tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" tab is selected, showing logs of file changes and compilation:

```
[15:07:07] File change detected. Starting incremental compilation...
[15:07:07] Found 0 errors. Watching for file changes.
[15:44:27] File change detected. Starting incremental compilation...
[15:44:27] Found 0 errors. Watching for file changes.
```

A modal window titled "Exemple de base" is open, showing the browser's developer tools with the "Console" tab selected. It displays the following log entries:

Log	File
Macky	main.js:3
58	main.js:5
false	main.js:7

# Vos propres types avec un alias de Type

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Sidebar Icons:** Document, Search, More, User, Settings.
- Code Editor:** A file named "main.ts" containing the following code:

```
exemple1 > main.ts > ...
1  type Point = {
2    x: number;
3    y: number;
4  };
5
6 // Déclaration d'un paramètre de type Point
7 function printCoord(pt: Point) {
8   console.log("L'abscisse est : " + pt.x);
9   console.log("L'ordonnée est: " + pt.y);
10 }
```
- Terminal:** An open terminal window titled "Exemple de base" showing the output of the code execution:

```
L'abscisse est : 100
L'ordonnée est: 100
```
- Status Bar:** CRLF, TS 4.0.3, Prettier: ✓.
- Message Bar:** Compilation Failed!
- Callout Box:** A red callout box highlights the explanatory text on the right side of the screen, which reads:

Nous avons utilisé des types d'objet et des types d'union en les écrivant directement dans les annotations de type. C'est pratique, mais il est courant de vouloir utiliser le même type plusieurs fois et de s'y référer par un seul nom. Un **alias** de type est exactement cela - un **nom** pour n'importe quel **type**.

# Opérateurs

1. Opérateurs d'incrémentation et de décrémentation ("**++**", "**--**")
2. Opérateurs binaires ("**+**", "**-**", "**/**", "**\***", "**%**", "**\*\***")
3. Opérateurs bit à bit ("**~**", "**&**", "**|**", "**^**", "**<<**", "**>>**")
4. Opérateurs de comparaison ("**==**", "**!=**", "**>**", "**<**", "**<=**", "**>=**")
5. Opérateurs logiques ("**&&**", "**||**", "**!**")
6. Opérateurs conditionnels ("**??**", "**?:**", "**??=**")

# L'opérateur conditionnel ??

- **expr1 ?? expr2**
- Si **expr1** est non-nulle, retourne sa valeur. Sinon **expr2** est évaluée et retournée

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active File:** main.ts - workspace-typescript - Visual Studio Code
- Code Editor:** Shows the following TypeScript code:

```
exemple1 > main.ts > ...
1 let str: string=null;
2 console.log(str??"Bonjour");
3 
```
- Terminal:** Shows the output: [15:44:27] File change detected. Starting incremental compilation.
- Console:** Shows the output: Bonjour (from main.js:2)
- Status Bar:** Compile Hero: Off, Ln 3, Col 1, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier: ✓, Notifications icon.

# L'opérateur d'affectation **??=**

- TypeScript propose des opérateurs pratiques pour traiter les valeurs qui peuvent être nulles. L'un est l'opérateur d'affectation **??=**, qui affecte une valeur à une variable uniquement si cette variable est actuellement nulle:

```
let a:number; // La valeur initiale de a est nulle.  
a ??= 3; console.log(a); // <-- Affiche 3.  
a ??= 5; console.log(a); // <-- Affiche toujours 3.
```

# L'opérateur d'affectation ??=

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar:** Welcome, index.html, main.ts (selected), and other icons for Search, Find, Open, and More.
- Code Editor:** The code file main.ts contains the following TypeScript code:

```
1 let a=5;
2 let b=7;
3 let d:number;
4 d??=a+b; //La somme sera assigée, la valeur de d étant nulle
5 console.log(d);
6
7 d??=a-b; //La somme ne sera assigée, d étant non nulle
8 console.log(d);
9
```
- Terminal:** Shows the command "node" and the message "[15:44:27] File change detected. Starting incremental compilation...".
- Output Panel:** Shows the browser console output with the value "12" at line 5 and line 7.
- Bottom Status Bar:** Includes icons for Profiler, Linting, and Compiler status, along with file information (Compile Hero: Off, Line 4, Col 52, Spaces: 4, CRLF, TS 4.0.3, Prettier: ✓).

# TypeScript

## Les Structures de Contrôle

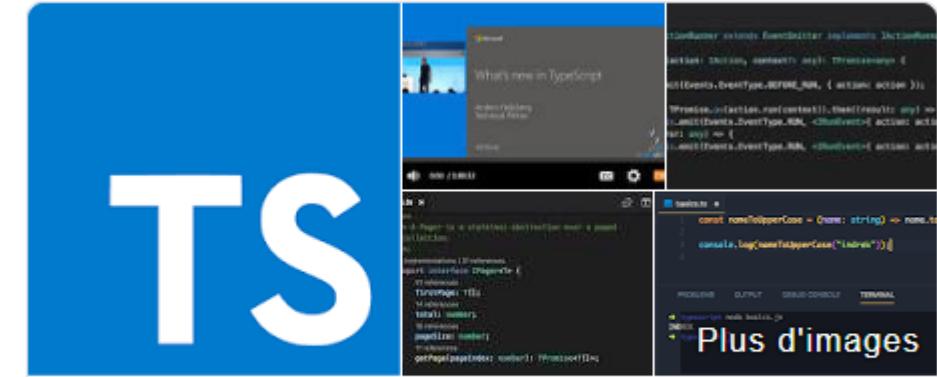
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- **Les Structures de Contrôle**
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



## TypeScript

Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript.  
[Wikipédia](#)

Conçu Par : [Microsoft](#)

Extension de fichier : `ts`

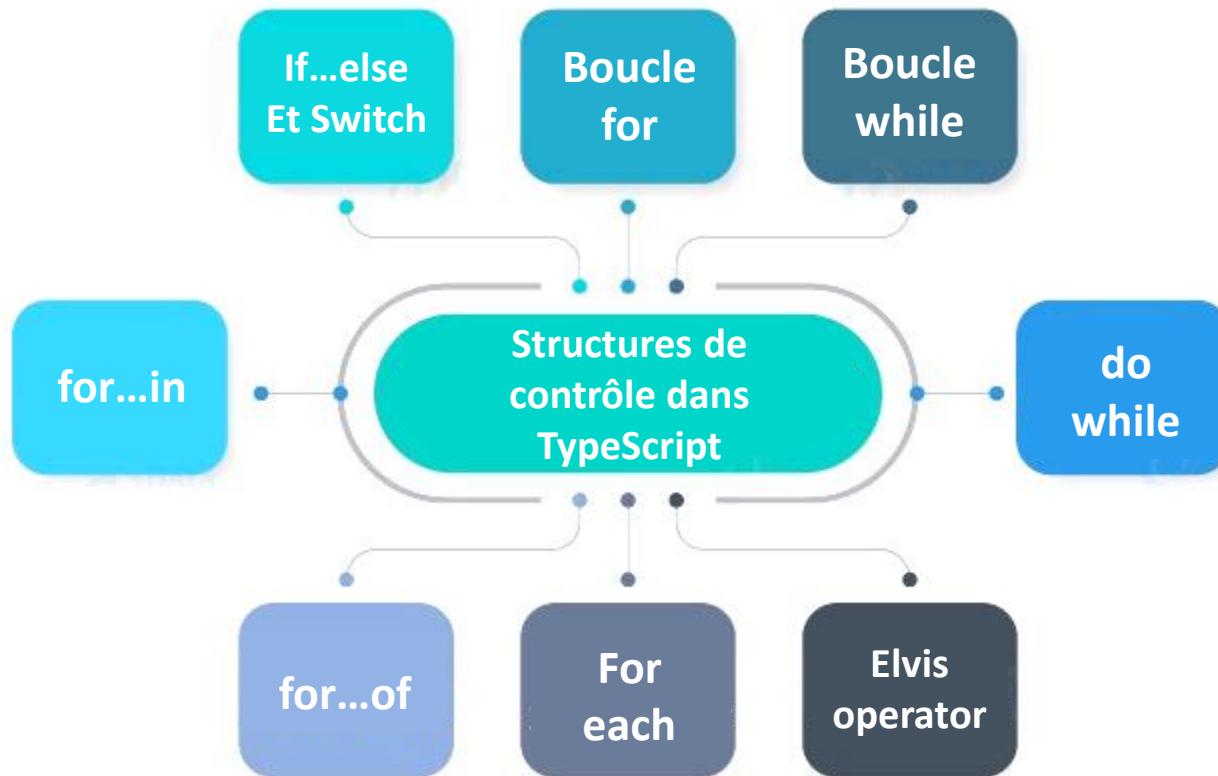
Date de première version : [9 février 2012](#)

Dernière version : [3.2.2 \(29 novembre 2018\)](#)

Typage : dynamique, faible, fort optionnel, statique optionnel

Paradigme : Multi-paradigme

# Les Structures de Contrôle dans TypeScript



# Initiation à TypeScript: **les structures conditionnelles : if**

File Edit Selection View Go Run ... main.ts - workspace-typescript - Visual St... — ×

Welcome index.html

exemple1 > main.ts > ...

```
1 let x=7;
2 let y=10;
3 if (x>y) {
4     console.log(x+" est supérieur à "+y);
5 } else if(x<y){
6     console.log(x+" est inférieur à "+y);
7 }else{
8     console.log(x+" est égale à "+y);
9 }
```

Exemple de base

Fichier | C:/Users/ousma/Desktop/Exemple de base/main.ts

Elements Console

7 est inférieur à 10 main.js:7

Compile Hero: Off Ln 9, Col 2 Spaces: 4 UTF-8 CRLF TS 4.0.3 Prettier: ✓

# Initiation à TypeScript : les structures conditionnelles : elvis operator

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Active File:** main.ts - workspace-typescript - Visual St...
- Code Editor:** Shows the following TypeScript code:

```
exemple1 > main.ts > [o] y
1 let x = 70;
2 let y = 10;
3 x > y
4 ? console.log(x + " est supérieur à " + y)
5 : x < y
6 ? console.log(x + " est inférieur à " + y)
7 : console.log(x + " est égale à " + y);
```
- Console Tab:** Exemple de base
- Console Output:** 70 est supérieur à 10
- Status Bar:** Compile Hero: Off, Ln 2, Col 11, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier: ✓, 123

# Initiation à TypeScript : les structures conditionnelles : switch

The screenshot shows a Visual Studio Code interface. On the left is a sidebar with icons for file operations, search, navigation, and other development tools. The main editor area displays a TypeScript file named 'main.ts' with the following code:

```
1 let x = 70;
2 let y = 70;
3 switch (x > y) {
4     case true:
5         console.log(x + " est supérieur à " + y);
6         break;
7     case false:
8         if (x == y) {
9             console.log(x + " est égale à " + y);
10        } else {
11            console.log(x + " est inférieur à " + y);
12        }
13        break;
14    default:
15        break;
16 }
```

A tooltip window titled 'Exemple de base' is open over the 'case true:' line, showing the output of the code: '70 est égale à 70'. The status bar at the bottom provides information about the file, including 'Compile Hero: Off', 'Ln 3, Col 17', and 'Spaces: 4'.

# Initiation à TypeScript : les structures itératives : for, for...of, for...in

The screenshot shows a Visual Studio Code interface with two main panes. The left pane displays a TypeScript file named `main.ts` containing three examples of iterative loops:

```
exemple1 > TS main.ts > ...
1  console.log("### Exemple avec la boucle for classique");
2  for (let i = 0; i < 5; i++) {
3      console.log("Valeur de i est "+i);
4  }
5
6
7  console.log("###Exemple avec la boucle for..of");
8  let tab=[10, "Moussa",845.22, false];
9  for (const elt of tab) {
10     console.log(elt);
11 }
12
13 console.log("###Exemple avec la boucle for..in");
14 for (const key in tab) {
15     if (Object.prototype.hasOwnProperty.call(tab, key)) [
16         const element = tab[key];
17         console.log("Index courant du tableau: "+key);
18         console.log("\tContenu: "+element);
19     ]
20 }
```

The right pane shows the output of the code execution in the terminal, which is also titled `main.ts - workspace-typescript - Visual Studio Code`. The terminal output is as follows:

```
Exemple de base
Fichier | C:/Users/ousma/Desktop/works...
+---+
| Exemple de base |
| Fichier | C:/Users/ousma/Desktop/works...
| +---+
| | Exemple de base |
| | Fichier | C:/Users/ousma/Desktop/works...
| | +---+
| | | Exemple avec la boucle for classique main.js:1
| | | Valeur de i est 0 main.js:3
| | | Valeur de i est 1 main.js:3
| | | Valeur de i est 2 main.js:3
| | | Valeur de i est 3 main.js:3
| | | Valeur de i est 4 main.js:3
| | | Exemple avec la boucle for..of main.js:5
| | | 10 main.js:9
| | | Moussa main.js:9
| | | 845.22 main.js:9
| | | false main.js:9
| | | Exemple avec la boucle for..in main.js:11
| | | Index courant du tableau: 0 main.js:15
| | | Contenu: 10 main.js:16
| | | Index courant du tableau: 1 main.js:15
| | | Contenu: Moussa main.js:16
| | | Index courant du tableau: 2 main.js:15
| | | Contenu: 845.22 main.js:16
| | | Index courant du tableau: 3 main.js:15
| | | Contenu: false main.js:16
| +---+
```

At the bottom of the interface, there are status indicators and a page number:

Prof. Ousmane Sow ① 0 ⚡ 0 ⚡ 125

Compile Hero: Off Ln 18, Col 44 Spaces: 4 UTF-8 CRLF TS 4.0.3 Prettier: ✓ ⚡ 🔔

# Initiation à TypeScript : les structures itératives : for...each

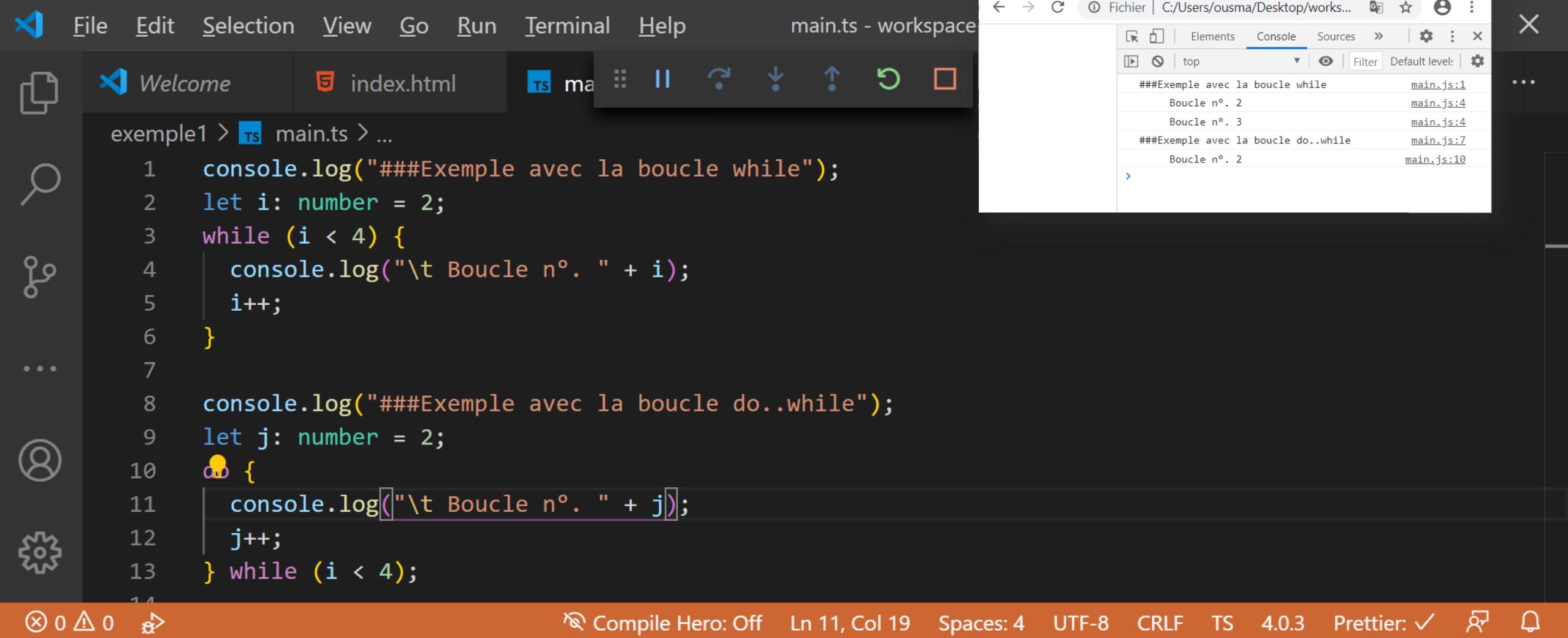
The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Tab Bar:** Welcome, index.html, main.ts (highlighted).
- Code Editor:** A TypeScript code example named main.ts:

```
exemple1 > main.ts > ...
1 let tab = [10, "Moussa", 845.22, false];
2 console.log("###Exemple avec la boucle for..each");
3 tab.forEach((element) => {
4   console.log(element);
5});
```
- Compiler Status:** Compile Hero: Off, Ln 5, Col 1.
- Console:** An open terminal window titled "Exemple de base" showing the execution results:

Output	Source
###Exemple avec la boucle for..each	main.js:10
10	main.js:12
Moussa	main.js:12
845.22	main.js:12
false	main.js:12
>	

# Initiation à TypeScript: les structures itératives : while et do...while



The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, there's a sidebar with various icons. The main area has a dark theme with a light background for the code editor. The title bar says "main.ts - workspace". In the editor, there are two sections of code:

```
1  console.log("###Exemple avec la boucle while");
2  let i: number = 2;
3  while (i < 4) {
4      console.log("\t Boucle n°. " + i);
5      i++;
6  }
7
8  console.log("###Exemple avec la boucle do..while");
9  let j: number = 2;
10 do {
11     console.log("\t Boucle n°. " + j);
12     j++;
13 } while (i < 4);
```

On the right, there's a browser developer tools window titled "Exemple de base". It has tabs for "Elements", "Console", and "Sources", with "Console" selected. The "Console" tab shows the output of the code:

```
Exemple avec la boucle while
Boucle n°. 2
Boucle n°. 3
Exemple avec la boucle do..while
Boucle n°. 2
```

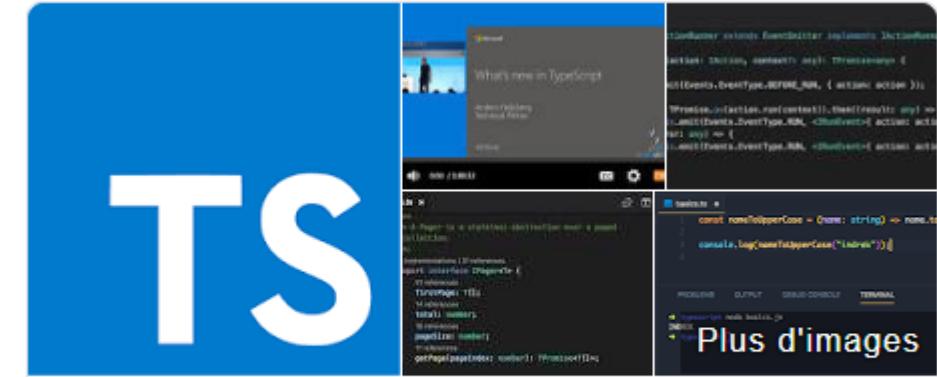
At the bottom of the screen, there's a status bar with the following information: "Compile Hero: Off", "Ln 11, Col 19", "Spaces: 4", "UTF-8", "CRLF", "TS 4.0.3", "Prettier: ✓", and some icons.

# TypeScript

## Les Fonctions

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- **Les Fonctions**
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



## TypeScript

Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)

Extension de fichier : `ts`

Date de première version : [9 février 2012](#)

Dernière version : [3.2.2 \(29 novembre 2018\)](#)

Typage : dynamique, faible, fort optionnel, statique optionnel

Paradigme : Multi-paradigme

# Initiation à TypeScript: les fonctions

- Séquence d'instructions réalisant un calcul ou une tâche.
- Permet de découper un problème global en plusieurs éléments plus simples et réutilisables.
- La syntaxe suivante est une des manières de définir une fonction en TypeScript

```
function name(params:type):type {  
    return someThing;  
}
```

# Initiation à TypeScript: Paramètres de fonctions

```
function sommme(x:number, y:number):number {  
    return x+y;  
}  
let val=sommme(10, 50);
```

```
function sommme(x, y){  
    return x+y;  
}  
let val=sommme(10, 50);
```

Nous pouvons ajouter des types à chacun des paramètres, puis à la fonction elle-même un type de retour. TypeScript peut inférer le type de retour en consultant les instructions de retour, nous pouvons donc également laisser cette option facultative dans de nombreux cas.

# Initiation à TypeScript: Paramètres de fonctions optionnels avec "?"

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Sidebar:** Welcome, index.html, main.ts (selected), and other icons for recent files, settings, and user profile.
- Code Editor:** The file main.ts contains the following code:

```
1  function saluer(nom?: string) {  
2    if (nom === undefined) return "Bonjour vous sans nom";  
3    return "Bonjour " + nom;  
4  }  
5  
6  console.log(saluer());  
7  console.log(saluer("10 000 Problèmes !!!"));
```
- Terminal:** Shows the output of the code execution:

```
Bonjour vous sans nom  
Bonjour 10 000 Problèmes !!!
```
- Status Bar:** Compile Hero: Off Ln 12.

# Initiation à TypeScript: Paramètres de fonctions avec valeur par défaut

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `main.ts`. The code is as follows:

```
exemple1 > TS main.ts > ...
1  function saluer(nom = "Jonh Doe"): string {
2  |   return "Bonjour " + nom;
3  }
4
5  console.log(saluer());
6  console.log(saluer("10 000 Problèmes !!!"));
7
```

- Output Panel:** A browser window titled "Exemple de base" shows the results of running the code:
  - Line 1: Bonjour Jonh Doe (from the first log statement)
  - Line 2: Bonjour 10 000 Problèmes !!! (from the second log statement)
- Bottom Status Bar:** Shows "Compile Hero: Off" and "Ln".

# Initiation à TypeScript: **Surcharge de fonctions**

- Fonctions portant le même nom avec un nombre ou des types de paramètres différents
- TypeScript ne permet qu'une surcharge de la définition pour un corps fonction unique
- Dans l'exemple, nous avons la même fonction ajouter() avec deux déclarations de fonction et une implémentation de fonction. La première signature a deux paramètres de type string, tandis que la seconde signature a deux paramètres du type number.
- La dernière fonction doit avoir l'implémentation de la fonction. Étant donné que le type de retour peut être une chaîne ou un nombre selon les deux premières déclarations de fonction, nous devons utiliser des paramètres compatibles et un type de retour comme n'importe lequel dans la définition de la fonction.
- La surcharge de fonction avec un nombre différent de paramètres et de types avec le même nom n'est pas prise en charge.

# Initiation à TypeScript: Surcharge de fonctions

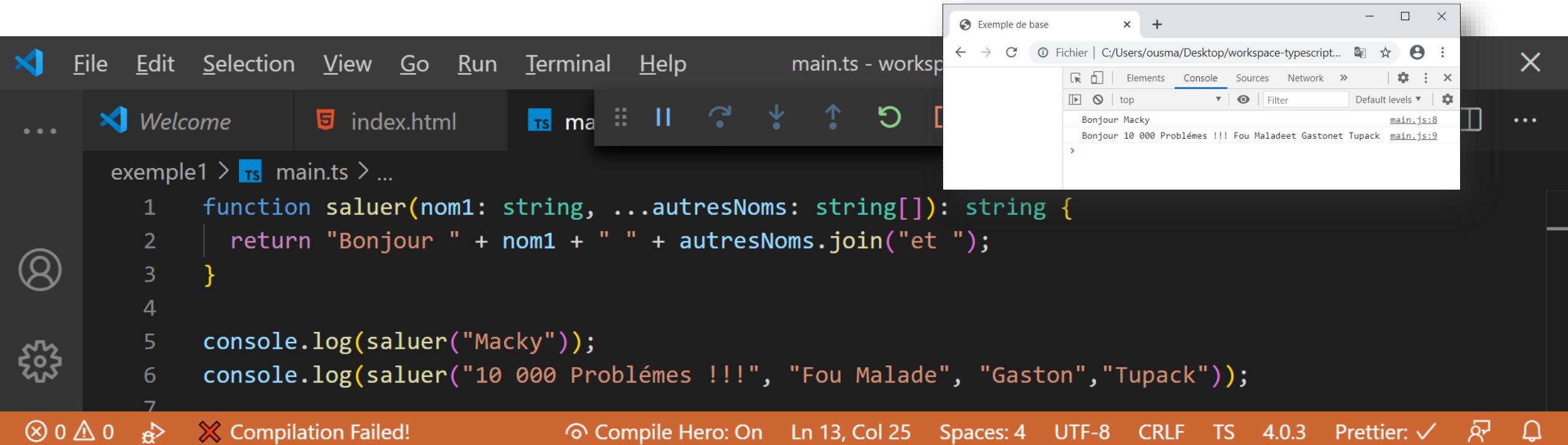
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Left Sidebar:** Welcome, index.html, main.ts (with a blue badge '1'), and several other icons for file management.
- Code Editor:** A file named "main.ts" containing the following code:

```
1  function ajouter(a: string, b: string): string;
2
3  function ajouter(a:number, b:number): number;
4
5  function ajouter(a: any, b:any): any {
6    console.log(a+b);
7    return a + b;
8  }
9
10 ajouter("Hello ", " Ousmane"); // retourne "Hello Ousmane"
11 ajouter(10, 20); // retourne 30
```
- Terminal:** An open terminal window titled "Exemple de base" showing the output:

```
Exemple de base
Fichier | C:/Users/ousma/Desktop/works...
Hello Ousmane
30
```
- Status Bar:** Shows the file is saved ("4"), encoding is UTF-8, line endings are CRLF, version is 4.0.3, Prettier is enabled, and icons for file operations like copy, paste, and refresh.
- Callout:** A red callout box highlights the function declarations in lines 1-3 and contains the following text:
  - Dans l'exemple, nous avons la même fonction ajouter() avec deux déclarations de fonction et une implémentation de fonction. La première signature a deux paramètres de type string, tandis que la seconde signature a deux paramètres du type number.
  - La dernière fonction doit avoir l'implémentation de la fonction. Étant donné que le type de retour peut être une chaîne ou un nombre selon les deux premières déclarations de fonction, nous devons utiliser des paramètres compatibles et un type de retour comme n'importe lequel dans la définition de la fonction.

# Initiation à TypeScript: Nombre de paramètres de fonctions indéfini ou Rest Parameters



The screenshot shows the Visual Studio Code interface. On the left, there's a sidebar with icons for user, settings, and extensions. The main area has a dark theme. A tab bar at the top includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. Below it, another tab bar shows 'Welcome', 'index.html', and 'main.ts - workspace'. The code editor contains the following TypeScript code:

```
exemple1 > TS main.ts > ...
1  function saluer(nom1: string, ...autresNoms: string[]): string {
2    return "Bonjour " + nom1 + " " + autresNoms.join("et ");
3  }
4
5  console.log(saluer("Macky"));
6  console.log(saluer("10 000 Problèmes !!!", "Fou Malade", "Gaston", "Tupack"));
```

At the bottom, a status bar shows 'Compilation Failed!' with a red X icon. Other status indicators include 'Compile Hero: On', 'Ln 13, Col 25', 'Spaces: 4', 'UTF-8', 'CRLF', 'TS 4.0.3', 'Prettier: ✓', and icons for search, refresh, and notifications.

To the right, a browser window titled 'Exemple de base' shows the output in the 'Console' tab. It displays two log entries:

```
Bonjour Macky
Bonjour 10 000 Problèmes !!! Fou Malade et Gaston et Tupack
```

Similaires aux arguments variables en Java. Les paramètres de Rest ne limitent pas le nombre de valeurs que vous pouvez transmettre à une fonction. Cependant, les valeurs transmises doivent toutes être du même type. Pour déclarer un paramètre de Rest, le nom du paramètre est précédé de trois points. Tout paramètre non rest doit précéder le paramètre rest.

# Initiation à TypeScript: les fonctions fléchées ou Lambda Functions

- Une expression de fonction fléchée (arrow function en anglais) permet d'avoir une syntaxe plus courte que les expressions de fonction et ne possède pas ses propres valeurs pour this, arguments, super, ou new.target.
- Les fonctions fléchées sont souvent anonymes et ne sont pas destinées à être utilisées pour déclarer des méthodes.

The screenshot shows the Visual Studio Code interface with a TypeScript file named 'main.ts' open. The code defines a function 'ajouter' that takes two parameters 'a' and 'b' of type 'any' and returns their sum. It then uses this function to log "Hello Ousmane" and the number 30 to the browser's developer tools console. The browser window displays the results: "Hello Ousmane" and "30".

```
File Edit Selection View Go Run Terminal Help main.ts - workspace-typescript - Vis... Welcome index.html ... exemple1 > main.ts > ...
1 let ajouter=(a: any, b:any) => a + b;
2
3 let str=ajouter("Hello ", " Ousmane"); // retourne "Hello Ousmane"
4 console.log(str);
5
6 str=ajouter(10, 20); // retourne 30
7 console.log(str);
```

Fichier | C:/Users/ousma/Desktop/workspace-typescript... Exemple de base Fichier | C:/Users/ousma/Desktop/workspace-typescript... Exemple de base

Console Sources Network Default levels

Hello Ousmane main.js:3  
30 main.js:5

137

# Initiation à TypeScript: les fonctions fléchées ou Lambda Functions

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active File:** main.ts - workspace-typescript - Visual Studio Code.
- Code Editor:** The code is as follows:

```
exemple1 > [ts] main.ts > ...
1 let saluer = (nom1: string, ...autresNoms: string[]) =>
2   "Bonjour " + nom1 + " " + autresNoms.join("et ");
3
4 console.log(saluer("Macky"));
5 console.log(saluer("10 000 Problèmes !!!", "Fou Malade", "Gaston", "Tupack"));
```
- Terminal:** Shows the output of the TypeScript compilation:

```
Exemple de base
Fichier | C:/Users/ousma/Desktop/workspace-typescript...
Elements Console Sources Network
top
Bonjour Macky
Bonjour 10 000 Problèmes !!! Fou Maladeet Gastonet Tupack main.js:9
```
- Status Bar:** Compilation Failed! (highlighted in orange), RLF, TS 4.0.3, Prettier: ✓.
- Bottom Status:** Prof. Ousmane SALL, Univ. Thiès, SN and page number 138.

# Spécificités de l'utilisation du caractère \_ dans les arguments de fonctions

- Le caractère **underscore** dans les arguments est une indication que vous n'allez pas utiliser ce paramètre à l'intérieur du bloc, c'est juste un bon moyen d'écrire du code, par exemple:

```
Fuction methode(utile:number, inutile:number) {  
    // dit que je n'utilisera que «utile» dans ce bloc  
}
```

- Le code ci-dessus peut également être écrit comme suit:

```
method (utile:number, _) {  
    // L'utilisation de '_' signifie que je ne vais pas utiliser le deuxième paramètre du bloc  
}
```

- builder: (\_ ,compteur, \_\_ ) => Traductions (counter.value), signifie que vous avez 3 paramètres \_, compteur et \_\_, et seul le compteur est ce que vous utilisez, donc les 1er et 3e paramètres sont indiqués par \_ et \_\_. C'est juste une manière plus propre d'écrire du code.

# Spécificités de l'utilisation du caractère \_ dans les arguments de fonctions

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar:** Shows icons for User, Settings, and Extensions.
- Current File:** main.ts (selected tab).
- Code Content:**

```
exemple1 > main.ts > ...
1  function saluer(nom:string, _): string {
2    return "Bonjour " + nom;
3  }
4
5  console.log(saluer("Ousmane", 0));
```
- Status Bar:** Compilation Failed! (highlighted in orange).
- Terminal:** Exemple de base window showing the output:

```
< -> C Fichier | C:/Users/ousma/Desktop/workspace-ty...
Elements Console Sources >
top
Bonjour Ousmane
main.js:4
```
- Bottom Status Bar:** CRLF, TS 4.0.3, Prettier: ✓.

# TypeScript

## Les tableaux et Collections

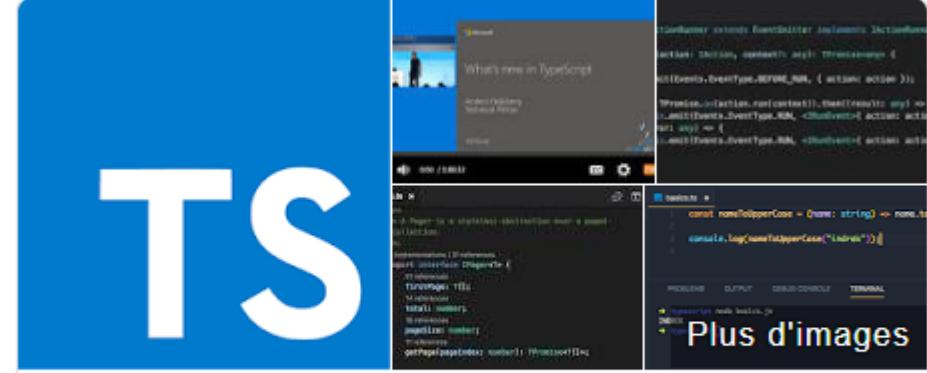
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- **Les tableaux et Collections**
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



**TypeScript**  
Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)  
Extension de fichier : `ts`  
Date de première version : [9 février 2012](#)  
Dernière version : [3.2.2 \(29 novembre 2018\)](#)  
Typage : dynamique, faible, fort optionnel, statique optionnel  
Paradigme : Multi-paradigme

[Plus d'images](#)

# Les Collections

- Une collection est un objet qui représente un groupe d'objets, appelés éléments. Les itérables sont une sorte de collection.
- Une collection peut être vide ou contenir de nombreux éléments. Selon l'objectif, les collections peuvent avoir différentes structures et implémentations. Voici quelques-uns des types de collections les plus courants:
  - **List**: Utilisé pour lire les éléments par leurs index.
  - **Set**: Utilisé pour contenir des éléments qui ne peuvent apparaître qu'une seule fois.
  - **Map**: Utilisé pour lire des éléments à l'aide d'une clé.

# Initiation à TypeScript: les tableaux avec [] et ou `Array<elemType>`

- **Array** : Comme en JavaScript, les tableaux en TypeScript sont définis par `[]`. Il y a également un autre moyen de les définir avec **Array<type>**:

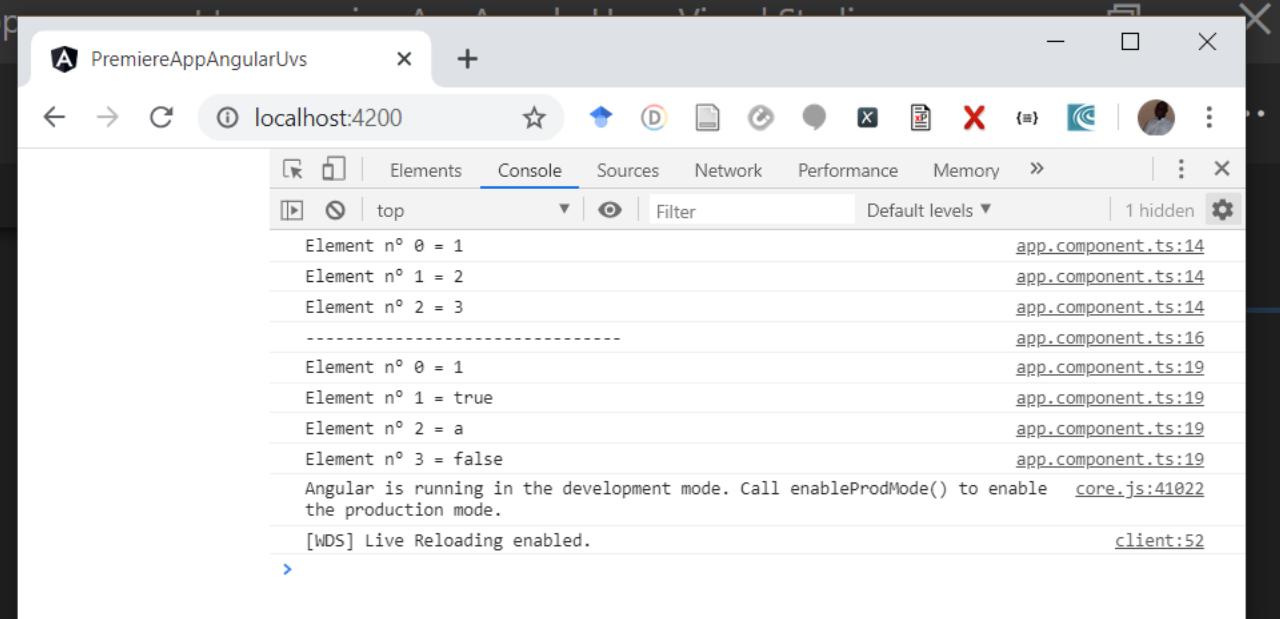
```
let list: number[] = [1, 2, 3];
let list: Array<number> = [1, 2, 3];
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active File:** main.ts - workspace-typescript - Visual Studio Code
- Code Editor:** The file main.ts contains the following code:

```
exemple1 > TS main.ts > ...
1 let tableau: Array<number> = [1, 2, 3];
2 console.log("### Affichage du tableau");
3 for (let index = 0; index < tableau.length; index++) {
4     const element = tableau[index];
5     console.log(element);
6 }
```
- Terminal:** Shows the command "Exemple de base".
- Output Panel:** Shows the execution results:

Output	Content	Line Number
Elements	main.js:2	
top	1	main.js:5
	2	main.js:5
	3	main.js:5
- Status Bar:** Compilation Failed!, Prof. Ousmane SALL, Univ. Thiès, SN, Compile Hero: On, Ln 8, Col 1, Spaces: 4, UTF-8, CRLF, TS 4.0.3, Prettier: ✓, 144.



The screenshot shows a Microsoft Edge browser window with the developer tools open, specifically the Console tab. The console output displays several log statements from the application code and Angular's core library.

```
Element n° 0 = 1  
Element n° 1 = 2  
Element n° 2 = 3  
-----  
Element n° 0 = 1  
Element n° 1 = true  
Element n° 2 = a  
Element n° 3 = false  
Angular is running in the development mode. Call enableProdMode() to enable core.js:41022  
the production mode.  
[WDS] Live Reloading enabled.
```

Below the browser window, the VS Code interface is visible, showing the file `app.component.ts` with the following code:

```
src > app > A app.component.ts < ...  
5     templateUrl: './app.component.html',  
6     styleUrls: ['./app.component.css']  
7 )  
8 export class AppComponent {  
| //title = uneFontion();  
10 }  
11  
12 let tab1: number[] = [1, 2, 3];  
13 for (let index = 0; index < tab1.length; index++) {  
| console.log("Element n° " + index + " = " + tab1[index]);  
15 }  
16 console.log("-----");  
17 let tab2: any[] = [1, true, 'a', false];  
18 for (let index = 0; index < tab2.length; index++) {  
| console.log("Element n° " + index + " = " + tab2[index]);  
20 }  
21
```

The status bar at the bottom of the screen shows the following information:

Prof. Ousmane SALL, Univ. Thiès, SN      Programmation Applications Mobiles      145  
master\*    ⊖ 0 △ 0      Ln 16, Col 49      Spaces: 2      UTF-8      LF      TypeScript      3.8.3      🔍      📰

# Méthodes associées aux Tableaux

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar:** Includes icons for File, Find, Replace, Open, Save, and Settings.
- Editor Area:** A file named "main.ts" is open. The code contains:

```
exemple1 > main.ts > ...
1  let tab=[1,2,3];
2  tab.
```

A code completion dropdown is displayed, listing methods for the Array type, such as push, map, forEach, length, join, concat, every, filter, indexOf, lastIndexOf, pop, reduce, reduceRight, reverse, shift, slice, some, sort, splice, toLocaleString, toString, and unshift. The "push" method is currently selected.
- Bottom Status Bar:** Shows the status "1: node" and other system information like date and time.
- Bottom Taskbar:** Displays a message "Compilation Failed!".



Welcome

index.html

ts

ma



exemple1 &gt; ts main.ts &gt; ...

```
1 // let tableau: number[] = [100, 20, 345]; ou bien
2 let tableau: Array<number> = [100, 20, 345];
3
4 console.log("### Ajout de -52 au tableau");
5 tableau.push(-52);
6 Affichage(tableau);
7
8 console.log("### Tri du tableau");
9 tableau.sort((n1,n2) => n1 - n2);
10 Affichage(tableau);
11
12 console.log("### Ajout de -52 au tableau");
13 tableau.pop();
14 Affichage(tableau);
15
16 function Affichage(tab: Array<number>) {
17     for (let index = 0; index < tab.length; index++) {
18         const element = tab[index];
19         console.log(element);
20     }
21 }
```

Exemple de base		
	Fichier   C:/Users/ousma/Desktop/workspace-type...	x
Elements	Console	Sources
top	14 messages	### Ajout de -52 au tableau main.js:3
	14 user mes...	100 main.js:15
		20 main.js:15
		345 main.js:15
		-52 main.js:15
	14 info	### Tri du tableau main.js:6
		-52 main.js:15
		20 main.js:15
		100 main.js:15
		345 main.js:15
	14 messages	### Ajout de -52 au tableau main.js:9
		-52 main.js:15
		20 main.js:15
		100 main.js:15

# Duplication de tableaux

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Left Sidebar:** Shows icons for Files, Folders, Recent, and Settings.
- Editor:** Displays the following TypeScript code:

```
exemple1 > main.ts > ...
58
59  let tableau: Array<number> = [100, 20, 345];
60  let copieTableau=[...tableau];// Duplication du tableau
61  console.log(copieTableau);
62
63
```
- Console:** An open browser developer tools window titled "Exemple de base". It shows the output of the console.log statement:

```
Fichier | C:/Users/ousma/Desktop/workspace-ty...
Elements Console Sources
top
(3) [100, 20, 345]
  0: 100
  1: 20
  2: 345
  length: 3
  __proto__: Array(0)
```
- Bottom Status Bar:** Shows the version 4.0.3, Prettier status (✓), and other icons.
- Page Footer:** Prof. Ousmane SALL, Univ. IIMES, SN and page number 148.

# Initiation à TypeScript: Les collections

## Les Maps

- L'objet **Map** est une simple paire **clé / valeur**. Les **clés** et les **valeurs** d'un **Map** peuvent être de n'importe quel type. Un **Map** est une collection dynamique.
- En d'autres termes, Maps peut croître et se réduire au moment de l'exécution
- Un **Map** peut être déclarés en utilisant un constructeur

### Ajouter, récupérer, supprimer des entrées du Map

Les opérations courantes disponibles sur un map sont:

**map.set()** - méthode pour ajouter des entrées dans Map.

**map.get()** - pour récupérer une entrée de Map.

**map.has()** - à l'existence d'une entrée dans le map.

**map.delete()** - supprime une entrée du map

**map.size** - la propriété «size» renverra la taille du map.



Welcome

index.html

ts

ma



exemple1 &gt; ts main.ts &gt; [?] monMap

1 const monMap = new Map&lt;number, string&gt;();

lib.es2015.c... C:\Users\ousma\AppData\Local\Programs\Microsoft VS Code\resources\app\extensions\node\_modules\typescript\lib - Definitions (5) X

```
19
20
21 interface Map<K, V> {
22     clear(): void;
23     delete(key: K): boolean;
24     forEach(callbackfn: (value: V, key: K, map: Map<K, V>) => void,
25             get(key: K): V | undefined;
26     has(key: K): boolean;
27     set(key: K, value: V): this;
28     readonly size: number;
29 }
```

```
30
31 interface MapConstructor {
32     new(): Map<any, any>;
```

```
2 monMap.set(1,"Abdoulaye WADE");
3 monMap.set(2,"Macky SALL");
4
```

```
✓ lib.es2015.collection.d.ts C:\Use...
  interface Map<K, V> {
    new<K, V>(entries?: readonly (readonly ...));
    declare var Map: MapConstructor;
  }
  > lib.es2015.iterable.d.ts C:\Users...
  > lib.es2015.symbol.wellknown.d....
```

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations like Open, Save, Find, and Settings. The main workspace has tabs for "Welcome", "index.html", and "main.ts". The "main.ts" tab is currently active, displaying the following TypeScript code:

```
exemple1 > ts main.ts > ...
1 const monMap = new Map<number, string>();
2 monMap.set(2, "Abdoulaye WADE");
3 monMap.set(1, "Macky SALL");
4 monMap.set(3, "Ousmane SONKO");
5
6 let val = monMap.get(2);
7 console.log("### La valeur à la position 2 est " + val);
8
9 let bool = monMap.has(5);
10 console.log("###Le clé 5 exsiste-t-elle ? " + bool);
11
12 console.log("### Le nombre d'éléments est " + monMap.size);
13
14 console.log("###Affichage des valeurs dans le tableau");
15 monMap.forEach(element) => {
16 |   console.log("\t" + element);
17 };
18
19 bool = monMap.delete(1);
20 console.log("### Suppression de l'élément à l'index 1: " + bool);
```

The status bar at the bottom indicates "Compilation Failed!".

On the right, a browser window is open, showing the output of the code execution. The title bar says "Exemple de base". The developer tools console tab is selected, showing the following log entries:

Message	File	Line
### La valeur à la position 2 est Abdoulaye WADE	main.js	6
###Le clé 5 exsiste-t-elle ? false	main.js	8
### Le nombre d'éléments est 3	main.js	9
###Affichage des valeurs dans le tableau	main.js	10
Abdoulaye WADE	main.js	12
Macky SALL	main.js	12
Ousmane SONKO	main.js	12
### Suppression de l'élément à l'index 1: true	main.js	15

The status bar at the bottom also shows "Compilation Failed!".

# Initiation à TypeScript: Les collections

## les Map

The screenshot shows the Visual Studio Code interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a tab for main.ts - workspace-typescript - Visual Studio Code. The left sidebar has icons for Welcome, index.html, and main.ts. The main editor area displays the following TypeScript code:

```
exemple1 > main.ts > ...
1 let monMap = { 1: "Abdoulaye WADE", 2: "Macky SALL" };
2 monMap[3] = "Ousmane SONKO";
3 let val = monMap[3];
4 console.log("### La valeur à la position 3 est " + val);
```

An orange status bar at the bottom indicates "Compilation Failed!". A DevTools window titled "Exemple de base" is open, showing the "Console" tab with the output: "### La valeur à la position 3 est Ousmane SONKO". The status bar also shows CRLF, TS 4.0.3, and Prettier checked.

# Convertir un Map en Json

The screenshot shows a Visual Studio Code interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a title main.ts - workspace-typescript - Visual Studio Code. The left sidebar has icons for File Explorer, Search, and Settings. The main editor area has tabs for index.html and main.ts, with main.ts currently selected. The code in main.ts is:

```
const monMap = new Map<number, string>();
monMap.set(1, "Abdoulaye WADE");
monMap.set(2, "Macky SALL");
monMap.set(3, "Ousmane SONKO");

let jsonObject = {};
monMap.forEach((value, key) => {
  jsonObject[key] = value;
});
console.log(JSON.stringify(jsonObject));
```

The status bar at the bottom shows 0 errors and 0 warnings, and a message Compilation Failed!. A tooltip box with a dashed border contains the following text:

Le map est créé avec une clé et des valeurs de type chaînes de caractères `map.forEach` est utilisée pour itérer sur le map. Le format `JSON` est retourné à l'aide de la méthode `JSON.stringify()`

In the background, a browser developer tools window titled "Exemple de base" is open, showing the console tab with the output: {"1": "Abdoulaye WADE", "2": "Macky SALL", "3": "Ousmane SONKO"}.

# Les objets

- Un **objet** est une instance qui contient un ensemble de paires **clé / valeur**. Les **valeurs** peuvent être des **valeurs scalaires** ou des **fonctions** ou même un **tableau d'autres objets**. La syntaxe est donnée ci-dessous

```
let nom_objet = {  
    key1: "valeur1", // valeur scalaire  
    key2: "valeur1",  
    key3: function() {  
        //fonctions  
    },  
    key4: ["contenu1", "contenu2"] //collection  
};
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.ts - workspace-typescript - Visual Studio Code.
- Sidebar Icons:** Document, Search, Find, Open, Recent, Settings, Help.
- Editor:** Two tabs are open: index.html and main.ts. The main.ts tab contains the following code:

```
exemple1 > main.ts > ...
1 let personne = {
2   prenom: "Macky",
3   nom: "SALL",
4   direBonjour: function () {}, //Type template
5 };
6
7 personne.direBonjour = function () {
8   console.log("Bonjour " + personne.prenom);
9 };
10 personne.direBonjour();
11
12 let invoquerPersonne = function (obj) {
13   console.log("Prénom :" + obj.prenom + " Nom :" + obj.nom);
14 };
15 invoquerPersonne(personne);
16
17 console.log("### Itérer sur les propriétés d'un objet ");
18 for (const key in personne) {
19   if (Object.prototype.hasOwnProperty.call(personne, key)) {
20     console.log(key + " : " + personne[key]);
21   }
22 }
23 console.log("### Conversion Json de l'objet ");
24 let jsonstr = JSON.stringify(personne);
25 console.log(jsonstr);
26
```

- Terminal:** Shows the message "Compilation Failed!".
- Output Panel:** A browser window titled "Exemple de base" is displayed, showing the execution results in the console tab:

Output	File
Bonjour Macky	main.js:7
Prénom :Macky Nom :SALL	main.js:11
### Itérer sur les propriétés d'un objet	main.js:14
prenom : Macky	main.js:17
nom : SALL	main.js:17
direBonjour : function () { console.log("Bonjour " + personne.prenom); }	main.js:17
### Conversion Json de l'objet	main.js:20
{"prenom":"Macky","nom":"SALL"}	main.js:22

- Status Bar:** Shows "Compile Hero: On", line 20, column 27, spaces: 4, encoding: UTF-8, CRLF, TS 4.0.3, Prettier: ✓.

# TypeScript

## Programmation Orientée Objet

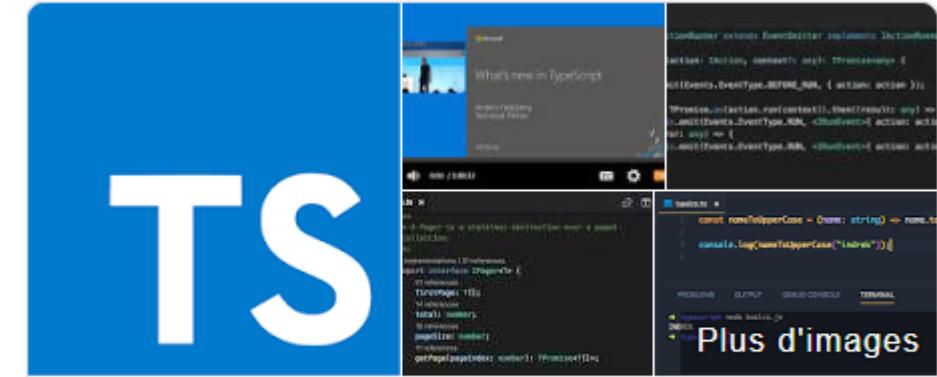
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- **Programmation Orientée Objet**
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- Les décorateurs



## TypeScript

Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)

Extension de fichier : `ts`

Date de première version : [9 février 2012](#)

Dernière version : [3.2.2 \(29 novembre 2018\)](#)

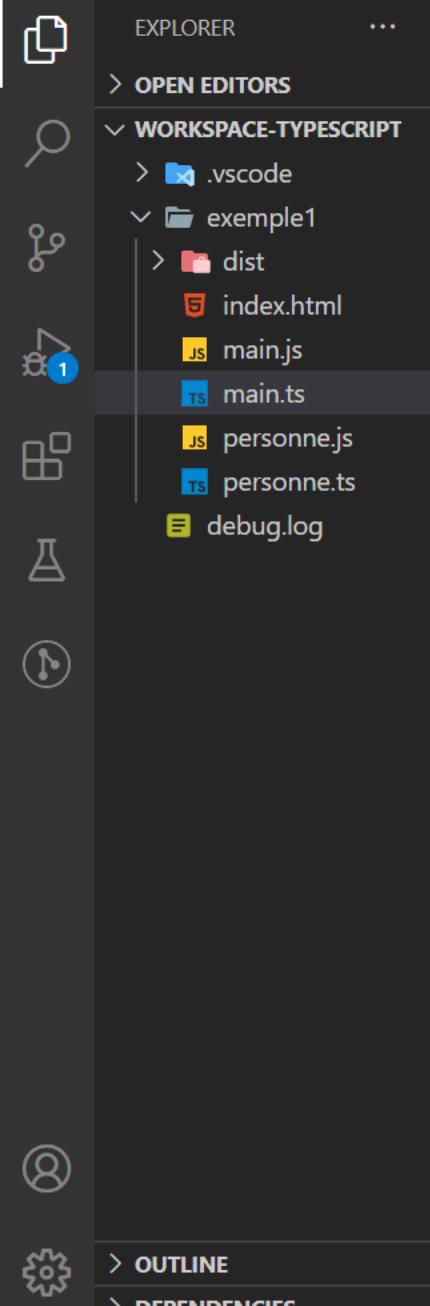
Typage : dynamique, faible, fort optionnel, statique optionnel

Paradigme : Multi-paradigme

# Initiation à TypeScript: Programmation orientée objet - Notions de classe

- TypeScript est un langage orienté objet. Il prend en charge les fonctionnalités de programmation orientée objet comme les classes, les interfaces, etc. Une classe en termes de POO est un modèle pour la création d'objets. Une classe encapsule les données de l'objet. TypeScript fournit un support intégré pour ce concept appelé classe.
- Utilisez le mot-clé **class** pour déclarer une classe dans TypeScript. Une définition de classe commence par le mot-clé **class** suivi du nom de la classe; et le corps de la classe entouré par une paire d'accolades. La syntaxe est donnée ci-dessous.

```
class class_name {  
    <attributs>  
    <getters/setters>  
    <constructeurs>  
    <fonctions>  
}
```



```
exemple1 > [ts] personne.ts > {} Personne > Personne
1 module Personne{
2     export class Personne {
3         private _nom: string;
4         private _age: number;
5
6         constructor(nom: string, age: number) {
7             this._nom = nom;
8             this._age = age;
9         }
10
11         public get nom(): string {
12             return this._nom;
13         }
14
15         public set nom(v: string) {
16             this._nom = v;
17         }
18
19         public get age(): number {
20             return this._age;
21         }
22
23         public set age(v: number) {
24             this._age = v;
25         }
26     }
27 }
```

```
exemple1 > [ts] main.ts > [e] pers
1 /// <reference path="personne.ts" />
2 let pers = new Personne.Personne("Sall Macky", 59);
3 console.log(pers.nom);
4 pers.nom="Marieme FAYE SALL";
5 console.log(pers);
```

[index.html]

exemple1 > [index.html] > [e] html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.
7     <title>Exemple de base</title>
8     <script src="personne.js"></script>
9     <script src="main.js"></script>
10    </head>
11
12    <body>
13
14    </body>
15
16    </html>
```

Utiliser tsc --target ES2020 .\personne.ts à cause des getters et setters

Le **constructeur** est une méthode spéciale qui est appelée lors de la création d'un objet. Dans TypeScript, le constructeur est toujours définie avec le nom "**constructor**".

# Initiation à TypeScript: **Les classes** - Visibilités des attributs et Accesseurs

- **public**: Par défaut, tous les membres d'une classe dans TypeScript sont publics. Tous les membres du public sont accessibles n'importe où sans aucune restriction.
- **private**: Le modificateur d'accès privé garantit que les membres de la classe sont visibles uniquement pour cette classe et ne sont pas accessibles en dehors de la classe conteneur.
- **protected**: Le modificateur d'accès protégé est similaire au modificateur d'accès privé, sauf que les membres protégés sont accessibles à l'aide de leurs classes dérivées.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer Bar (Left):** Shows the project structure under "WORKSPACE-TYPESCRIPT". The "exemple1" folder contains ".vscode", "dist", "index.html", "main.js", "main.ts" (1 unsaved), "personne.js", "personne.ts", and "debug.log".
- Editor (Top):** Two tabs are open: "main.ts" and "personne.ts".
- Editor (Bottom):** The "main.ts" editor shows the following code:

```
1 module Personne{  
2     export class Personne {  
3         private _nom: string;  
4         private _age: number;  
5  
5         constructor(nom: string, age: number) {  
6             this._nom = nom;  
7             this._age = age;  
8         }  
9  
11         public get nom(): string {  
12             return this._nom;  
13         }  
14  
15         public set nom(v: string) {  
16             this._nom = v;  
17         }  
18  
19         public get age(): number {  
20             return this._age;  
21         }  
22  
23         public set age(v: number) {  
24             this._age = v;  
25         }  
26     }  
27 }
```

- Editor (Right):** The "main.ts" editor has a tooltip for the line "Property '\_nom' is private and only accessible within class 'Personne'. ts(2341)".
- Editor (Bottom):** The "index.html" editor shows the following code:

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4     <head>  
5         <meta charset="UTF-8">  
6         <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7         <title>Exemple de base</title>  
8         <script src="personne.js"></script>  
9         <script src="main.js"></script>  
10    </head>  
11  
12    <body>  
13  
14    </body>  
15  
16    </html>
```

File Edit Selection View Go Run Terminal Help personne.ts - workspace-typescript - Visual Studio Code

main.ts      personne.ts X

exemple1 > personne.ts > {} Personne > Personne > constructor

```
1 module Personne {  
2     export class Personne {  
3         constructor(private _nom?: string, private _age?: number) {  
4             //  
5         }  
6  
7         public get nom(): string {  
8             return this._nom;  
9         }  
10        public set nom(v: string) {  
11            this._nom = v;  
12        }  
13  
14        public get age(): number {  
15            return this._age;  
16        }  
17  
18        public set age(v: number) {  
19            this._age = v;  
20        }  
21    }  
22}
```

Possibilité de déclarer les attributs dans le constructeur

index.html      main.ts X      personne.ts X

exemple1 > main.ts > ...

```
1 //<reference path="personne.ts" />  
2 let pers = new Personne.Personne("Sall Macky", 59);  
3 console.log(pers.nom);  
4 pers.nom="Marieme FAYE SALL";  
5 console.log(pers);  
6
```

index.html X

exemple1 > index.html > html

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5     <meta charset="UTF-8">  
6     <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">  
7     <title>Exemple de base</title>  
8     <script src="personne.js"></script>  
9     <script src="main.js"></script>  
10    </head>  
11  
12    <body>  
13  
14    </body>  
15  
16    </html>
```

Exemple de base

Sall Macky

Personne {\_nom: "Marieme FAYE SALL", \_age: 59}

\_age: 59  
\_nom: "Marieme FAYE SALL"  
age: (...)  
nom: (...)

main.js:3  
main.js:5

Compile Hero: On    Ln 4, Col 9    Spaces: 4    UTF-8    CRLF    TS    4.0.3    Prettier: ✓

0 ▲ 0    Compilation Failed!

# Initiation à TypeScript: Les classes - Héritage avec le mot clé **extends**

```
class classeFille extends classeMere
```

File Edit Selection View Go Run Terminal Help • personne.ts - workspace-typescript - Visual Studio Code

main.ts ● personne.ts ● etudiant.ts

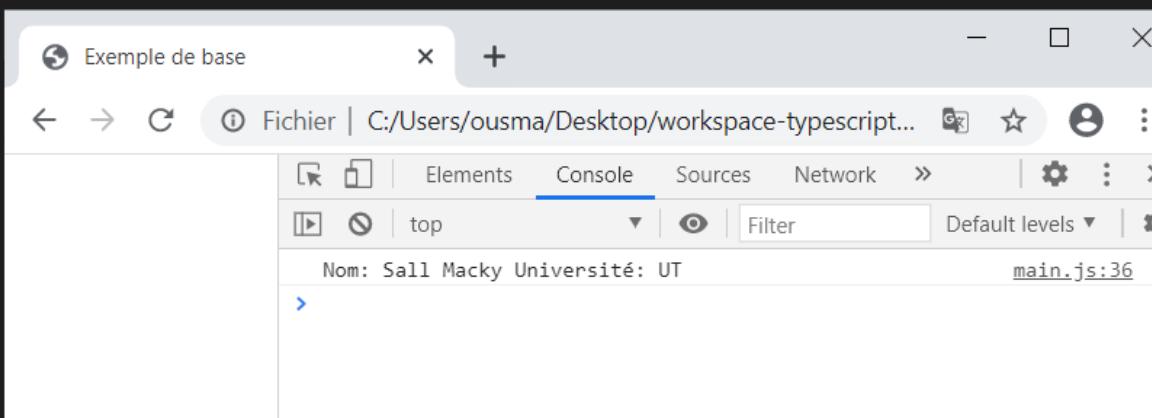
exemple1 > main.ts > Etudiant > (set) universite

```
1 //<reference path="personne.ts" />
2 class Etudiant extends Personne.Personne {
3     constructor(_nomEtu: string, _ageEtu: number, private _universite?: string) {
4         super(_nomEtu, _ageEtu);
5     }
6
7     public get universite(): string {
8         return this._universite;
9     }
10
11    public set universite(v: string) {
12        this._universite = v;
13    }
14
15    /**
16     * afficherEtudiant
17     */
18    public afficherEtudiant() {
19        console.log("Nom: " + this.nom + " Université: " + this._universite);
20    }
21}
22let etudiant = new Etudiant("Sall Macky", 59, "UT");
23etudiant.afficherEtudiant();
24
```

lex.html main.ts ● personne.ts ●

exemple1 > personne.ts > {} Personne > Personne

```
1 module Personne {
2     export class Personne {
3         constructor(private _nom?: string, private _age?: number) {}
4
5         public get nom(): string {
6             return this._nom;
7         }
8
9         public set nom(v: string) {
10            this._nom = v;
11        }
12
13        public get age(): number {
14            return this._age;
15        }
16
17        public set age(v: number) {
18            this._age = v;
19        }
20    }
21}
22
```



index.html

exemple1 > index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4     <head>
5         <meta charset="UTF-8">
6         <meta name="viewport" content="width=device-width, initial-scale=1.0">
7         <title>Exemple de base</title>
8         <script src="personne.js"></script>
9         <script src="main.js"></script>
10    </head>
11
12    <body>
13
14    </body>
```

# Initiation à TypeScript: Interface

- Une interface TypeScript permet de définir la signature (*ou le contrat*) d'une classe où même une fonction.
- Les interfaces TypeScript ne sont utilisées que lors de la transpilation. Elles disparaissent totalement en runtime étant donné qu'elles ne contiennent pas de code.
- Le compilateur TypeScript ne convertit pas l'interface en JavaScript. Il utilise l'interface pour la vérification de type. Ceci est également connu comme "typage de canard" ou "sous-typage structurel".
- Une interface est définie avec le mot clé `interface` et peut inclure des propriétés et des déclarations de méthodes à l'aide d'une fonction ou d'une fonction de flèche.



EXPLORER

...

main.ts

●

personne.ts

●

main.ts

●

personne.ts

●

main.ts

●

...

...



&gt; OPEN EDITORS 1 UNSAVED

## WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1

&gt; dist

etudiant.js

etudiant.ts

index.html

main.js

main.ts

personne.js

personne.ts

debug.log

&gt; OUTLINE

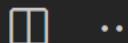
&gt; DEPENDENCIES

exemple1 &gt; personne.ts &gt; {} Personne &gt; Personne &gt; afficherPers

```
1 module Personne {
2     export class Personne {
3         constructor(private _nom?: string, private _age?: number) {}
4
5         public get nom(): string {
6             return this._nom;
7         }
8
9         public set nom(v: string) {
10            this._nom = v;
11        }
12
13        public get age(): number {
14            return this._age;
15        }
16
17        public set age(v: number) {
18            this._age = v;
19        }
20        afficherPers() {
21            console.log(`Nom: ${this._nom} - Age: ${this._age}`);
22        }
23    }
24}
```

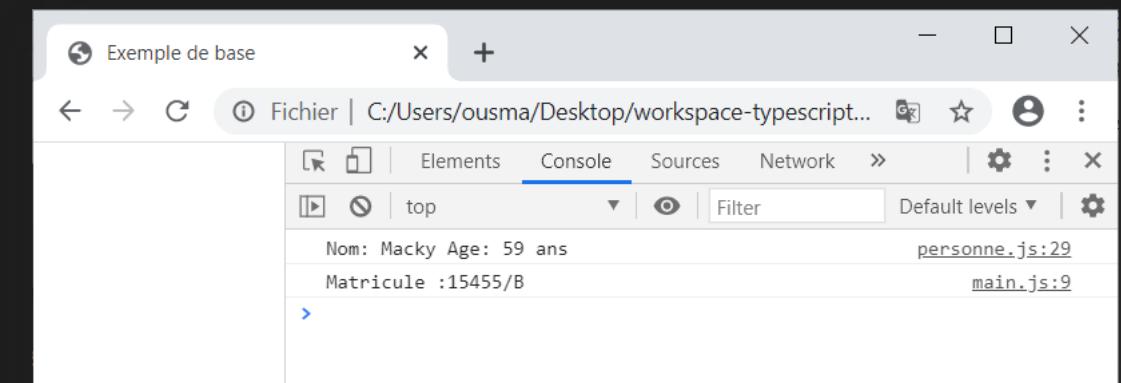


main.ts X



exemple1 &gt; main.ts &gt; IEmploye

```
1  /// <reference path="personne.ts" />
2
3  interface IEmploye {
4      matricule: string;
5      afficherEmploye(): void;
6  }
7
8  class EmployeUT implements IEmploye {
9      matricule: string;
10     constructor(private _matricule?: string, private _pers?: Personne.Personne) {}
11
12     afficherEmploye(): void {
13         pers.afficherPers();
14         console.log("Matricule :" + this._matricule);
15     }
16 }
17
18 let pers = new Personne.Personne("Macky", 59);
19 let emp = new EmployeUT("15455/B", pers);
20 emp.afficherEmploye();
```



# TypeScript

## Programmation Asynchrone

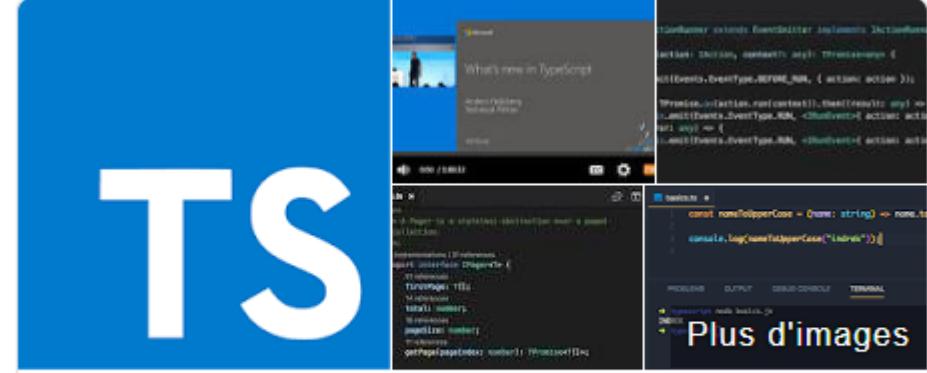
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- **Programmation Asynchrone**
- Les Espaces de nom et les Modules
- Les décorateurs



**TypeScript**  
Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript.

[Wikipédia](#)

Conçu Par : [Microsoft](#)  
Extension de fichier : `ts`  
Date de première version : [9 février 2012](#)  
Dernière version : [3.2.2 \(29 novembre 2018\)](#)  
Typage : dynamique, faible, fort optionnel, statique optionnel  
Paradigme : Multi-paradigme

[Plus d'images](#)

# Programmation asynchrone: **promise**, **async**, **await**, **.then**

- Pourquoi le code asynchrone est-il important?
  - Les opérations asynchrones permettent à votre programme de terminer son travail en attendant qu'une autre opération se termine.
- Voici quelques opérations asynchrones courantes:
- Récupération de données sur un réseau.
  - Ecrire dans une base de données.
  - Lire des données à partir d'un fichier.
- Pour effectuer des opérations asynchrones dans TypeScript, vous pouvez utiliser la classe **Promise** et les mots-clés **async** et **await**.

En JavaScript, une promesse fait référence à l'attente que quelque chose se produise à un moment donné, et votre application s'appuie sur le résultat de cet événement futur pour effectuer certaines autres tâches.

# Programmation asynchrone

The screenshot shows a code editor with a dark theme. On the left, there are several icons for file operations, search, and navigation. The main area displays a file named "main.ts". The code is as follows:

```
1  console.log("Début " + new Date());
2  const start = (...callback) => {
3      setTimeout(() => {
4          callback("Hello " + new Date());
5          setTimeout(() => {
6              callback("Et Bienvenue " + new Date());
7              setTimeout(() => {
8                  callback("Async Await avec TypeScript " + new Date());
9                  }, 10000);
10                 }, 20000);
11                 }, 30000);
12             };
13             start(...text) => console.log(text));
14
15 console.log("### Fin " + new Date());
```

At the bottom, there are status indicators: "0 0 ⚡ 0" and "Compilation Failed!".

Une façon de programmer de manière asynchrone consiste à utiliser les callbacks. Nous passons à une fonction asynchrone une fonction qu'elle appellera lorsque la tâche sera terminée.

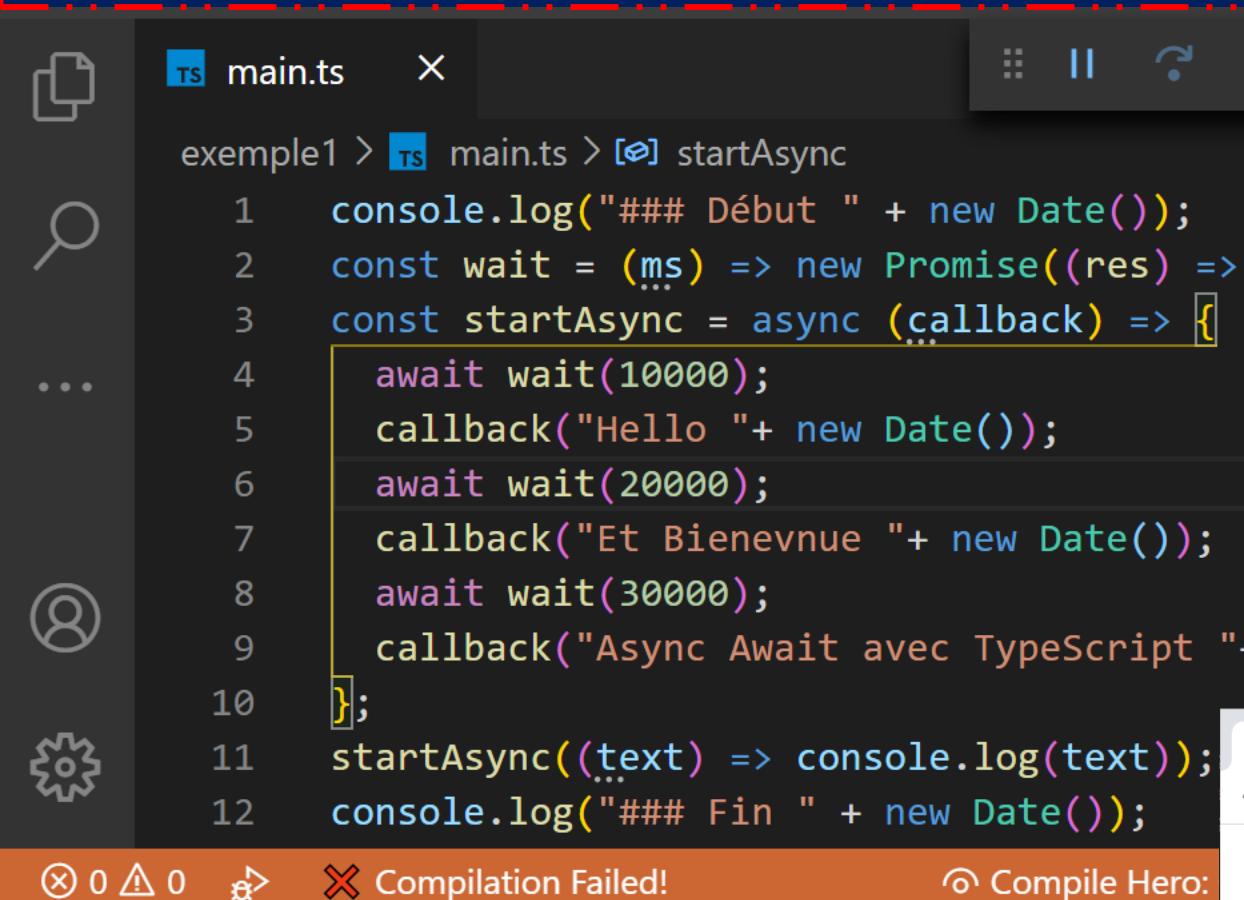
Dans l'exemple, une fonction normale est définie et un argument est passé qui est une fonction et sera appelé une fois la tâche à l'intérieur du bloc fonction terminée. Nous allons d'abord écrire une fonction appelée `start` qui prend un callback et l'appelle à l'aide de la fonction **setTimeout**.

The screenshot shows a browser's developer tools with the "Console" tab selected. The title bar says "Exemple de base". The console output is as follows:

```
Début Thu Oct 15 2020 23:50:33 GMT+0000 (heure moyenne de Greenwich)
### Fin Thu Oct 15 2020 23:50:33 GMT+0000 (heure moyenne de Greenwich)
Hello Thu Oct 15 2020 23:51:03 GMT+0000 (heure moyenne de Greenwich)
Et Bienvenue Thu Oct 15 2020 23:51:23 GMT+0000 (heure moyenne de Greenwich)
Async Await avec TypeScript Thu Oct 15 2020 23:51:33 GMT+0000 (heure moyenne de Greenwich)
```

On the right side of the console, there are file paths: "main.js:14", "main.js:13", and "main.js:13".

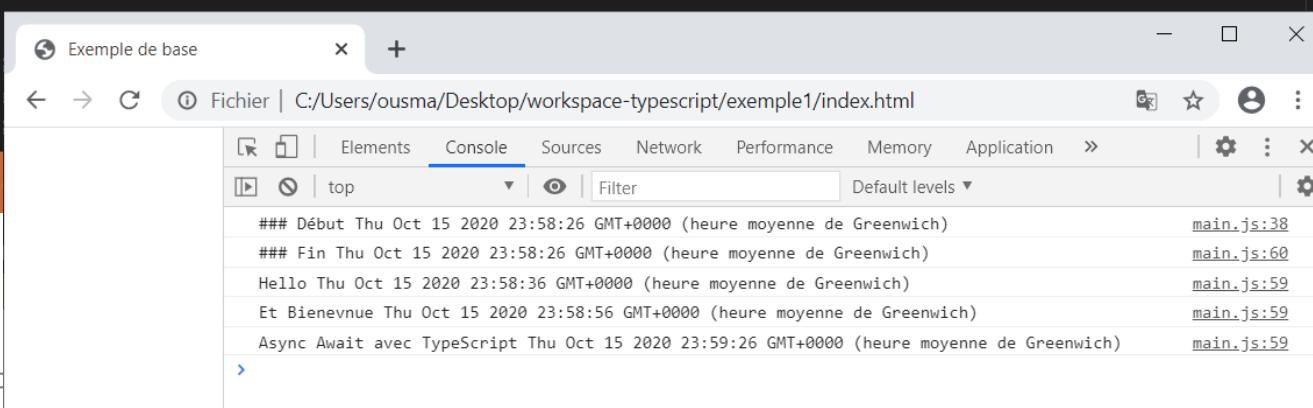
Cette fonction «**wait**» prend un certain nombre de millisecondes et renvoie une promesse qui est résolue à l'aide de **setTimeout** après le nombre de millisecondes donné.



```
main.ts
example1 > main.ts > startAsync
1  console.log("### Début " + new Date());
2  const wait = (ms) => new Promise((res) => setTimeout(res, ms));
3  const startAsync = async (callback) => {
4      await wait(1000);
5      callback("Hello "+ new Date());
6      await wait(2000);
7      callback("Et Bienvenue "+ new Date());
8      await wait(3000);
9      callback("Async Await avec TypeScript "+ new Date());
10 };
11 startAsync((text) => console.log(text));
12 console.log("### Fin " + new Date());
```

Compilation Failed!

Créez maintenant une fonction asynchrone appelée **startAsync**. Cette fonction est en fait assez similaire à la fonction de démarrage que nous avions écrite auparavant. À l'intérieur de cette fonction, nous utiliserons l'attente pour suspendre l'exécution du code jusqu'à ce que la promesse soit résolue et appeler le rappel qui passe dans le temps.



# Qu'est-ce qu'une **Promise**?

- Une **promise** («p» minuscule) est une instance de la classe **Promise** («P» majuscule). Une **promise** représente est un objet qui représente l'état d'une opération asynchrone. Une opération asynchrone peut être dans l'un des états suivants : Opération en cours (non terminée) ; Opération terminée avec succès (promesse résolue) ; Opération terminée ou plus exactement stoppée après un échec (promesse rejetée).
- Si la promesse est tenue, la fonction **resolve()** sera appelée tandis que si la promesse est rompue la fonction **reject()** va être appelée. Ces deux fonctions sont des fonctions prédéfinies en JavaScript et nous n'avons donc pas besoin de les déclarer.

# Qu'est-ce qu'une **Promise**?

- Nous allons définir une fonction dont le rôle est d'effectuer une opération asynchrone et cette fonction va, lors de son exécution, créer et renvoyer un objet Promesse
- En pratique, la majorité des opérations asynchrones qu'on va vouloir réaliser en JavaScript vont déjà être pré-codées et fournies par des API. Ainsi, nous allons rarement créer nos propres promesses mais plutôt utiliser les promesses renvoyées par les fonctions de ces API.
- Lorsque nos fonctions asynchrones s'exécutent, elles renvoient une promesse. Cette promesse va partager les informations liées à l'opération qui vient de s'exécuter et on va pouvoir l'utiliser pour définir quoi faire en fonction du résultat qu'elle contient (en cas de succès de l'opération ou en cas d'échec).

<https://www.pierre-giraud.com/javascript-apprendre-coder-cours/promesse-promise/>

# Qu'est-ce qu'une **Promise**?

- Pour obtenir et exploiter le résultat d'une promesse, on va généralement utiliser la méthode **then()** du constructeur Promise.
- Dans le cas où on est intéressé uniquement par le cas où une promesse est rompue, on va pouvoir utiliser la méthode **catch()** qui va prendre une unique fonction de rappel en argument qui va être appelée si la promesse est rompue.

# Qu'est-ce qu'une **Promise**?

## Le chainage des promesses

- « **Chainer** » des méthodes signifie les exécuter les unes à la suite des autres. On va pouvoir utiliser cette technique pour exécuter plusieurs opérations asynchrones à la suite et dans un ordre bien précis.
- Cela est possible pour une raison : la méthode **then()** retourne automatiquement une nouvelle promesse. On va donc pouvoir utiliser une autre méthode **then()** sur le résultat renvoyé par la première méthode **then()** et ainsi de suite.

File Edit Selection View Go Run Terminal Help main.ts - work

main.ts

```
exemple1 > main.ts > promise > then() callback > setTimeout() c
1 console.log("### Début " + new Date());
2 let promise = new Promise((resolve, reject) => {
3     setTimeout(() => {
4         console.log("### Hello " + new Date());
5         resolve();
6     }, 1500);
7 })
8 .then(() => {
9     setTimeout(() => {
10        console.log("### Et Bienvenue " + new Date());
11    }, 3000);
12 })
13 .then(() => {
14     setTimeout(() => {
15         console.log("### Et après " + new Date());
16     }, 6000);
17 })
18
19 console.log("### Fin " + new Date());
20
21 // const wait = (ms) => new Promise((res) => setTimeout(res, ms))
```

Exemple de base

Fichier | C:/Users/ousma/Desktop/workspace... Elements Console Sources

	Console	Default levels
top	### Début Fri Oct 16 2020 00:21:18 GMT+0000 ### Fin Fri Oct 16 2020 00:21:18 GMT+0000 ### Hello Fri Oct 16 2020 00:21:19 GMT+0000 ### Et Bienvenue Fri Oct 16 2020 00:21:22 GMT+0000 ### Et après Fri Oct 16 2020 00:21:25 GMT+0000	main.js:1 main.js:20 main.js:4 main.js:10 main.js:15

Dans cette promesse, le constructeur de promesse est utilisé pour prendre dans string comme type générique pour la valeur de résolution de la promesse. Le constructeur de promesse prend deux arguments:

- «**resolve**» - Il s'agit d'une fonction utilisée pour résoudre la promesse.
- "**reject**" - Il s'agit d'une fonction utilisée pour rejeter la promesse.

Ainsi, notre promesse peut être résolue ou rejetée. La partie de résolution est prise en charge par **.then**, et la partie de rejet est prise en charge par **.catch**.

# TypeScript

## Les Espaces de nom et les Modules

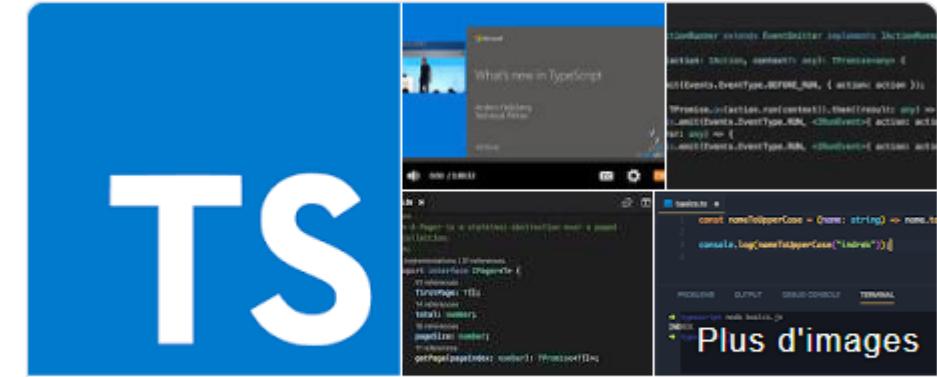
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- **Les Espaces de nom et les Modules**
- Les décorateurs



## TypeScript

Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)

Extension de fichier : `ts`

Date de première version : [9 février 2012](#)

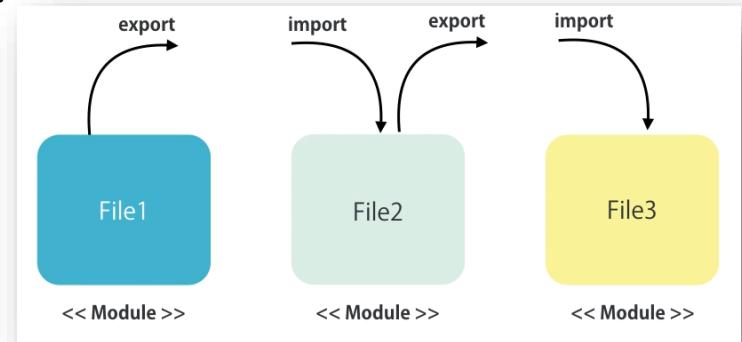
Dernière version : [3.2.2 \(29 novembre 2018\)](#)

Typage : dynamique, faible, fort optionnel, statique optionnel

Paradigme : Multi-paradigme

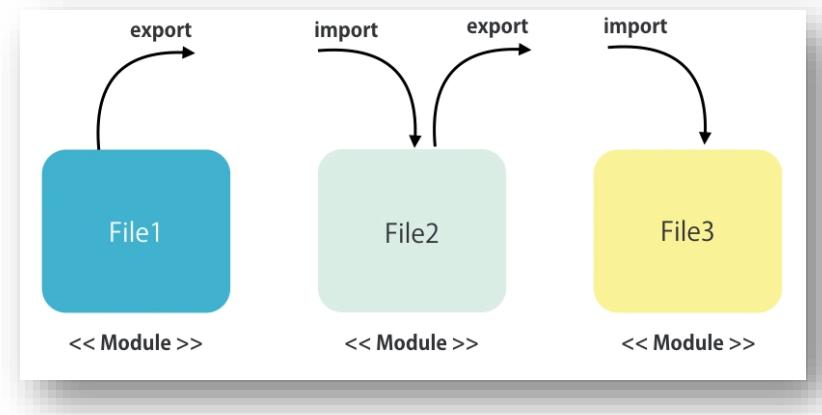
# Les espaces de nom et les modules

- Façon de découper le code pour le réutiliser plus facilement
- Dans TypeScript, nous divisons notre programme en plusieurs fichiers dans chaque fichier, nous exportons un ou plusieurs types, ces types peuvent être des classes, des fonctions, des variables simples ou des objets, et chaque fois que nous devons utiliser ces types, nous devons d'abord les importer.
- Lorsque nous avons une instruction d'importation ou d'exportation au-dessus d'un fichier, ce fichier est un module du point de vue de TypeScript.



# Les espaces de nom et les modules

- Un module peut ainsi être créé à l'aide du mot clé export et un module peut être utilisé dans un autre module à l'aide du mot clé import.
- Les espaces de noms sont équivalents aux modules internes.



# Les espaces de nom et les modules

- Les applications angular sont modulaires et Angular possède son propre système de modularité appelé **NgModules**.
- Chaque application Angular possède au moins une classe **NgModule**, le module racine, nommé conventionnellement AppModule.
- Les modules sont un excellent moyen d'organiser une application et de l'étendre avec des fonctionnalités à partir de bibliothèques externes.
- De nombreuses bibliothèques Angular sont des modules (tels que **FormsModule**, **HttpModule** et **RouterModule**). De nombreuses bibliothèques tierces sont disponibles en tant que **NgModules** (telles que Material Design, Angular, AngularFire).

# Un exemple de namespace



EXPLORER

...

## &gt; OPEN EDITORS

## WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1



&gt; dist

&gt; node\_modules

JS etudiant.js

TS etudiant.ts

ES index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json 1

JS personne.js

TS personne.ts 2

{...} tsconfig.json

debug.log

## &gt; OUTLINE

## &gt; DEPENDENCIES

TS main.ts X

TS etudiant.ts

TS personne.ts

{...} tsconfig.json

JS package.json



exemple1 &gt; TS main.ts &gt; {} espaceCalcul

```
1  namespace espaceCalcul{  
2      export function somme(x:number, y:number){  
3          return x+y;  
4      }  
5  }  
6  
7  console.log("La somme est de "+espaceCalcul.somme(10,41));  
8
```

PROBLEMS

3

OUTPUT

TERMINAL

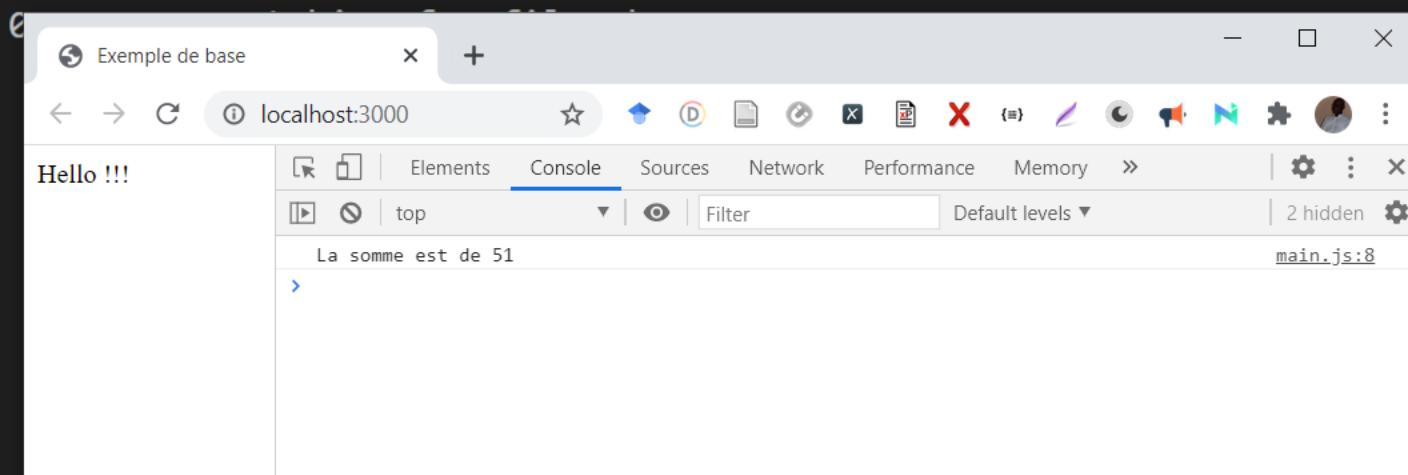
...

2: node



[02:32:18] File change detected. Starting incremental compilation...

[02:32:19] Found 0 errors, 0 warnings.





EXPLORER

OPEN EDITORS 1 UNSAVED

WORKSPACE-TYPESCRIPT

- .vscode
- exemple1
- dist
- node\_modules
- etudiant.js
- exempleEspaceDeNomExterne.ts

index.html

main.js

main.ts

package-lock.json

package.json

personne.js

personne.ts

tsconfig.json

debug.log

OUTLINE

DEPENDENCIES

main.ts    ●    exempleEspaceDeNomExterne.ts X

```
exemple1 > exempleEspaceDeNomExterne.ts > {} espaceCalcul
1  namespace espaceCalcul{
2      export function somme(x:number, y:number){
3          return x+y;
4      }
5 }
```

PROBLEMS TERMINAL ... 2: node

[02:32:18] File change detected. Starting incremental compilation...

[02:32:19] Found 0 errors. Watching for file changes.





EXPLORER

...

&gt; OPEN EDITORS

&gt; WORKSPACE-TYPESCRIPT



&gt; exemple1



&gt; node\_modules

JS etudiant.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json

JS personne.js

TS personne.ts

{...} tsconfig.json

Text debug.log

&gt; OUTLINE

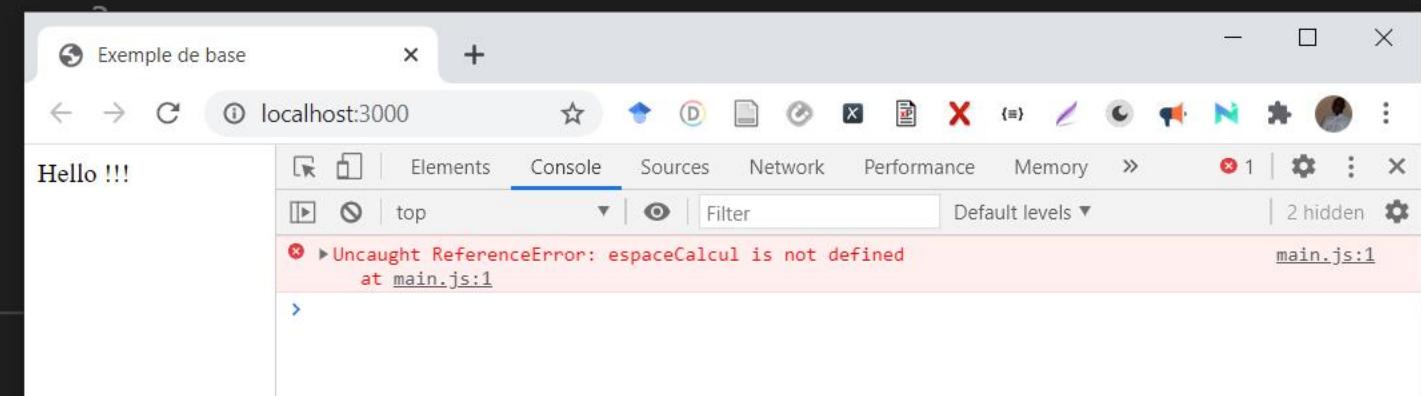
&gt; DEPENDENCIES

TS main.ts X TS exempleEspaceDeNomExterne.ts

exemple1 &gt; TS main.ts

1

2 console.log("La somme est de "+espaceCalcul.summe(10,41));



[02:35:09] File change detected. Starting incremental compilation...

main.ts:2:32 - error TS2304: Cannot find name 'espaceCalcul'.

2 console.log("La somme est de "+espaceCalcul.summe(10,41));

~~~~~

[02:35:10] Found 1 error. Watching for file changes.





EXPLORER

...

main.ts

index.html

exempleEspaceDeNomExterne.ts X



...

&gt; OPEN EDITORS

WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1

&gt; dist

&gt; node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

JSON package-lock.json

JSON package.json

JS personne.js

TS personne.ts

{...} tsconfig.json

&gt; OUTLINE

&gt; DEPENDENCIES

exemple1 &gt; TS exempleEspaceDeNomExterne.ts &gt; {} espaceCalcul

```
1 namespace espaceCalcul{  
2     export function somme(x:number, y:number){  
3         return x+y;  
4     }  
5 }
```

PROBLEMS

TERMINAL

...

2: node



~~~~~

[02:37:33] Found 1 error. Watching for file changes.

^CTerminer le programme de commandes (O/N) ? o

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1&gt; tsc .\exempleEspaceDeNomExterne.ts

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1&gt;

0 0



Compilation Failed!

Compile Hero: On

Ln 5, Col 2

Spaces: 4

UTF-8

CRLF

TS 4.0.3

Prettier: ✓





EXPLORER

...

main.ts

index.html X

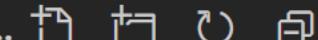
exempleEspaceDeNomExterne.ts



...

&gt; OPEN EDITORS

&lt; WORKSPACE-TYPESCR...



&gt; .vscode

&gt; exemple1

&gt; dist

&gt; node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

S index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json

JS personne.js

TS personne.ts

{...} tsconfig.json

&gt; OUTLINE

&gt; DEPENDENCIES

exemple1 &gt; S index.html

```
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Exemple de base</title>
8     <script src="exempleEspaceDeNomExterne.js"></script>
9     <script src="personne.js"></script>
10    <script src="main.js"></script>
11 </head>
12
```

PROBLEMS

TERMINAL

...

1: node



20.10.16 0

20.10.16 0

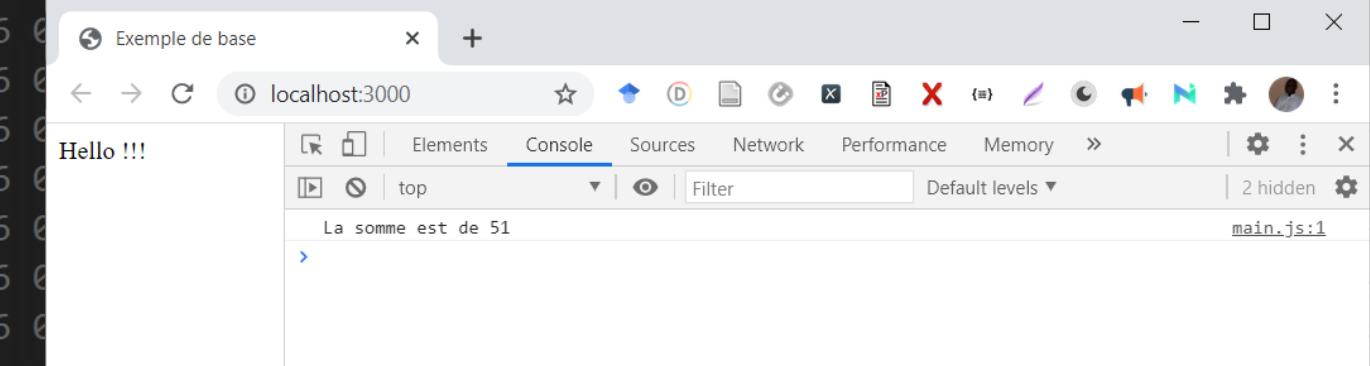
20.10.16 0

20.10.16 0

20.10.16 0

20.10.16 0

20.10.16 0





EXPLORER

...

main.ts

exempleEspaceDeNomExterne.ts

index.html



...

&gt; OPEN EDITORS

&lt; WORKSPACE-TYPESCRIPT

- > dist
- > node\_modules
- JS etudiant.js
- JS exempleEspaceDeNomExterne.js
- TS exempleEspaceDeNomExterne.ts
- ES6 index.html
- JS main.js
- TS main.ts
- JSON package-lock.json
- JSON package.json
- JS personne.js
- TS personne.ts
- {} tsconfig.json
- Text debug.log

exemple1 &gt; ES6 index.html

```
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Exemple de base</title>
8     <!-- <script src="exempl...></script> -->
9     <script src="personne.js"></script>
10    <script src="main.js"></script>
11 </head>
12
```

PROBLEMS

TERMINAL

...

2: node

```
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> tsc .\main.ts
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> tsc .\main.ts
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1>
```

&gt; OUTLINE

&gt; DEPENDENCIES





EXPLORER

...

main.ts

exempleEspaceDeNomExterne.ts

index.html X



&gt; OPEN EDITORS

&lt; WORKSPACE-TYPESCRIPT

&gt; dist

&gt; node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

exemple1 &gt; HTML index.html

```
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Exemple de base</title>
8      <script src="exempleEspaceDeNomExterne.js"></script>
9      <script src="personne.js"></script>
10     <script src="main.js"></script>
11
12
```

Exemple de base

Hello !!!

localhost:3000

La somme est de 51

Console

main.js:2

```
:node
pace-typescript\exemple1> tsc .\main.ts
pace-typescript\exemple1> 
```

&gt; OUTLINE

&gt; DEPENDENCIES



EXPLORER

...

main.ts

exempleEspaceDeNomExterne.ts

index.html

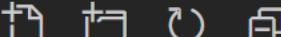


...

&gt; OPEN EDITORS 1 UNSAVED



WORKSPACE-TYPESCR...



- > dist
- > node\_modules
- JS etudiant.js
- JS exempleEspaceDeNomExterne.js
- TS exempleEspaceDeNomExterne.ts
- ES index.html
- JS main.js
- TS main.ts
- JS package-lock.json
- JS package.json
- JS personne.js
- TS personne.ts
- {..} tsconfig.json
- debug.log

&gt; OUTLINE

&gt; DEPENDENCIES

exemple1 &gt; ES index.html

```
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Exemple de base</title>
8     ... <!--<script src="exempleEspaceDeNomExterne.js"></script>-->
9     <script src="personne.js"></script>
10    <script src="main.js"></script>
11
12
```

PROBLEMS

TERMINAL

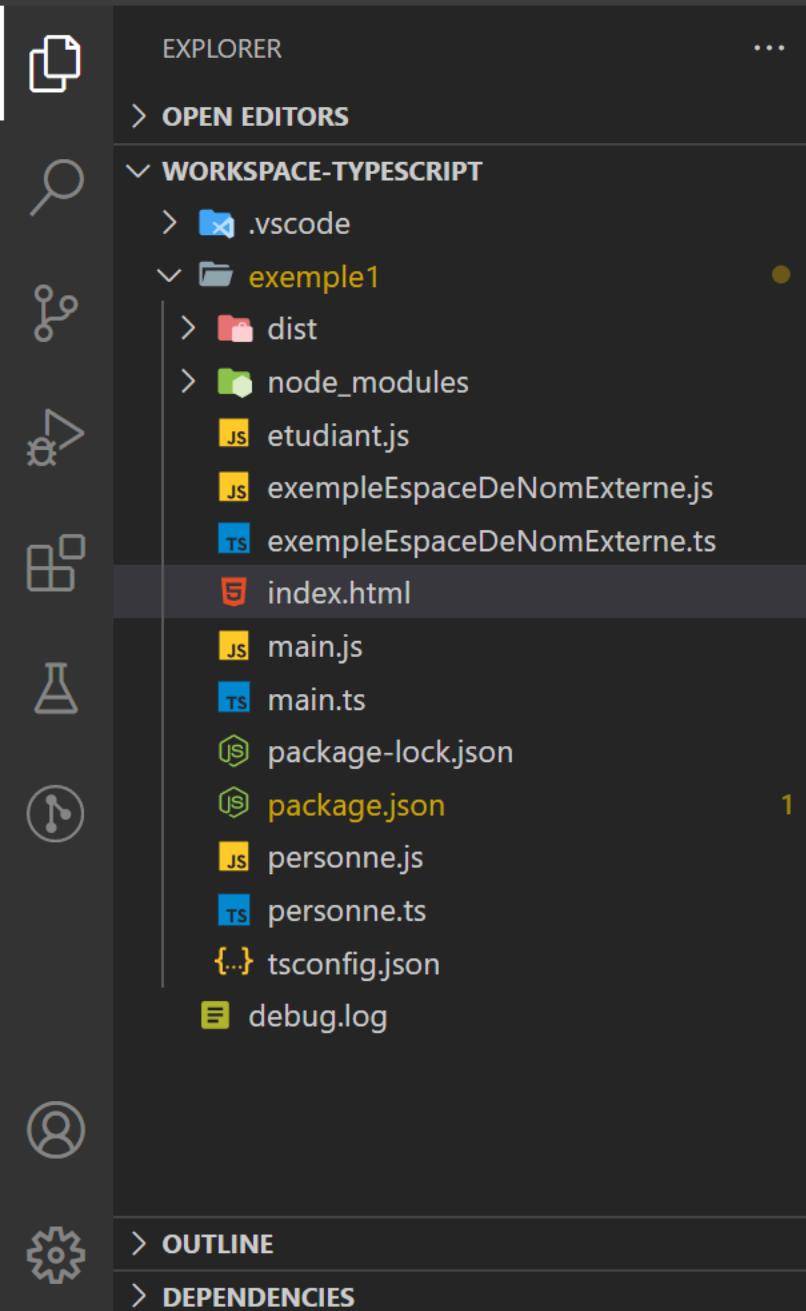
...

2: node



```
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> tsc .\main.ts
outFile .\exempleEspaceDeNomExterne.js
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1>
```

# Exemple avec un Module



```
main.ts          exempleEspaceDeNomExterne.ts      index.html X  (...) tsconfig.json

exemple1 > index.html
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Exemple de base</title>
8       <script src="exempleEspaceDeNomExterne.js"></script>
9       <script src="personne.js"></script>
10      <script src="main.js"></script>
11  </head>
12
13 <body>
14     Hello !!!
15 </body>
16
17 </html>
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: node < + ⊞ ^ ×

```
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> ts
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> ts
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> █
```

EXPLORER ...

> OPEN EDITORS

✓ WORKSPACE-TYPESCRIPT ⌂ ⌂ ⌂ ⌂ ⌂

> .vscode

< exemple1 ●

> dist

> node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

JSON package-lock.json

JSON package.json 1

JS personne.js

TS personne.ts

JSON tsconfig.json

Text debug.log

> OUTLINE

> DEPENDENCIES

main.ts exempleEspaceDeNomExterne.ts index.html personne.ts { } ...

exemple1 > TS personne.ts > {} persModule

```
1 module persModule{  
2     export function somme(x:number, y:number){  
3         return x+y;  
4     }  
5     export class Personne {  
6         constructor(private _nom?: string, private _age?: number) {}  
7  
8         public get nom(): string {  
9             return this._nom;  
10        }  
11  
12        public set nom(v: string) {  
13            this._nom = v;  
14        }  
15  
16        public get age(): number {  
17            return this._age;  
18        }  
19  
20        public set age(v: number) {  
21            this._age = v;  
22        }  
23        afficherPers() {  
24            console.log("Nom: " + this._nom + " Age: " + this._age + " ans");  
25        }  
26    }
```

File Edit Selection View Go Run Terminal Help main.ts - workspace-typescript - Visual Studio Code

EXPLORER ...

> OPEN EDITORS

✓ WORKSPACE-TYPESCRIPT

> .vscode

< exemple1

> dist

> node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

JSON package-lock.json

JSON package.json

JS personne.js

TS personne.ts

JSON tsconfig.json

Text debug.log

> OUTLINE

> DEPENDENCIES

EXPLORER ...

main.ts X exempleEspaceDeNomExterne.ts index.html {...} tsconfig.json

exemple1 > main.ts > ...

```
1 //<reference path="exempleEspaceDeNomExterne.ts" />
2 //<reference path=".//personne.ts" />
3
4 let pers=new persModule.Personne("Macky SALL",25);
5 pers.afficherPers();
6
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: node

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> tsc --target ES2020 .\personne.ts  
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> tsc --target ES2020 .\main.ts  
PS C:\Users\ousma\Desktop\workspace-typescript\exemple1>

Exemple de base - localhost:3000

Hello !!!

Console Elements Sources Network Performance Memory

top Filter Default levels 2 hidden

Nom: Macky SALL Age: 25 ans persone.js:25

Console

0 △ 1 Compilation Failed! Compile Hero: On Ln 5, Col 10 Spaces: 4 UTF-8 CRLF TS 4.0.3 Prettier: ✓

# TypeScript

Autre Configuration possible du  
projet utilisant les fichiers de  
configuration et un serveur

Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique  
[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Autre Configuration possible du projet utilisant les fichiers de configuration et un serveur

- Lancer dans le dossier du projet la commande npm init pour générer le fichier de configuration **package.json**
- Installer le serveur **lite-server** et le démarrer
- Executer la commande **tsc --init** pour générer le fichier de configuration **tsconfig.json** file.

- Le fichier **package.json** : contient l'ensemble des dépendances(genre pom.xml sous java) de l'application
- Le fichier **tsconfig.json** spécifie les fichiers racine et les options du compilateur nécessaires pour compiler le projet.



EXPLORER

...

main.ts



&gt; OPEN EDITORS 1 UNSAVED

WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1

debug.log

exemple1 &gt; main.ts &gt; [s] startAsync

```
24     const wait = (ms) => new Promise((res) => setTimeout(res, ms));
25     const startAsync = async (callback) => [
26         await wait(10000);
27         callback("Hello " + new Date());
28         await wait(20000);
29         callback("Et Bienvenue " + new Date());
30         await wait(30000);
31         callback("Async Await avec TypeScript " + new Date());
32     ];
33     startAsync((text) => console.log(text));
34
35     // const start = (callback) => {
36     //     setTimeout(() => {
37     //         callback("Hello " + new Date());
```

PROBLEMS

OUTPUT

TERMINAL

...

5: powershell

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

PS C:\Users\ousma\Desktop\workspace-typescript&gt; cd .\exemple1\

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1&gt; npm init



Compilation Failed!

Compile Hero: On

Ln 28, Col 21

Spaces: 4

UTF-8

CRLF

TS

4.0.3

Prettier: ✓





EXPLORER

...

main.ts

package.json X



...

&gt; OPEN EDITORS 1 UNSAVED

exemple1 &gt; package.json &gt; ...

```
1  {
2    "name": "exemple1",
3    "version": "1.0.0",
4    "description": "",
5    "main": "main.js",
6    ▶ Debug
7    "scripts": {
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC"
12 }
```

WOR...

&gt; .vscode

exemple1

&gt; dist

etudiant.js

etudiant.ts

index.html

main.js

main.ts

package.json 1

personne.js

personne.ts

debug.log

PROBLEMS

1

OUTPUT

TERMINAL

...

5: powershell



Is this OK? (yes) yes

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1&gt;



main.ts



package.json X



exemple1 &gt; package.json &gt; ...

```
1  {
2    "name": "exemple1",
3    "version": "1.0.0",
4    "description": "",
5    "main": "main.js",
6    ▶ Debug
7    "scripts": {
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "author": "",
11   "license": "TSC"
```



PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

5: node



Is this OK? (yes) yes

PS C:\Users\ousma\Desktop\workspace-typescript\exemple1> npm install lite-server --save-dev  
[redacted].....] | extract:safer-buffer: sill extract ms@2.0.0



EXPLORER

...

main.ts

package.json X



&gt; OPEN EDITORS 1 UNSAVED

## WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1 ●

&gt; dist

&gt; node\_modules

JS etudiant.js

TS etudiant.ts

ES index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json 1

JS personne.js

TS personne.ts

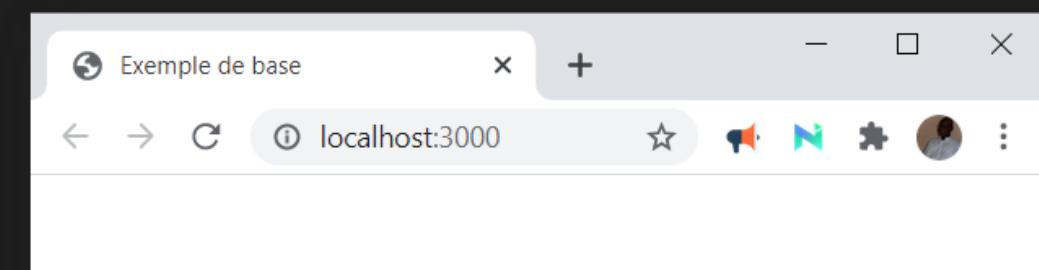
M debug.log

&gt; OUTLINE

&gt; DEPENDENCIES

exemple1 &gt; package.json &gt; {} scripts &gt; start

```
1  {
2      "name": "exemple1",
3      "version": "1.0.0",
4      "description": "",
5      "main": "main.js",
6      "scripts": {
7          "test": "echo \\\"Error: no test specified\\\" && exit 1",
8          "start": "lite-server"
9      },
10     "author": "",
11     "license": "ISC",
```



PROBLEMS

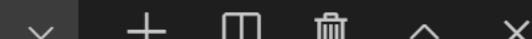
1

OUTPUT

DEBUG CONSOLE

TERMINAL

5: node



PS C:\Users\ousma\Desktop\workspace-typescript\exemple1&gt; npm start

> exemple1@1.0.0 start C:\Users\ousma\Desktop\workspace-typescript\exemple1  
> lite-server

Did not detect a `bs-config.json` or `bs-config.js` override file. Using lite-server defaults...



EXPLORER

...

main.ts



package.json

index.html X



...

&gt; OPEN EDITORS 1 UNSAVED

exemple1 &gt; index.html



WORKSPACE-TYPESCRIPT

&gt; .vscode

&lt; exemple1 ●

&gt; dist

&gt; node\_modules

JS etudiant.js

TS etudiant.ts

ES index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json 1

JS personne.js

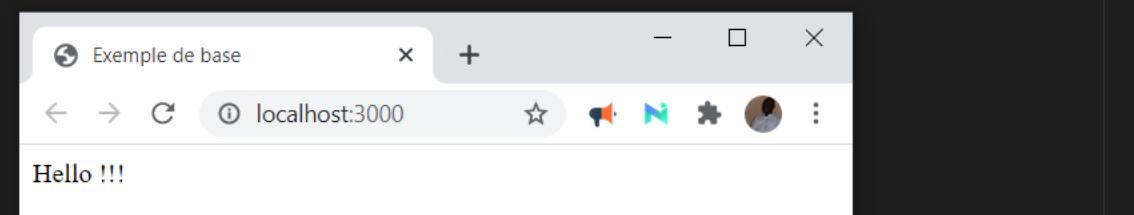
TS personne.ts

ES debug.log

&gt; OUTLINE

&gt; DEPENDENCIES

```
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Exemple de base</title>
8      <script src="personne.js"></script>
9      <script src="main.js"></script>
10     </head>
11
12     <body>
13         Hello !!!
14     </body>
```



PROBLEMS



OUTPUT

DEBUG CONSOLE

TERMINAL

5: node



```
20.10.16 01:43:36 200 GET /main.js
20.10.16 01:43:37 404 GET /favicon.ico
[Browsersync] Reloading Browsers... (buffered 2 events)
20.10.16 01:46:30 200 GET /index.html
20.10.16 01:46:30 304 GET /personne.js
20.10.16 01:46:30 304 GET /main.js
```





EXPLORER

...

main.ts

package.json

index.html X



...

&gt; OPEN EDITORS 1 UNSAVED

exemple1 &gt; index.html &gt; html &gt; body



WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1 ●

&gt; dist

&gt; node\_modules

JS etudiant.js

TS etudiant.ts

index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json 1

JS personne.js

TS personne.ts

debug.log



```
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Exemple de base</title>
8   <script src="personne.js"></script>
9   <script src="main.js"></script>
10  </head>
11
12 <body>
13   Hello !!!
14  </body>
```

PROBLEMS 1 OUTPUT

20.10.16 01:43:36 200

20.10.16 01:43:37 404

[Browsersync] Reloading

20.10.16 01:46:30 200 GET /index.html

20.10.16 01:46:30 304 GET /personne.js

20.10.16 01:46:30 304 GET /main.js

Exemple de base

localhost:3000

Hello !!!

Console

Default levels

main.js:1

main.js:20

main.js:4

main.js:10

main.js:15

### Début Fri Oct 16 2020 01:46:31 GMT+0000 (heure moyenne de Greenwich)

### Fin Fri Oct 16 2020 01:46:31 GMT+0000 (heure moyenne de Greenwich)

### Hello Fri Oct 16 2020 01:46:32 GMT+0000 (heure moyenne de Greenwich)

### Et Bienvue Fri Oct 16 2020 01:46:35 GMT+0000 (heure moyenne de Greenwich)

### Et après Fri Oct 16 2020 01:46:38 GMT+0000 (heure moyenne de Greenwich)





EXPLORER

...

main.ts



package.json

index.html



&gt; OPEN EDITORS 1 UNSAVED

exemple1 &gt; index.html &gt; html &gt; body



...

WORKSPACE-TYPESCRIPT

&gt; .vscode

&gt; exemple1



&gt; dist

&gt; node\_modules

JS etudiant.js

TS etudiant.ts

ES index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json 1

JS personne.js

TS personne.ts

M debug.log

```
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Exemple de base</title>
8      <script src="personne.js"></script>
9      <script src="main.js"></script>
10     </head>
11
12     <body>
13         Hello !!!
14     </body>
```

PROBLEMS



OUTPUT

DEBUG CONSOLE

TERMINAL

6: powershell



Windows PowerShell

Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

PS C:\Users\ousma\Desktop\workspace-typescript&gt; tsc --init



# TypeScript

## Les Décorateurs

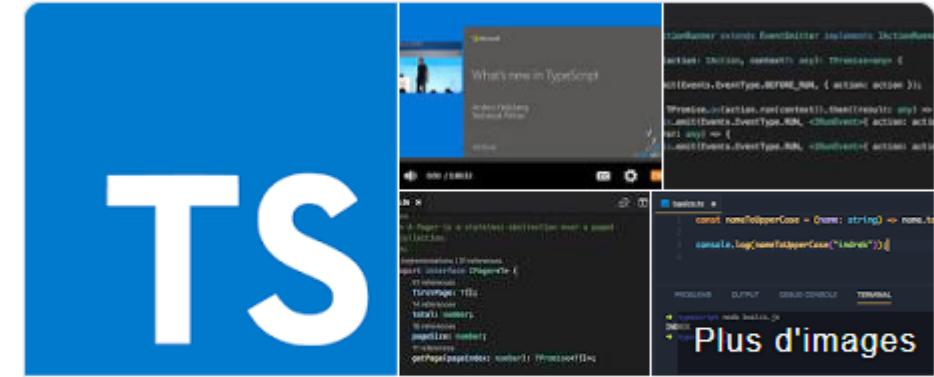
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)

# Contenu de la section

- Variables, Types de base et Opérateurs
- Les Structures de Contrôle
- Les Fonctions
- Les tableaux et Collections
- Programmation Orientée Objet
- Programmation Asynchrone
- Les Espaces de nom et les Modules
- **Les décorateurs**



## TypeScript

Langage de programmation

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript. [Wikipédia](#)

Conçu Par : [Microsoft](#)

Extension de fichier : `ts`

Date de première version : [9 février 2012](#)

Dernière version : [3.2.2 \(29 novembre 2018\)](#)

Typage : dynamique, faible, fort optionnel, statique optionnel

Paradigme : Multi-paradigme

# Décorateurs

- Les décorateurs permettent d'ajouter à la fois des annotations et une syntaxe de méta-programmation pour les déclarations de classe et les membres.
- Un décorateur est un type spécial de déclaration qui peut être attaché à une déclaration de classe, une méthode, un accesseur, une propriété ou un paramètre.
- Pour activer la prise en charge expérimentale des décorateurs, vous devez activer l'option du compilateur **experimentalDecorators** soit sur la ligne de commande, soit dans votre **tsconfig.json**

EXPLORER ...

> OPEN EDITORS

✓ WORKSPACE-TYPESCRIPT ⌂ ⌂ ⌂ ⌂ ⌂

- > .vscode
- ✓ exemple1
  - > dist
  - > node\_modules
  - JS etudiant.js
  - JS exempleEspaceDeNomExterne.js
  - TS exempleEspaceDeNomExterne.ts
  - HTML index.html
  - JS main.js
  - TS main.ts
  - JSON package-lock.json
  - JSON package.json
  - JS personne.js
  - TS personne.ts
- ... tsconfig.json
- debug.log

> OUTLINE

> DEPENDENCIES

ts main.ts ... tsconfig.json x

exemple1 > ... tsconfig.json > {} compilerOptions

```
51 "esModuleInterop": true
52 // "preserveSymlinks": > exper
53 // "allowUmdGlobalAccess": true,
54 /* Allow accessing UMD globals from mod
55 /* Source Map Options */
56 // "sourceRoot": "",
57 // "mapRoot": "",
58 // "inlineSourceMap": true,
59 // "inlineSources": true,
60
61 /* Experimental Options */
62 "experimentalDecorators": true,
63 // "emitDecoratorMetadata": true,
64
65 /* Advanced Options */
66 "skipLibCheck": true,
67 "forceConsistentCasingInFileNames": true
68 }
69 }
70 }
```

Aa AbI \* 2 of 4 ↑ ↓ ≡ x twe...  
syml...  
/\* Specify the location where debugger...  
/\* Specify the location where debugger...  
/\* Emit a single file with source maps...  
/\* Emit the source alongside the source...  
/\* Enables experimental support for ES7 de...  
/\* Enables experimental support for emi...  
/\* Skip type checking of declaration fi...  
/\* Disallow inconsistently-cased refere...

# Décorateurs

- Le rôle des décorateurs est de factoriser certains "patterns" en surchargeant le comportement natif.
- Les décorateurs utilisent la forme **@expression**, où **expression** doit s'évaluer en une fonction qui sera appelée au moment de l'exécution avec des informations sur la déclaration décorée.
- **Decorator Factories**: Si nous voulons personnaliser la façon dont un décorateur est appliqué à une déclaration, nous pouvons écrire une fabrique de décorateurs. Une **Decorator Factory** est simplement une fonction qui renvoie l'expression qui sera appelée par le décorateur lors de l'exécution.

# Décorateurs

- Un décorateur est une fonction et sa signature dépend de la cible. Une signature pour les décorateurs de classe et de méthode ne sera pas la même. Voici les quatre signatures des décorateurs:
  - `declare type ClassDecorator = <TFunction extends Function>(target: TFunction) => TFunction | void;`
  - `declare type PropertyDecorator = (target: Object, propertyKey: string | symbol) => void;`
  - `declare type MethodDecorator = <T>(target: Object, propertyKey: string | symbol, descriptor: TypedPropertyDescriptor<T>) => TypedPropertyDescriptor<T> | void;`
  - `declare type ParameterDecorator = (target: Object, propertyKey: string | symbol, parameterIndex: number) => void;`

File Edit Selection View Go Run Terminal Help main.ts - workspace-typescript - Visual Studio Code

EXPLORER ...

> OPEN EDITORS

✓ WORKSPACE-TYPESCRIPT

> .vscode

> exemple1

> dist

> node\_modules

JS etudiant.js

JS exempleEspaceDeNomExterne.js

TS exempleEspaceDeNomExterne.ts

HTML index.html

JS main.js

TS main.ts

JS package-lock.json

JS package.json

JS personne.js

TS personne.ts

{...} tsconfig.json

debug.log

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: node

Le décorateur affiche le contenu du constructeur

```
1  function Connexion(f:Function){  
2  |   console.log(f);  
3  }  
4  
5  @Connexion  
6  class MaClasse{  
7  |   constructor(){  
8  |       console.log("Hello !");  
9  }  
10 }
```

20.10.16 06:47:09 304 GET /personne.js  
20.10.16 06:47:09 200 GET /main.js  
20.10.16 06:47:09 304 GET /exempleEspaceDeNomExterne.js  
20.10.16 06:47:09 304 GET /personne.js  
20.10.16 06:47  
20.10.16 06:47  
20.10.16 06:47  
20.10.16 06:47  
Hello !!!

Exemple de base Exemple de base

localhost:3000

Console

Elements Sources Network Performance Memory

MaClasse() {  
 console.log("Hello !");  
}

Console

0 △ 0 ➤ X Compilation Failed! ⚡ Compile Hero: On Ln 6, Col 16 Spaces: 4 UTF-8 CRLF TS 4.0.3 Prettier: ✓

The screenshot shows a development environment with a code editor and a browser window.

**Code Editor:**

- Title Bar:** File Edit Selection View Go Run ...
- File Explorer (Left):** Shows a file named `main.ts` with a blue icon and a blue circle containing the number 1.
- Search (Search Bar):** A magnifying glass icon.
- Problems (Bottom):** Shows a red X icon and the message "Compilation Failed!"
- Code Area:**

```
exemple1 > main.ts > ...
1  function Connexion(f: Function) {
2  |   console.log(f);
3  }
4
5  // Decorator Factories
6  function seConnecter(val: boolean) {
7  |   return val ? Connexion : null;
8  }
9
10 @seConnecter(false) //Malasse2 ne va pas s'afficher
11 class Malasse2 {}
12
13 @seConnecter(true) //Malasse3 va s'afficher
14 class Malasse3 {}
15
16
```
- Status Bar:** Ln 18, Col 1 Spaces: 4 UTF-8 CRLF TS 4.0.3 Prettier: ✓

**Browser Window (Top Right):**

- Title Bar:** Exemple de base
- Address Bar:** localhost:3000
- Console Tab:** Elements, Console (highlighted), Sources, Network
- Console Output:**

```
Hello !!!
f MaClasse() {
    console.log("Hello !");
}
f Malasse3()
>
```
- Bottom Bar:** Console

# Décorateurs

- Dans Angular, les classes et leurs propriétés sont définies par les décorateurs.
- Les décorateurs Angular ressemblent aux annotations (Java, C#) et sont utilisés avec le signe **@** préfixé.
- Dans Angular, presque tout est essentiellement une classe. Mais les décorateurs les transforment en **module**, **composant**, **service**, **pipe** ou **directive**.
- Voici quelques décorateurs utilisés dans Angular

**@NgModule**

**@Pipe**

**@Input**

**@Component**

**@Injectable**

**@Output**

**@Directive**

**@Inject**

**@HostListener**

# Travaux Pratiques n°1

<https://codelab.fun/angular/typescript>

Travaux Pratiques n°1

# Préparer la fiche de TP n°1 sur TypeScript

# Webographie

- <https://angular.io/>
- <http://typescriptlang.org>
- <https://www.youtube.com/watch?v=KJEJejxcnEQ&list=PLxr551TUsmAphhzmpn6lVyRGjAC1URrO>
- [https://www.youtube.com/results?search\\_query=TypeScript](https://www.youtube.com/results?search_query=TypeScript)
- [https://www.youtube.com/results?search\\_query=Angular](https://www.youtube.com/results?search_query=Angular)
- Tutos sur Google et Youtube entre autres.



Visual Studio Code



Firebase



TypeScript



# Développement d'applications web avec le Framework front-end Angular



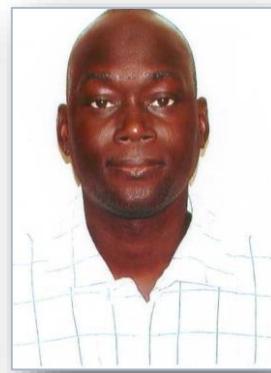
Prof. Ousmane SALL

Université de THIES - UFR Sciences et Technologies -Département Informatique

[osall@univ-thies.sn](mailto:osall@univ-thies.sn)



# A propos de moi

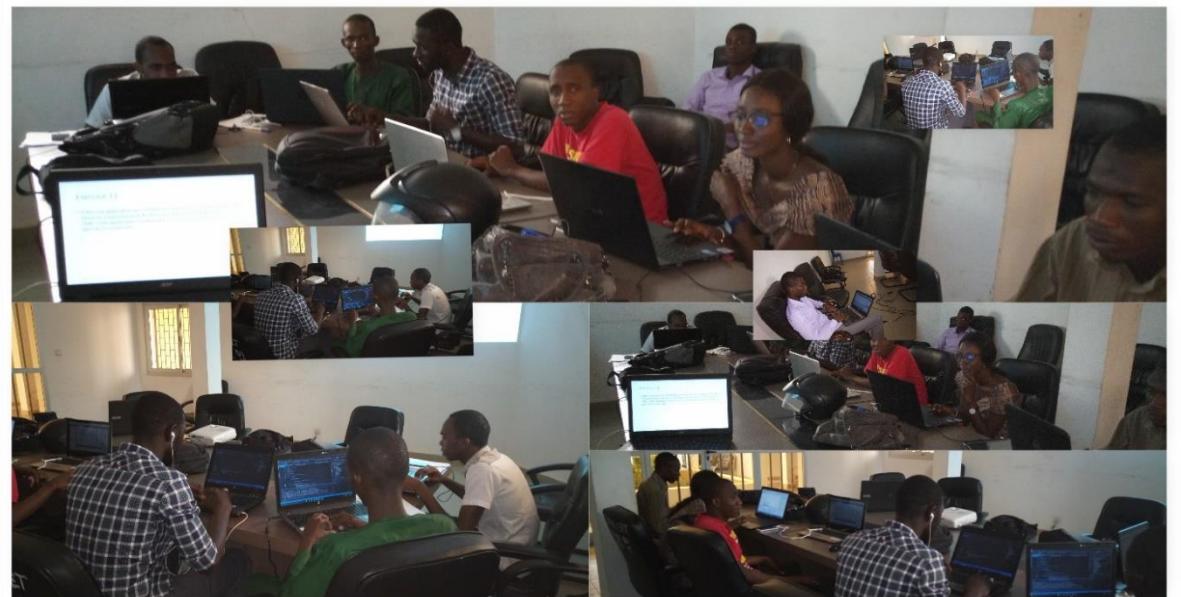


- Enseignant-Chercheur à l'UFR SET- Université de THIES <http://sites.univ-thies.sn/osall751/>
- Enseignements:
  - Algorithmique et Programmation(C, Java, PHP)
  - Programmation WEB dynamique(HTML 5 CSS, PHP, MySQL, CMS,...)
  - Programmation Java
  - Programmation JavaEE, JSF, Spring
  - Technologies Mobiles Android, Xamarin, Ionic
  - Programmation .Net, C#
  - Gestion de Projet Informatique
  - Génie Logiciel
- Contact:
  - [osall@univ-thies.sn](mailto:osall@univ-thies.sn)
  - UFR SET, Université de THIES -Dpt Informatique, BP 967 THIES.



# Une sagesse chinoise...

*« J'écoute et j'oublie; je lis et je comprends; je fais et j'apprends »*  
[Proverbe chinois]



# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



Application Angular  
TypeScript

Application  
Mobile

IOT

RestFul API  
Spring Boot

Base de  
Données



# Les composants

# Contenu Cours Angular

- **Composants**
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage

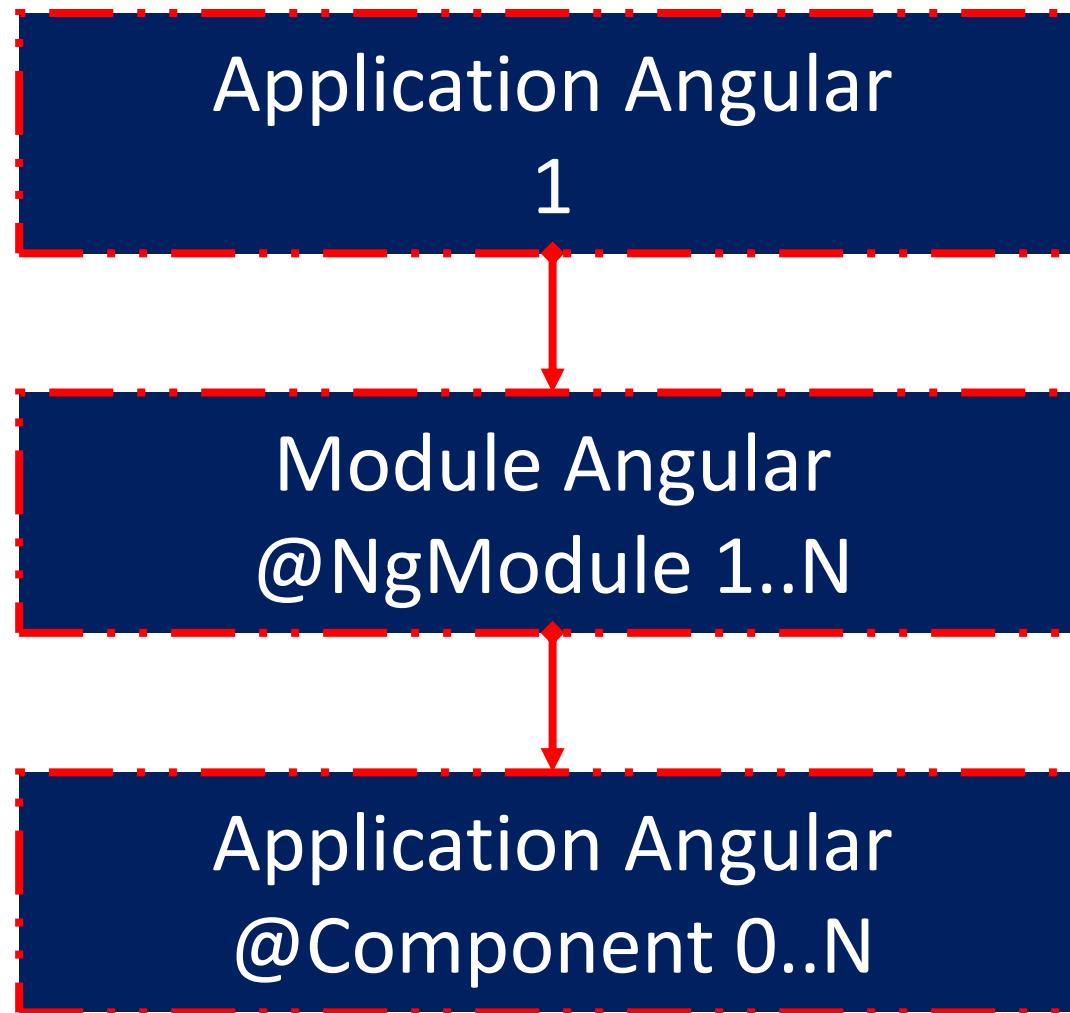


# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Composants, Modules et Décorateurs



# Angular : une Architecture d'une Application basée sur les composants



- Exemple simple avec 05 composants fils du composant racine **app-root**
- **AppComponent** est le composant principal : tous les autres composants seront emboîtés ou "nested" dans celui-ci.
- **Un Module est ici un groupe de composants** par exemple **app.module.ts**

# Composants

Chaque composant se compose principalement des éléments suivants:

**HTML Template** : représentant sa  
**vue**(personne.component.html)

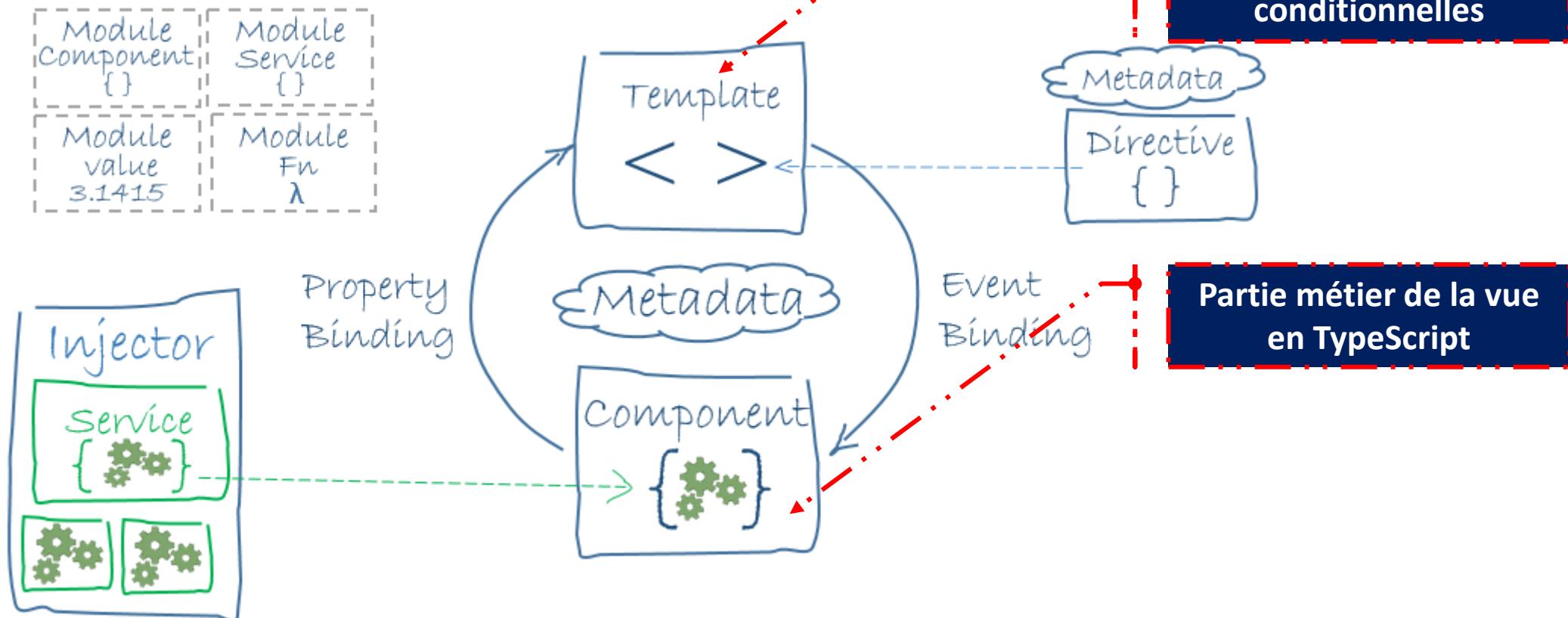
**Composant**

Une **classe typescript** représentant sa  
**logique métier** (personne.component.ts)

**Feuille de style CSS** pour le  
**design** (personne.component.css)

**Fichier spec** sont des **tests**  
**unitaires**(personne.component.spec.ts)

# Architecture du Framework Angular



Une application Angular se compose de un à plusieurs modules dont un est principal. Chaque module peut inclure : Des composants web : La partie visible de l'application Web (IHM) Des services pour la logique applicative.

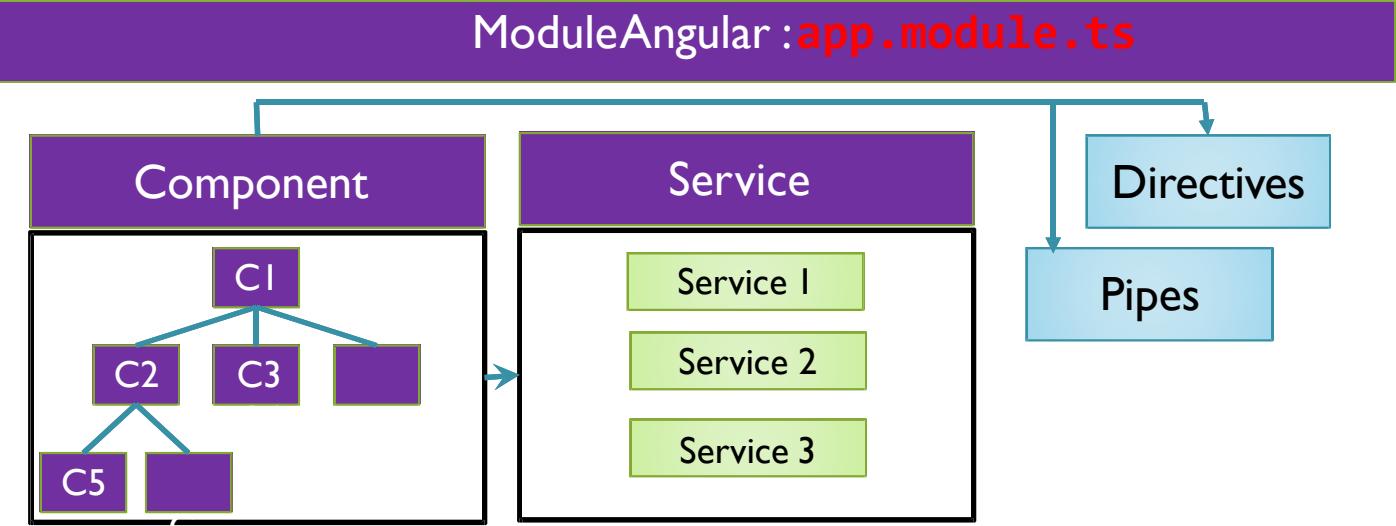
# Architecture du Framework Angular

Une application **Angular** se compose d'un à plusieurs **modules** dont un est principal **app.module.ts**

## Chaque module peut inclure :

- Des **composant web** : La partie visible de l'application Web (IHM)
- Des **services** pour la logique applicative. Les composants peuvent utiliser les services via le principe de **l'injection des dépendances**.
- Les **directives** : un composant peut utiliser des directives
- Les **pipes** : utilisés pour formater l'affichage des données dans les composants.

## Chaque composant(@Component) doit être associé à un module(@NgModule)



EXPLORER

> OPEN EDITORS

PREMIEREAPPANG... 34

- app
  - app-routing.module.ts A
  - app.component.css A
  - app.component.html M
  - app.component.spec.ts A
  - app.component.ts M
  - app.module.ts A
- assets
- environments
- favicon.ico A
- index.html A
- main.ts A
- polyfills.ts A
- styles.css A

> OUTLINE

> TIMELINE

> DEPENDENCIES

A app.component.ts X

src > app > A app.component.ts > AppComponent > title

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = "Exemples du cours d'Angular";
10 }
11
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANG...

- app
  - app-routing.module.ts A
  - app.component.css A
  - app.component.html A
  - app.component.spec.ts A
  - app.component.ts A
  - app.module.ts A
- assets
- environments
- favicon.ico
- index.html A
- main.ts A
- polyfills.ts A
- styles.css A

> OUTLINE

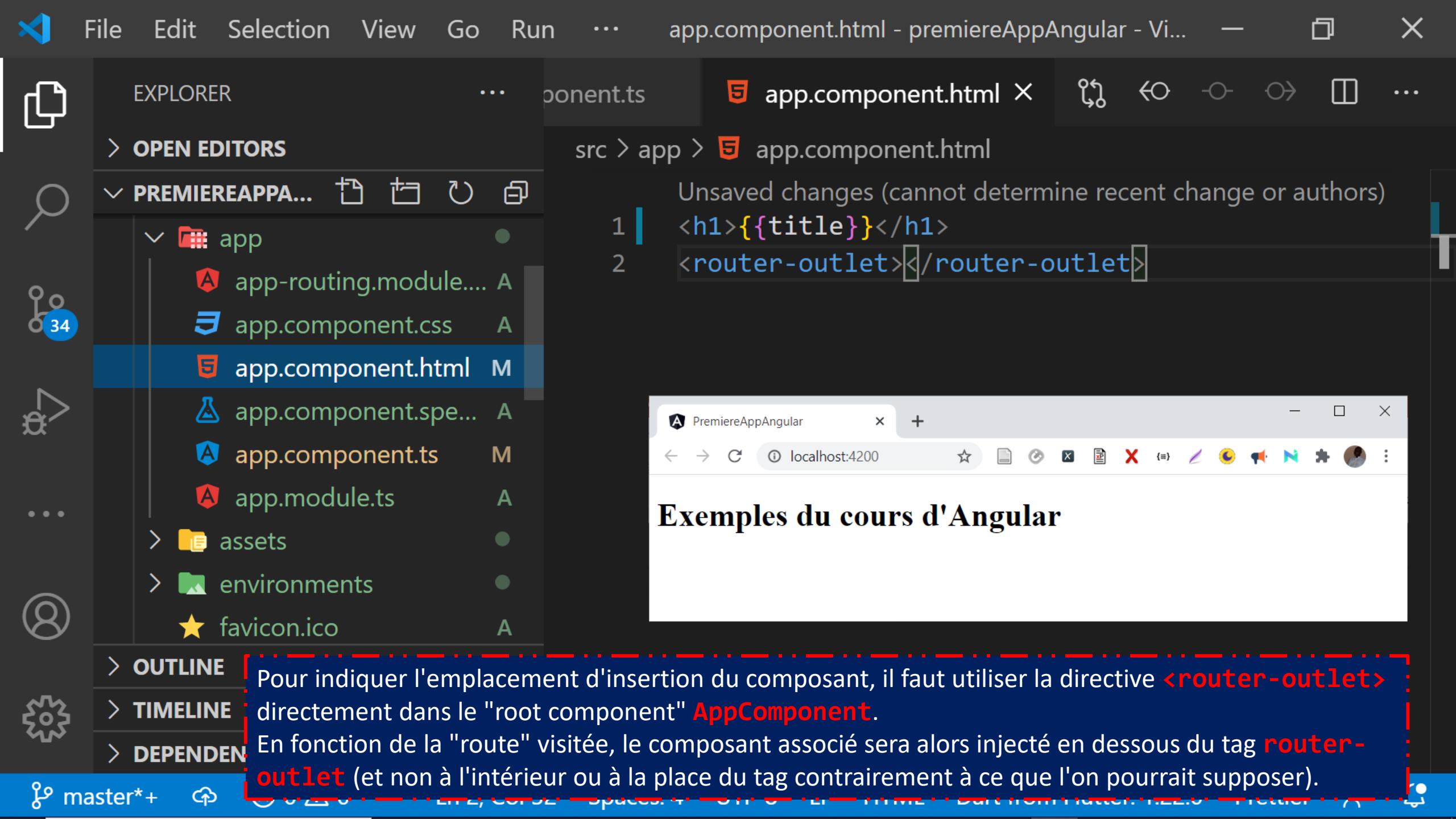
> TIMELINE

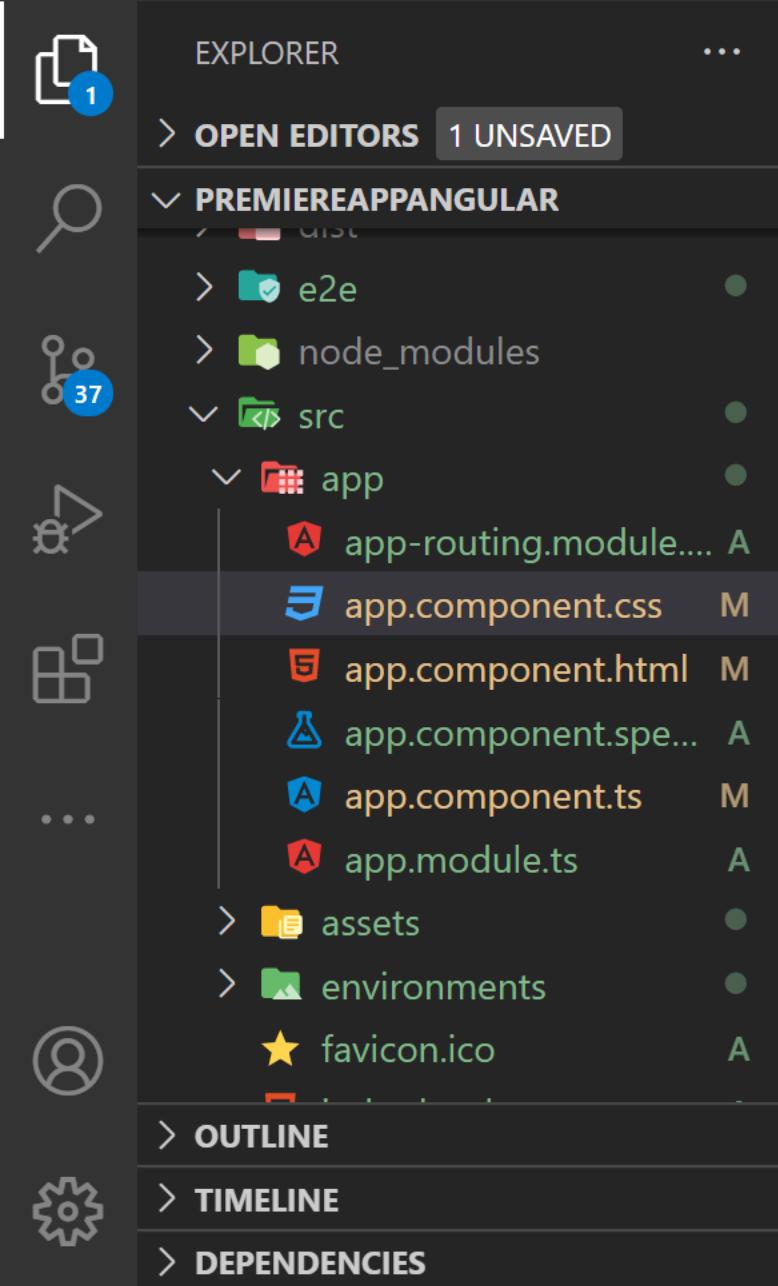
> DEPENDENCIES

A app.module.ts X

src > app > A app.module.ts > ...

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 
7 @NgModule({
8   declarations: [
9     AppComponent
10   ],
11   imports: [
12     BrowserModule,
13     AppRoutingModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
19 
```





ENT.ts app.component.html ● app.component.css X 1 2 3 4 5

src > app > app.component.css > h1

Unsaved changes (cannot determine recent change or authors)

```
1 h1{  
2   text-align: center;  
3   background-color: black;  
4   color: white;  
5 }
```



# Utiliser Bootstrap

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a tab for app.component.html - premiereAppAngular - Vis. The left sidebar has icons for files (1), search (34), symbols, settings, and user profile. The main editor area shows app.component.ts and app.component.html. The code in app.component.html is:

```
src > app > app.component.html > router-outlet
1 |   <h1>{{title}}</h1>
2 |   <router-outlet></router-outlet>
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and a terminal window titled "2: powershell" showing PowerShell output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> npm install bootstrap --save
```

The status bar at the bottom shows master\*, 0 △ 0, Compile Hero: Off, Ln 2, Col 33, Spaces: 4, UTF-8, LF, HTML, Dart from Flutter: 1.22.0, Prettier, 236.

**Bootstrap**

Front-end framework

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. [Wikipedia](#)

Original author(s): Mark Otto, Jacob Thornton

License: MIT License (Apache License 2.0 prior to 3.1.0)

Stable release: 4.4.1 / November 28, 2019; 4 months ago

Initial release date: August 19, 2011

Platform: Web platform

Written in: HTML, Cascading Style Sheets, Less (v3), Sass (v4), JavaScript



EXPLORER

> OPEN EDITORS

PREMIEREAPPA... 37

- test.ts
- .browserslistrc
- .editorconfig
- .gitignore
- angular.json
- karma.conf.js
- package-lock.json
- package.json M
- README.md
- tsconfig.app.json
- tsconfig.json
- tsconfig.spec.json
- tslint.json

> OUTLINE

> TIMELINE

> DEPENDENCIES

package.json X

```
private : true,
"dependencies": {
  "@angular/animations": "~10.1.5",
  "@angular/common": "~10.1.5",
  "@angular/compiler": "~10.1.5",
  "@angular/core": "~10.1.5",
  "@angular/forms": "~10.1.5",
  "@angular/platform-browser": "~10.1.5",
  "@angular/platform-browser-dynamic": "~10.1.5",
  "@angular/router": "~10.1.5",
  "bootstrap": "^4.5.3",
  "rxjs": "~6.6.0",
  "tslib": "^2.0.0",
  "zone.js": "~0.10.2"
},
"devDependencies": {
  "@angular-devkit/build-angular": "~0.1001.6",
  "@angular/cli": "~10.1.6",
  "@angular/compiler-cli": "~10.1.5",
  "@types/node": "^12.11.1",
  "@types/jasmine": "~3.5.0",
}
```



EXPLORER

...

bootstrap.css X

≡ ⌂ ⌂ ...

&gt; OPEN EDITORS

PREMIEREAPPANGULAR

&gt; bonjour

&gt; boolbase

bootstrap

dist

css

bootstrap-grid.css

bootstrap-grid.css.map

bootstrap-grid.min.css

bootstrap-grid.min.css....

bootstrap-reboot.css

bootstrap-reboot.css....

bootstrap-reboot.min....

bootstrap-reboot.min....

bootstrap.css

bootstrap.css.map

bootstrap.min.css

bootstrap.min.css.map

js

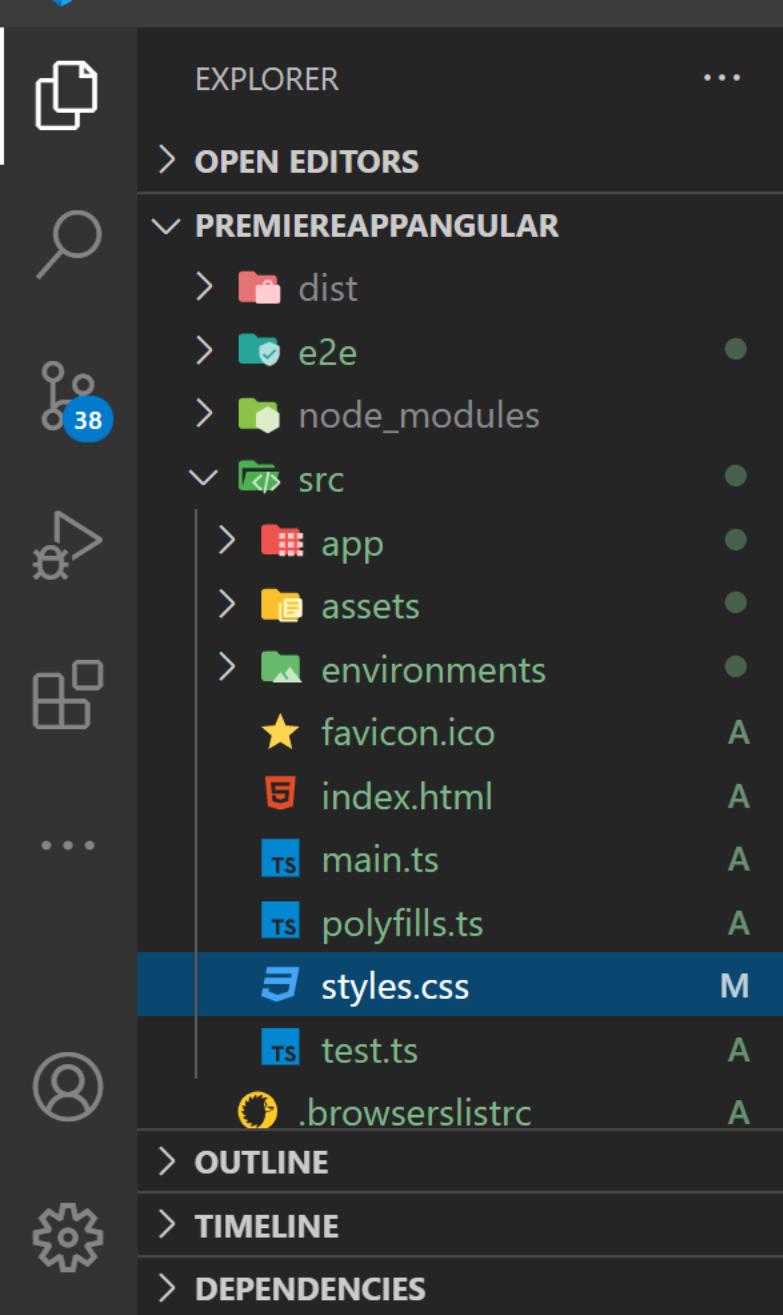
&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

node\_modules &gt; bootstrap &gt; dist &gt; css &gt; bootstrap.css &gt; ...

```
1  /*!
 2   * Bootstrap v4.5.3 (https://getbootstrap.com/)
 3   * Copyright 2011-2020 The Bootstrap Authors
 4   * Copyright 2011-2020 Twitter, Inc.
 5   * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 6   */
 7   :root {
 8     --blue: #007bff;
 9     --indigo: #6610f2;
10     --purple: #6f42c1;
11     --pink: #e83e8c;
12     --red: #dc3545;
13     --orange: #fd7e14;
14     --yellow: #ffc107;
15     --green: #28a745;
16     --teal: #20c997;
17     --cyan: #17a2b8;
18     --white: #fff;
19     --gray: #6c757d;
20     --gray-dark: #343a40;
21     --primary: #007bff;
22     --secondary: #6c757d;
23     --success: #28a745;
24     --info: #17a2b8;
25     --warning: #ffc107;
26     --danger: #dc3545;
```



styles.css X

src > styles.css

```
1  /* You can add global styles to this file, and also import other style
2  @import "~bootstrap/dist/css/bootstrap.css"
```

EXTENSIONS: MAR... ...

bootstrap

**Bootstrap 4, Font awesom...** 6.1.0  
Bootstrap 4 snippets based on d...  
Ashok Koyi [Install](#)

**Bootstrap 3 Snippets** 0.1.0  
Bootstrap 3 Snippets for VS Code  
William Whitehead [Install](#)

**Bootstrap v4 Snippets** 1.1.3  
Bootstrap v4 Snippets for VS Code  
Zaczero [Installing](#)

**Bootstrap 4 CDN Snippet** 1.9.0  
This will get you up and running ...  
Simen Daehlin [Install](#)

**Angular UI Bootstrap Snip...** 4.0.9  
Angular UI Bootstrap Snippets fo...  
HerrHerrmann [Install](#)

**bootstrap-snippets-vscode** 0.3.0  
bhavana [Install](#)



# Bootstrap v4 Snippets

Zaczero | ⚡ 280 338 | ★★★★★ | Repository | License

Bootstrap v4 Snippets for VS Code

Installing

[Details](#) [Feature Contributions](#) [Changelog](#)

EXPLORER

> OPEN EDITORS

✓ PREMIEREAPPA... + - ⌂ ⌂ ⌂ ⌂

- > dist
- > e2e
- > node\_modules
- ✓ src
  - ✓ app
    - A app-routing.module.... A
    - Ξ app.component.css M
    - S app.component.html M
    - ⚠ app.component.spe... A
    - A app.component.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

Ξ app.component.css X

src > app > Ξ app.component.css > h1

```
1 h1{  
2     background-color: black;  
3     color: white;  
4 }
```

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF CSS SCSS Formatter Prettier

EXPLORER

> OPEN EDITORS

✓ PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- ✓ src
  - > app
    - A app-routing.module.... A
    - Ξ app.component.css M
    - Ѕ app.component.html M
    - Ξ app.component.spec.... A
    - A app.component.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

app.component.html X

src > app > app.component.html

```
1 <div class="container">
2   <h1 class="text-center">{{title}}</h1>
3   <router-outlet></router-outlet>
4 </div>
```

Exemples du cours d'Angular

master\*+ ⚡ 0 ▲ 0

Ln 1, Col 24

Spaces: 4

UTF-8

LF

HTML

Dart from Flutter: 1.22.0

Prettier

↻

⌚



app.component.html

app.component.ts X



src &gt; app &gt; app.component.ts &gt; AppComponent

Unsaved changes (cannot determine recent change or authors)

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

2

Unsaved changes (cannot determine recent change or authors)

3 @Component({

4 selector: 'app-root',

5 // templateUrl: './app.component.html',

6 template: "&lt;h1 class='text-center'&gt;Titre: {{title}}&lt;/h1&gt;",

7 styleUrls: ['./app.component.css']

8 })

9 export class AppComponent {

10 | title = "Exemples du cours d'Angular";

11 }

12



master\*+



0 △ 0

Ln 6, Col 11 (8 selected)

Spaces: 2

UTF-8

LF

TS

4.0.3

Prettier



Pour générer un nouveau composant **accueil** avec CLI  
**ng generate(ou g) component(ou c) nomDuComposant**



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

3: powershell

▼ + □ └ ▲ ▾ ×

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> **ng g c accueil** ]



EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- ✓ src
  - ✓ app
    - > accueil
      - accueil.component.css U
      - accueil.component.html U
      - accueil.component.spec.ts U
      - accueil.component.ts U
    - app-routing.module.ts A
    - app.component.css M
    - app.component.html M
    - app.component.spec.ts A
    - app.component.ts M
    - app.module.ts M
  - > assets
  - > environments

> OUTLINE

> TIMELINE

> DEPENDENCIES

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

3: powershell

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g c accueil
CREATE src/app/accueil/accueil.component.html (22 bytes)
CREATE src/app/accueil/accueil.component.spec.ts (633 bytes)
CREATE src/app/accueil/accueil.component.ts (279 bytes)
CREATE src/app/accueil/accueil.component.css (0 bytes)
UPDATE src/app/app.module.ts (479 bytes)
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> 
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

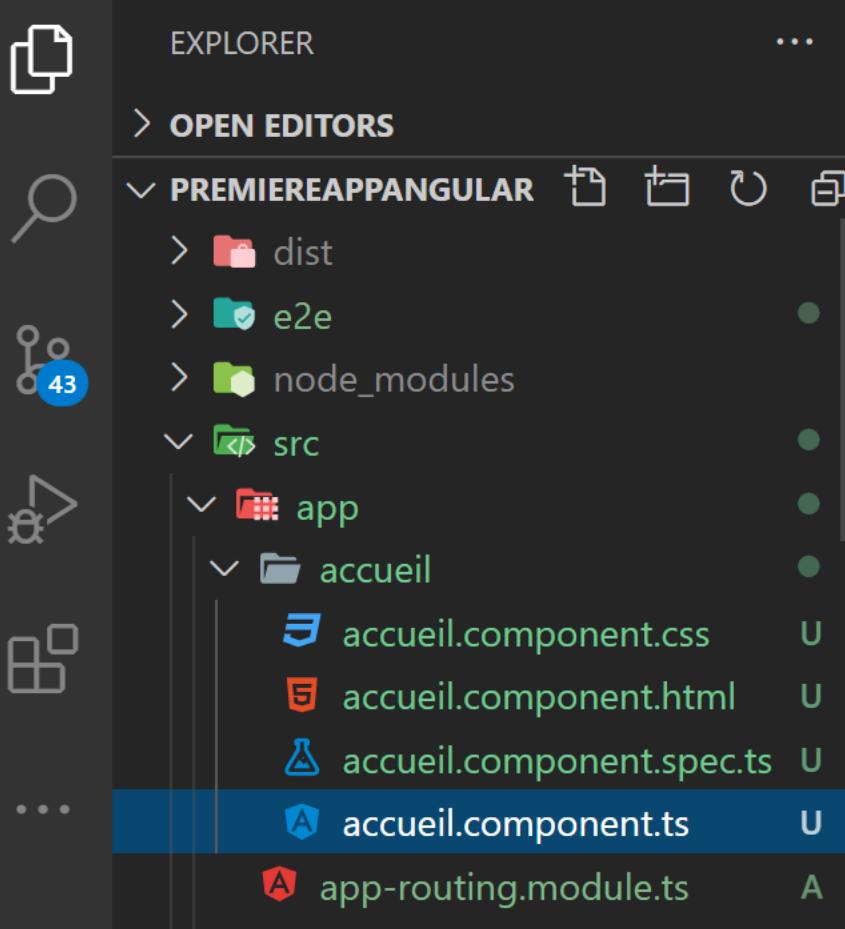
- > dist
- > e2e
- > node\_modules
- 43
- > src
  - app
    - accueil
      - accueil.component.css
      - accueil.component.html
      - accueil.component.spec.ts
      - accueil.component.ts
    - app-routing.module.ts
    - app.component.css
    - app.component.html
    - app.component.spec.ts
    - app.component.ts
  - app.module.ts
- > assets

## app.module.ts ×

```
src > app > A app.module.ts > ...
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { AccueilComponent } from './accueil/accueil.component';
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     AccueilComponent
12   ],
13   imports: [
14     BrowserModule,
15     AppRoutingModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
```

Déclaration automatique du composant créé dans le module principal

- **import:** Définit la liste des dépendances du module. Il s'agit généralement de la liste des modules contenant les composants utilisés par les composants de la section declarations.
- **déclarations:** Définit la liste des composants (ou directives, pipes etc...) contenus dans ce module.. Seuls les éléments déclarables (composants, directives et pipe) sont dans le tableau des déclarations.
- **imports:** Définit la liste des dépendances du module. Il s'agit généralement de la liste des modules contenant les composants utilisés par les composants de la section declarations..
- **providers:** tableau contenant une liste de services exposés à l'intérieur votre module et que les composants et autres services de votre module utiliseront.
- **bootstrap:** tableau que vous ne verrez généralement que dans le fichier app.module.ts.



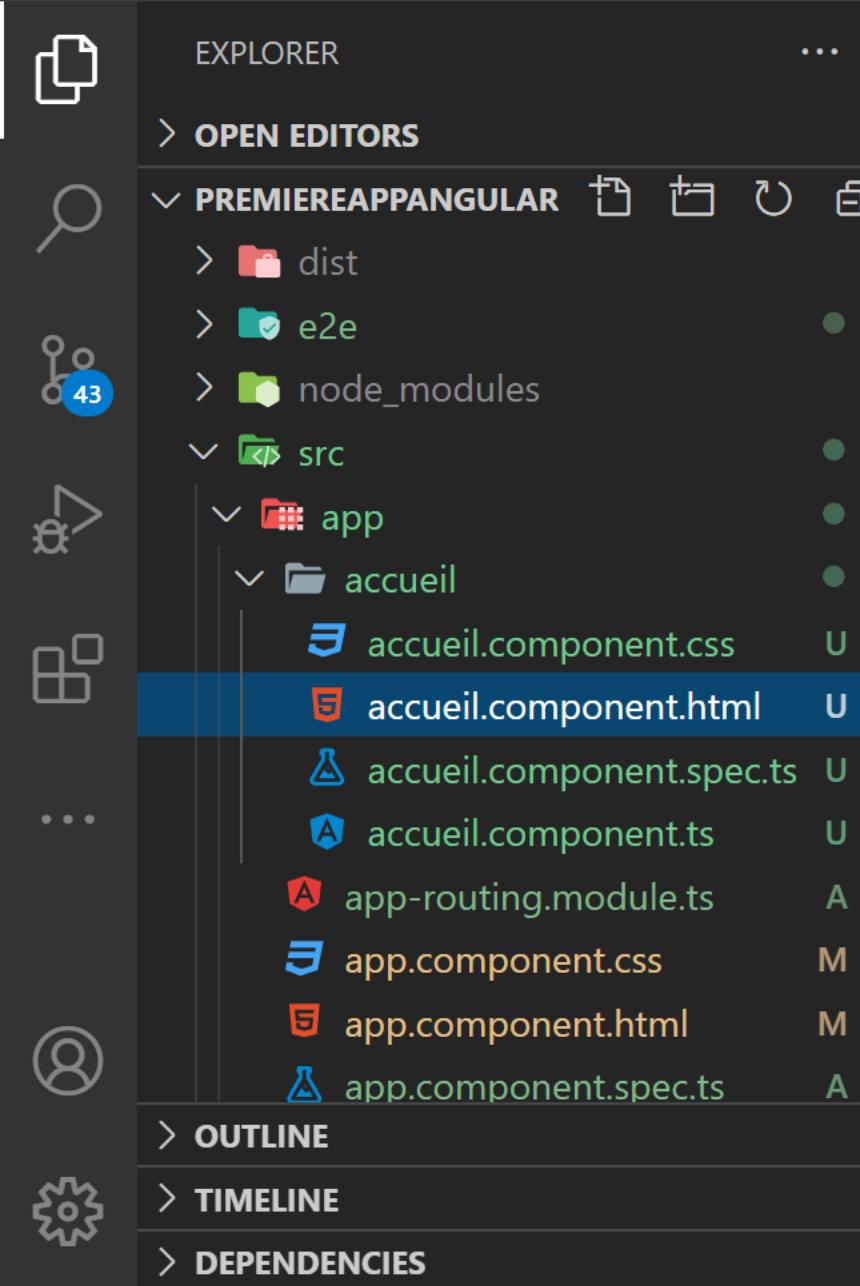
A accueil.component.ts X

```
src > app > accueil > A accueil.component.ts > AccueilComponent
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-accueil',
5   templateUrl: './accueil.component.html',
6   styleUrls: ['./accueil.component.css']
7 })
8 export class AccueilComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit(): void {
13   }
14
15 }
```

**@Component** : décorateur qui indique à Angular que cette classe joue le rôle d'un composant avec :

- **selector** : pour définir le nom de balise correspondant à ce composant web
  - **templateUrl** : pour indiquer le fichier contenant le code HTML correspondant au composant web
  - **styleUrls** : pour indiquer le(s) fichier(s) contenant le code CSS correspondant au composant web

**ngOnInit()** : Méthode du cycle de vie sera déclenchée une fois après l'initialisation du composant.



accueil.component.html X

src > app > accueil > accueil.component.html

```
1 <div class="container">
2   <p>accueil works!</p>
3 </div>
4
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- > src
  - > app
    - > accueil
    - app-routing.module.ts
    - app.component.css
    - app.component.html
    - app.component.spec.ts

...

> OUTLINE

> TIMELINE

> DEPENDENCIES

master\*+ 0 0 0

app.component.html X

src > app > app.component.html

Unsaved changes (cannot determine recent change or authors)

```
1 <div class="container">
2   <h1 class="text-center">{{title}}</h1>
3 </div>
4
5 <app-accueil></app-accueil>
6
7 <router-outlet></router-outlet>
```

PremiereAppAngular - localhost:4200

Exemples du cours d'Angular

accueil works!

Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier

# Deux autres manières d'inclure un composant : comme classe ou comme propriété

- Le nom de l'attribut associé à ce composant :
  - **selector** : [app-root]
  - Dans ce cas le composant sera inséré par : <div app-root></div>
- Le nom de la classe associé à ce composant :
  - **selector** : .app-root
  - Dans ce cas le composant sera inséré par : <div class="app-root"></div>



app.component.html X



src &gt; app &gt; app.component.html &gt; ...

Unsaved changes (cannot determine recent change or authors)

```
1 <div class="container">
2   |   <h1 class="text-center">{{title}}</h1>
3 </div>
4
5 <app-accueil></app-accueil>
6
7 <router-outlet></router-outlet>
```



PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

3: powershell



PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular&gt; ng g c login

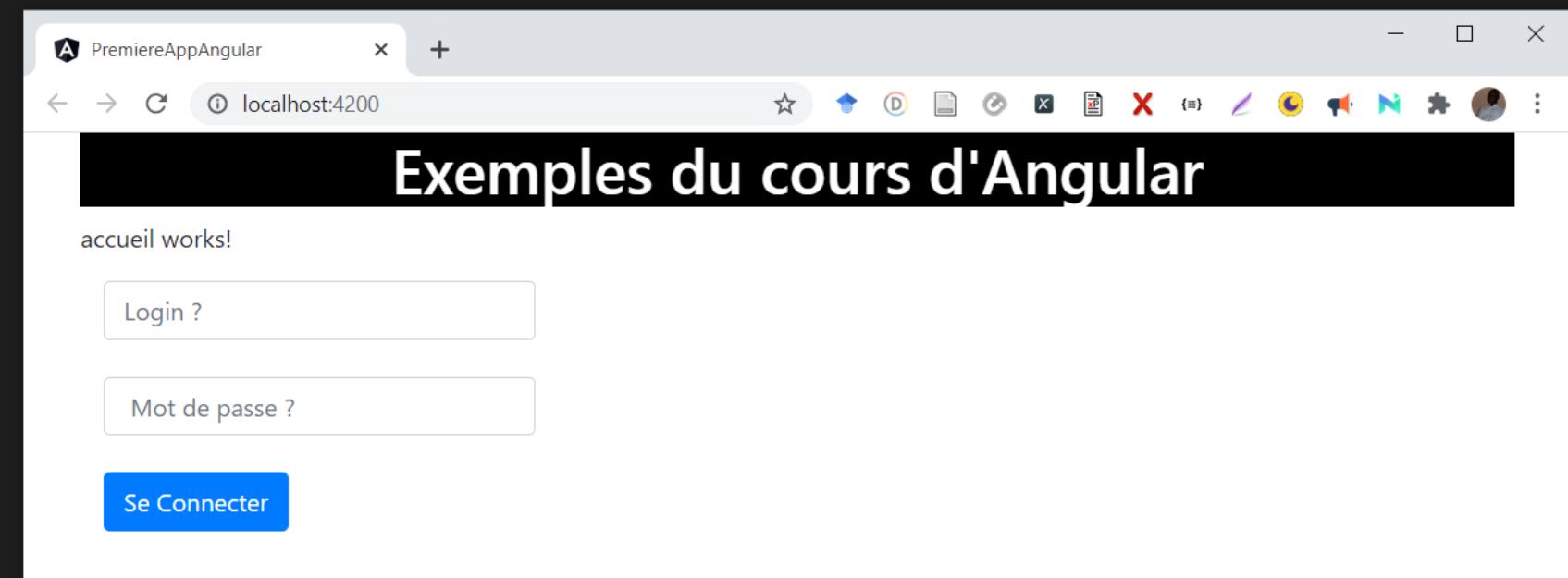


login.component.html X



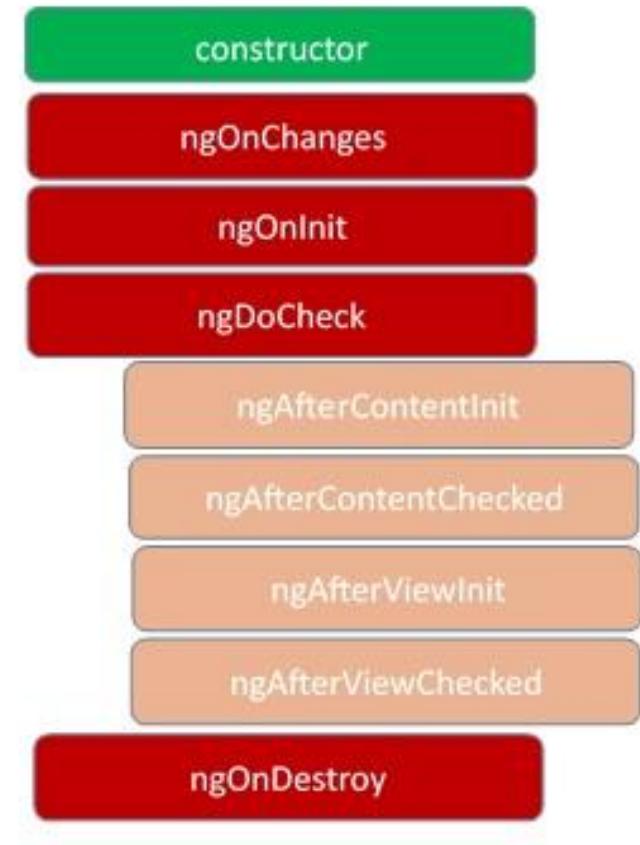
src &gt; app &gt; login &gt; login.component.html &gt; div.container &gt; div.col.col-md-4

```
1  <div class="container">
2      <div class="col col-md-4">
3          <input class="form-control" type="text" name="login" placeholder="Login ?" /> <br>
4          <input class="form-control" type="password" name="password" placeholder=" Mot de passe ?" />
5          <br> <button class="btn btn-primary" type="button">Se Connecter</button>
6      </div>
7  </div>
```

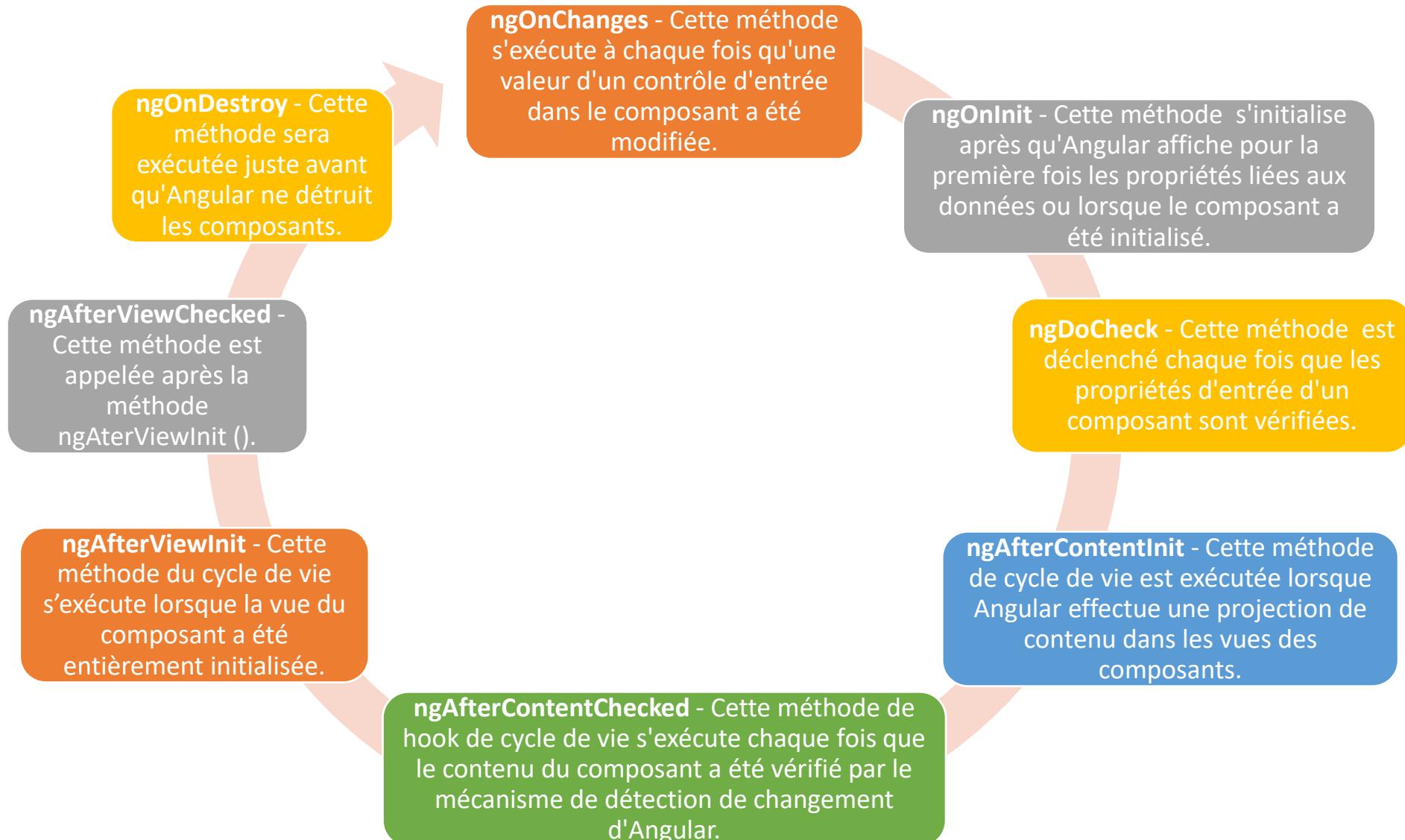


# Cycle de vie des Composants

- Chaque composant a un cycle de vie géré par Angular
- Il y a 8 phases dans la vie d'un composant angular. Lorsque nous initialisons le composant, il crée et présente ses propres composants. Il est alors conçu et crée ses propres héritiers. Il continuera également à vérifier quand les propriétés de liaison de données sont modifiées et mises à jour. Et enfin, lorsque le composant n'est plus utilisé, il est décimé et expulsé du DOM.



# Cycle de vie des Composants



# Travaux Pratiques n°2

<https://codelab.fun/angular/angular-cli>



# La liaison de données ou **Data Binding**

# Contenu Cours Angular

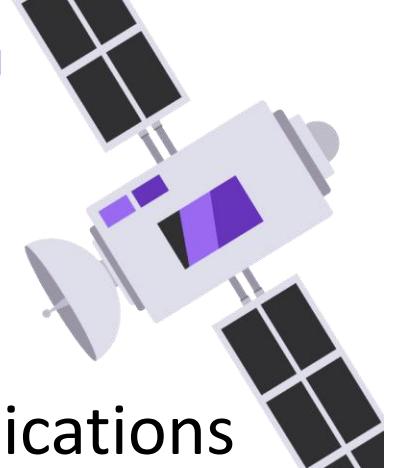
- Composants
- **Data Binding**
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



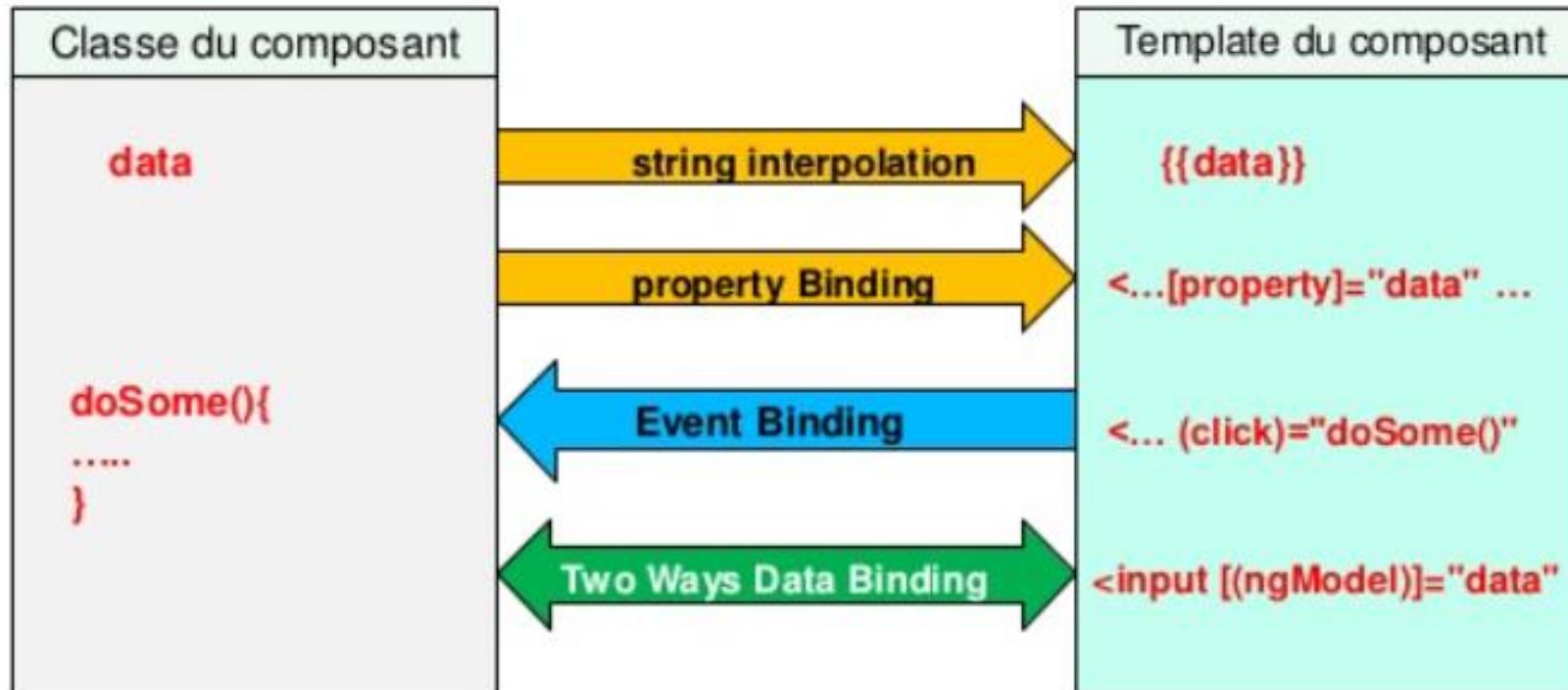


# Data-binding

- La plupart des frameworks qui permettent de créer des applications web utilisent cette notion de data-binding. C'est le cas pour Angular.
- Il s'agit d'un moyen de lier la partie vue à la partie logique.
- En d'autres termes, grâce à cela, les éléments de votre code HTML seront liés à votre contrôleur.

# Data-binding: Interaction entre la classe composante et son template

Permet de faire un lien entre les données de la classe du composant et son Template associé



`{{interpolation}}` est un raccourci de [one way binding]

# Data binding: Interpolation

- **Interpolation**: permet de récupérer la valeur d'un attribut déclarée dans le **.component.ts**
- Une expression Angular est comme une expression JavaScript entourée d'accolades - **`{{expression}}`**. Angular évalue l'expression spécifiée et lie les données de résultat au HTML.
- L'expression Angular peut contenir des littéraux, des opérateurs et des variables telles que l'expression JavaScript. Par exemple, une expression **`{{2+2}}`** produira le résultat 4 et sera liée à HTML.

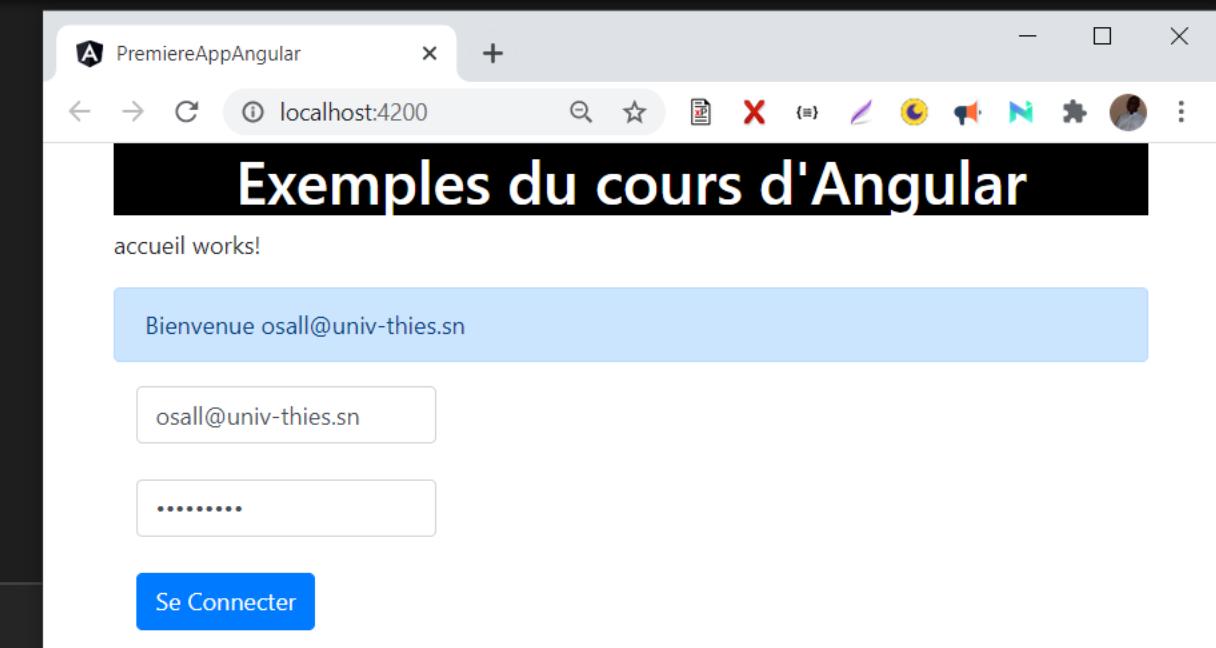


A login.component.ts X

```
src > app > login > A login.component.ts > LoginComponent > nomUtilisateur
2
3   @Component({
4     selector: 'app-login',
5     templateUrl: './login.component.html',
6     styleUrls: ['./login.component.css'],
7   })
8   export class LoginComponent implements OnInit {
9     nomUtilisateur = 'osall@univ-thies.sn';
10    motDePasse = 'passer123';
11    constructor() {}
12
13    ngOnInit(): void {}
14 }
```

S login.component.html X

```
src > app > login > S login.component.html > div.container > div.col.col-md-4
1   <div class="container">
2     <div class="alert alert-primary" role="alert">
3       Bienvenue {{nomUtilisateur}}
4     </div>
5     <div class="col col-md-4">
6       <input class="form-control" type="text" name="login" placeholder="Login ?" value="{{nomUtilisateur}}> <br>
7       <input class="form-control" type="password" name="password" placeholder=" Mot de passe ?" value="{{motDePasse}}>
8       <br> <button class="btn btn-primary" type="button">Se Connecter</button>
9     </div>
10  </div>
```



Binding du composant vers l'IU

# Data binding: Binding dans les deux sens avec [(ngModel)]="name "

- Un changement de valeur dans **.component.ts** sera aperçu dans **.component.html** et un changement dans **.component.html** sera reçu dans **.component.ts**
- Pour la liaison bidirectionnelle, il nous faut la propriété **ngModel**
- Pour pouvoir utiliser la propriété **ngModel**, il faut ajouter le module **FormsModule** dans **app.module.ts**
- **ngModel** est une directive qui lie la valeur des contrôles HTML (input, select, textarea) aux données de l'application.
- Nécessaire d'importer **FormsModule** dans **app.module.ts**  
**import { FormsModule } from "@angular/forms";**

EXPLORER 1

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR 47

- node\_modules
- src
  - app
    - accueil
    - login
      - login.component.css U
      - login.component.html 9+, U
      - login.component.spec.ts U
      - login.component.ts U
    - app-routing.module.ts A
    - app.component.css M
    - app.component.html M
    - app.component.spec.ts A
    - app.component.ts M
  - assets
  - environments
  - favicon.ico A
  - index.html A

> OUTLINE

> TIMELINE

> DEPENDENCIES

login.component.html ● app.module.ts X

src > app > app.module.ts > ...

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { AccueilComponent } from './accueil/accueil.component';
7 import { LoginComponent } from './login/login.component';
8 import { FormsModule } from '@angular/forms'; Calculating...
```

9

```
10 @NgModule({
11   declarations: [
12     AppComponent,
13     AccueilComponent,
14     LoginComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule,
19     FormsModule
20   ],
21   providers: [],
22   bootstrap: [AppComponent]
23 })
24 export class AppModule { }
```

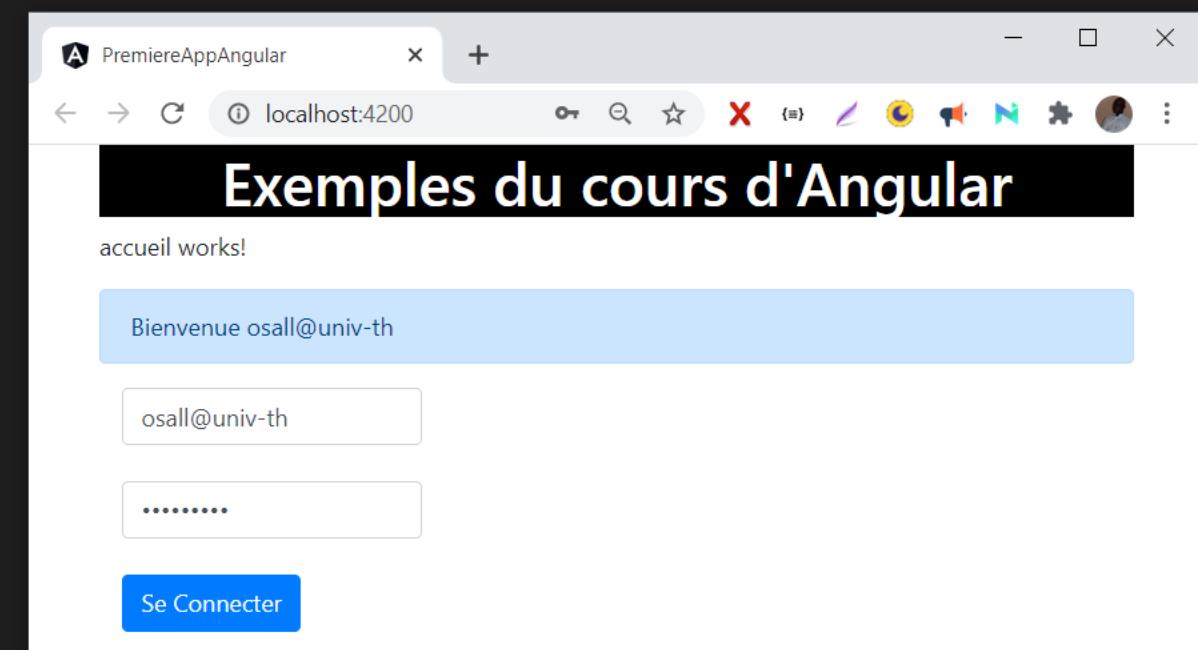
25



login.component.html X

src &gt; app &gt; login &gt; login.component.html &gt; div.container &gt; div.col.col-md-4 &gt; input.form-control

```
1  <div class="container">
2      <div class="alert alert-primary" role="alert">
3          Bienvenue {{nomUtilisateur}}
4      </div>
5      <div class="col col-md-4">
6          <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
7          <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse">
8          <br> <button class="btn btn-primary" type="button">Se Connecter</button>
9      </div>
10 </div>
```



# Data binding: **Binding par les propriétés**

- La liaison de propriété dans Angular Application est utilisée pour lier les valeurs des propriétés du composant ou du modèle à l'élément HTML.
- En fonction des valeurs, cela changera le comportement existant de l'élément HTML.
- La syntaxe pour utiliser la propriété est:

**[property] = "expression"**

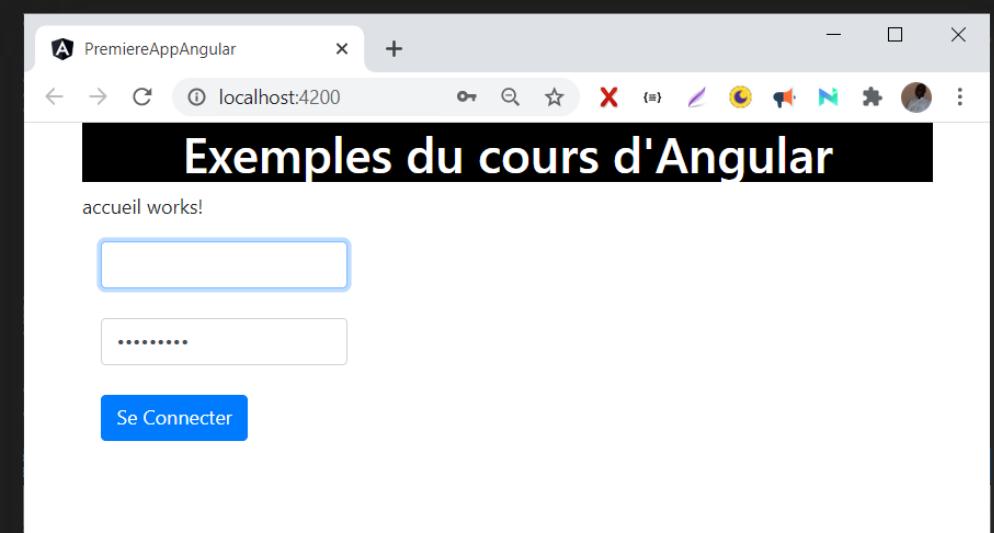
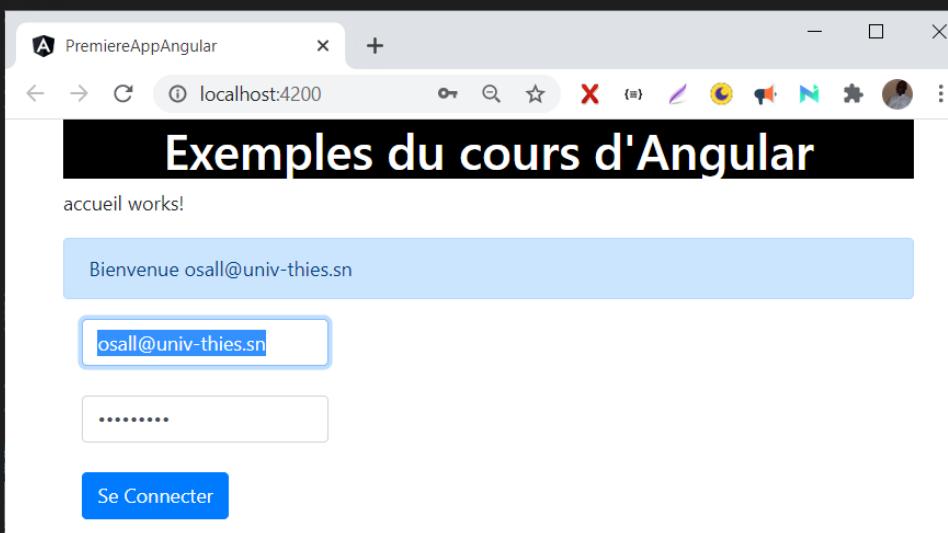


login.component.html X



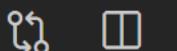
src > app > login > login.component.html > div.container > div.alert.alert-primary

```
1  <div class="container">
2      <div class="alert alert-primary" role="alert" [hidden]="nomUtilisateur.length<=0">
3          Bienvenue {{nomUtilisateur}}
4      </div>
5      <div class="col col-md-4">
6          <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
7          <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse">
8          <br> <button class="btn btn-primary" type="button">Se Connecter</button>
9      </div>
10     </div>
```





login.component.html X



src &gt; app &gt; login &gt; login.component.html &gt; div.container &gt; img.img-thumbnail

```
1  <div class="container">
2    <img class="img-thumbnail" [src]="imgUrl" alt="">
3    <div class="alert alert-primary" role="alert" [hidden]="nomUtilisateur.length<=0">
4      Bienvenue {{nomUtilisateur}}
5    </div>
6    <div class="col col-md-4">
7      <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
8      <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse">
9      <br> <button class="btn btn-primary" type="button">Se Connecter</button>
10   </div>
11 </div>
```



...

login.component.ts X

...

src &gt; app &gt; login &gt; login.component.ts &gt; LoginComponent &gt; imgUrl

```
8  export class LoginComponent implements OnInit {
9    nomUtilisateur = 'osall@univ-thies.sn';
10   motDePasse = 'passer123';
11   imgUrl="https://miro.medium.com/max/600/1*VKY-Ldkt-iHobItql7G_5w.png";
12   constructor() {}
13
14   ngOnInit(): void {
```





login.component.html X

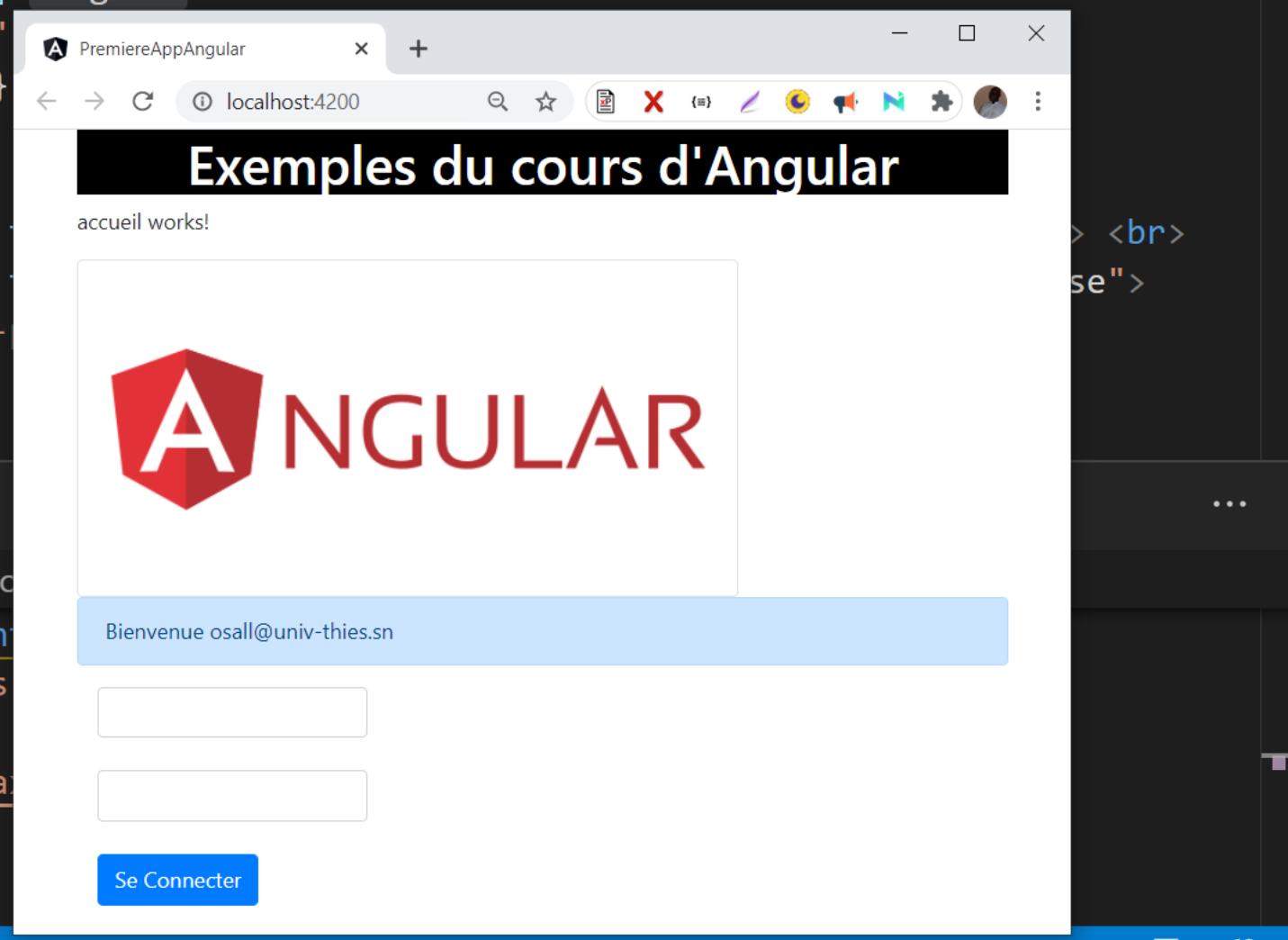
src &gt; app &gt; login &gt; login.component.html &gt; div.container &gt; img.img-thumbnail

```
1  <div class="container">
2      <img class="img-thumbnail" [src]=" imgUrl " alt="">
3      <div class="alert alert-primary">
4          Bienvenue {{nomUtilisateur}}
5      </div>
6      <div class="col col-md-4">
7          <input class="form-control" type="text" placeholder="Nom d'utilisateur"/>
8          <input class="form-control" type="password" placeholder="Mot de passe"/>
9          <br> <button class="btn btn-primary" type="submit">Se Connecter</button>
10     </div>
11 </div>
```

login.component.ts X

src &gt; app &gt; login &gt; login.component.ts &gt; LoginCo

```
8  export class LoginComponent implements OnInit {
9      nomUtilisateur = 'osall@univ-thies.sn';
10     motDePasse = 'passer123';
11     imgUrl="https://miro.medium.com/max/1400/1*H";
12     constructor() {}
13
14     ngOnInit(): void {
```



# Data binding: **Event binding**

Permet au .component.ts de récupérer des valeurs passées par le .component.html

- Lorsqu'un utilisateur interagit avec une application sous la forme d'un mouvement de clavier, d'un clic de bouton, d'un survol de la souris, d'une sélection dans une liste déroulante, d'une saisie dans une zone de texte, etc., il génère un événement. Ces événements doivent être gérés pour effectuer une sorte d'action. C'est là que la liaison d'événement entre en jeu et dans Angular Application, nous pouvons utiliser la liaison d'événement pour être averti lorsque ces événements se produisent.
- La syntaxe de la liaison à l'événement click d'un bouton par exemple comprend entre parenthèses sur le côté gauche du signe égal, nous avons l'événement cible, (cliquez dans ce cas) et sur le côté droit, nous avons l'instruction de modèle telle que les propriétés ou les méthodes des composants. Dans ce cas, c'est la méthode du composant, c'est-à-dire la méthode onClick (), qui sera appelée lorsque l'événement de clic sur le bouton se produit.

# Event binding avec les gestionnaires d'événements

- **(drag)**="maFonction()"
- **(drop)**="maFonction()"
- **(dragover)**="maFonction()"
- **(blur)**=" maFonction()"
- **(focus)**=" maFonction()"
- **(scroll)**="maFonction()"
- **(submit)**="maFonction()"
- **(click)**="maFonction()"
- **(dblclick)**="maFonction()"
- **(cut)**="maFonction()"
- **(copy)**="maFonction()"
- **(paste)**="maFonction()"
- **(keyup)**="maFonction()"
- **(keypress)**="maFonction()"
- **(keydown)**="maFonction()"
- **(mouseup)**="maFonction()"
- **(mousedown)**="maFonction()"
- **(mouseenter)**="maFonction()"

```
app.component.html
```

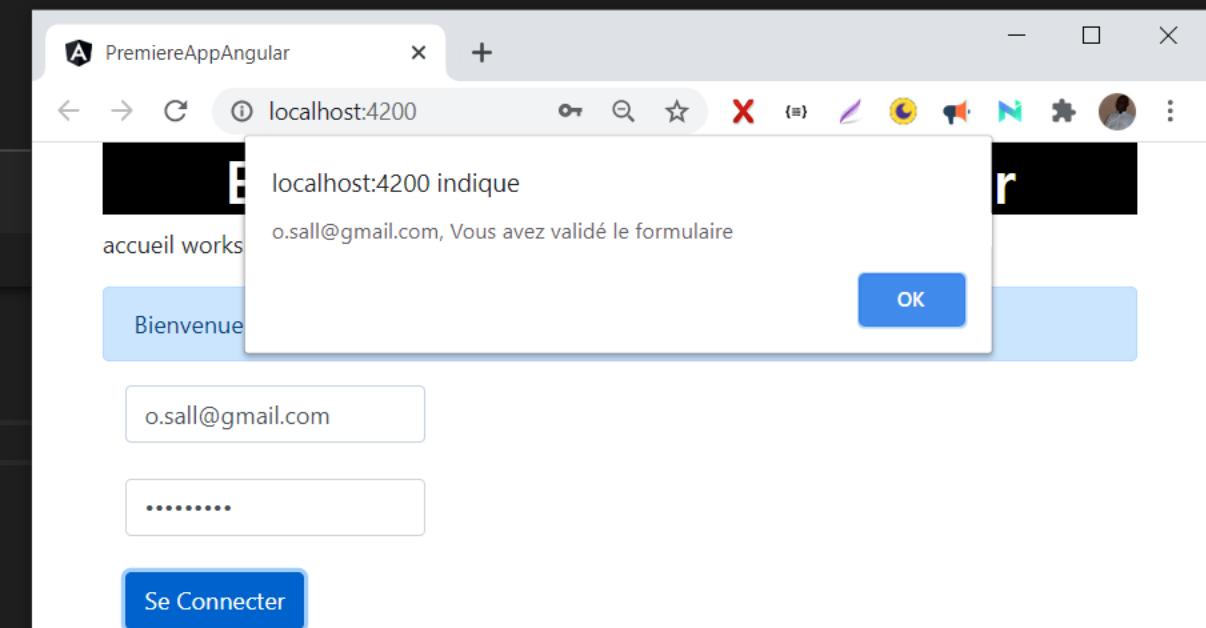
```
<button (click)="maFonction()>Click Here</button>
```

```
app.component.ts  
maFonction(event) {  
  console.log('Hi There!');  
}
```



login.component.html X

```
src > app > login > login.component.html > div.container
1  <div class="container">
2    <div class="alert alert-primary" role="alert" [hidden]="nomUtilisateur.length<=0">
3      Bienvenue {{nomUtilisateur}}
4    </div>
5    <div class="col col-md-4">
6      <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
7      <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse">
8      <br> <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>
9    </div>
10   </div>
```



# Travaux Pratiques n°3

<https://angular-presentation.firebaseioapp.com/angular/template/exercise>



# Routage et Navigation

# Contenu Cours Angular

- Composants
- Data Binding
- **Routage et Navigation**
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Routage et Navigation

- Une application Angular comporte plusieurs composants
- Chaque composant comporte sa propre vue
- Il faut donc un mécanisme pour naviguer entre les vues en fonction de l'action de l'utilisateur
- Créons un nouveau composant rendez-vous qui permettra une fois que l'utilisateur s'est logué d'afficher une page permettant de gérer des rendez-vous.

# Routage et Navigation

- Vous pouvez également naviguer dynamiquement à partir d'un composant ou d'un service. Un cas d'utilisation courant consiste à attendre que quelque chose d'async se termine, puis à envoyer l'utilisateur à une nouvelle page.
- Angular fournit un service de routeur que nous pouvons injecter dans les composants en le déclarant dans le constructeur.
- Il suffit d'injecter une instance de Router



EXPLORER

...



&gt; OPEN EDITORS

&lt; PREMIEREAPPANGULAR

src

cls

app

accueil

login

app-routing.module.ts M

app.component.css M

app.component.html M

app.component.spec.ts A

app.component.ts M

app.module.ts M

assets

environments

favicon.ico A

index.html

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

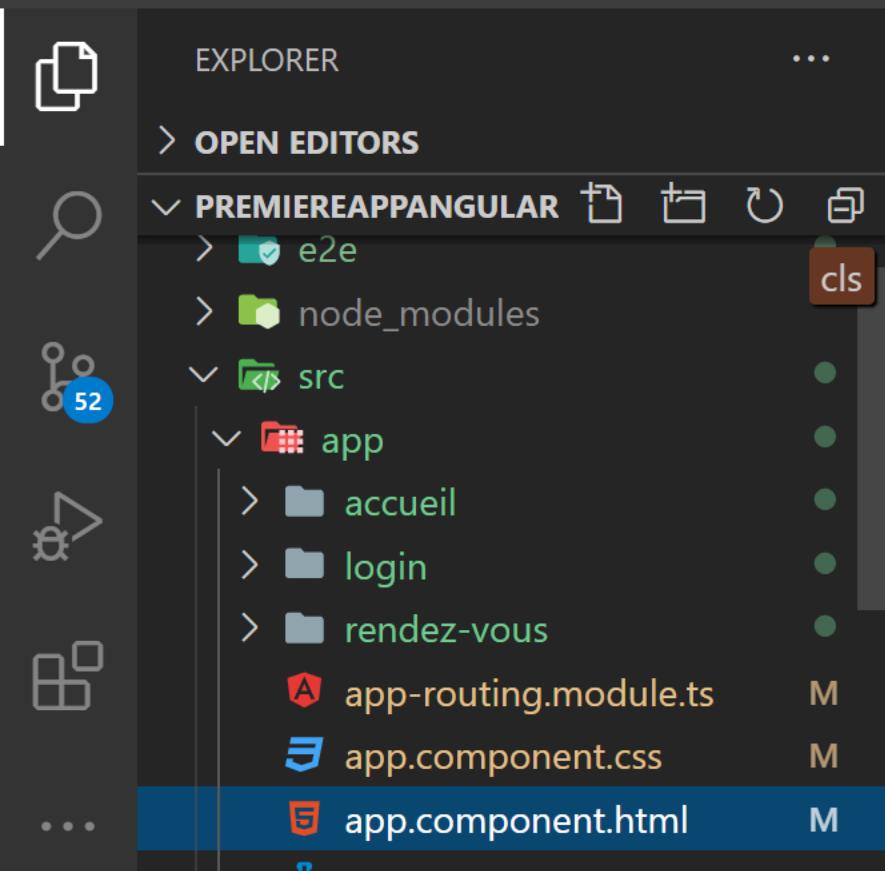
A app-routing.module.ts X

src &gt; app &gt; A app-routing.module.ts &gt; ...

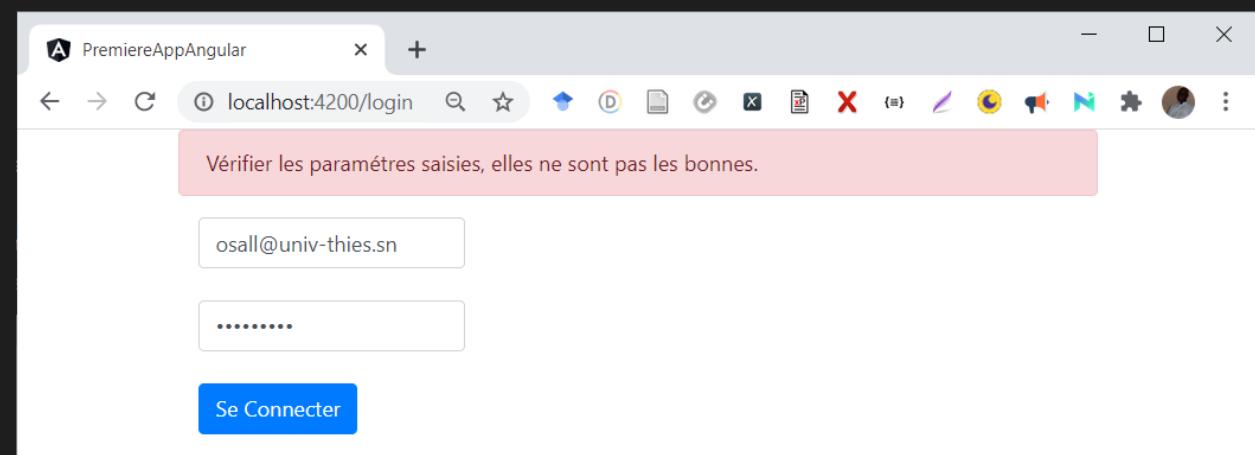
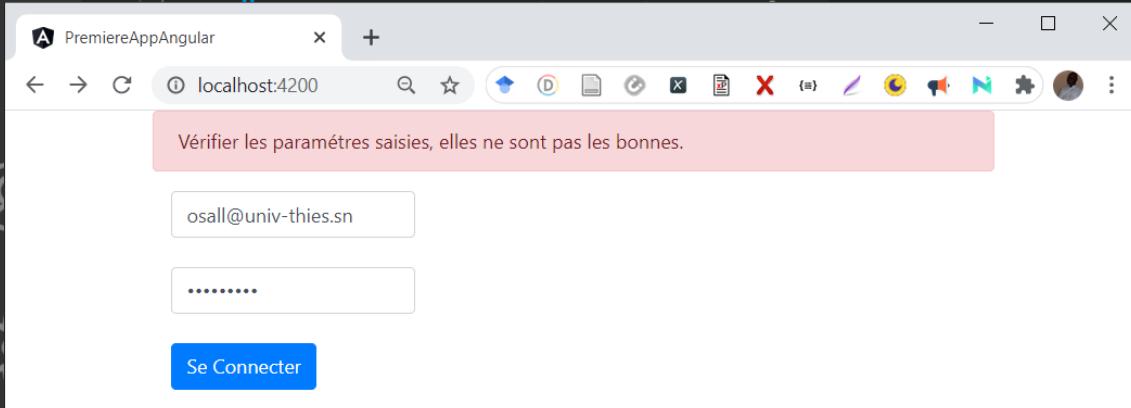
```
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { LoginComponent } from './login/login.component';
4 import { AccueilComponent } from './accueil/accueil.component';
5 const routes: Routes = [
6   { path: '', component: LoginComponent },
7   { path: 'accueil', component: AccueilComponent },
8   { path: 'login', component: LoginComponent }
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule],
14 })
15 export class AppRoutingModule {}
```

RouterModule a deux méthode statiques qui prennent en paramètre un tableau de Routes routes

- .forRoot(tableau) : pour le module principal
- .forChild(tableau) : pour les sous-modules inclus dans le module principal



```
src > app > app.component.html
1 |   <!-- <div class="container" style="text-align: center;"> -->
2 |     <h1>Welcome to Angular!</h1>
3 |   </div> -->
4 |
5 |   <!-- <app-accueil></app-accueil> -->
6 |
7 |   <!-- <app-login></app-login> -->
8 |
9 |   <router-outlet></router-outlet>
10|
11|   <!-- Les Composants seront affichées ici -->
```



The image shows a development environment with a code editor and a browser. The code editor on the left displays `login.component.ts` with code related to user authentication. The browser on the right shows two tabs: one for the login page and one for the home page.

**Code Editor (login.component.ts):**

```
src > app > login > A login.component.ts > LoginComponent > component
```

```
7 |   styleUrls: ['./login.component.css'],
8 | }
9 | export class LoginComponent implements OnInit {
10|   nomUtilisateur = 'osall@univ-thies.sn';
11|   motDePasse = '';
12|   parametresInvalides = true;
13|
14|   constructor(private router:Router) {}
15|
16|   ngOnInit(): void {}
17|
18|   valider() {
19|     if (this.nomUtilisateur === 'osall@univ-thies.sn' && this.motDePasse === 'passer123'){
20|       this.parametresInvalides = false;
21|       this.router.navigate(["accueil"]);
22|     }
23|     else this.parametresInvalides = true;
24|   }
25| }
26| }
```

**Browser (localhost:4200/login):**

Vérifier les paramètres saisis, elles ne sont pas les bonnes.

`osall@univ-thies.sn`

`Se Connecter`

**Browser (localhost:4200/accueil):**

accueil works!

At the bottom, the status bar shows:

master\*+ ⌂ ⊞ 0 △ 0 Compile Hero: Off Ln 14, Col 29 (6 selected) Spaces: 2 UTF-8 LF TS 4.0.3 Prettier: ✓ ⌂ ⌂

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- > src
  - > app
    - > accueil
      - accueil.component.html
      - accueil.component.css
      - accueil.component.ts
    - > login
  - > app-routing.module.ts
  - > app.component.css
  - > app.component.html

> OUTLINE

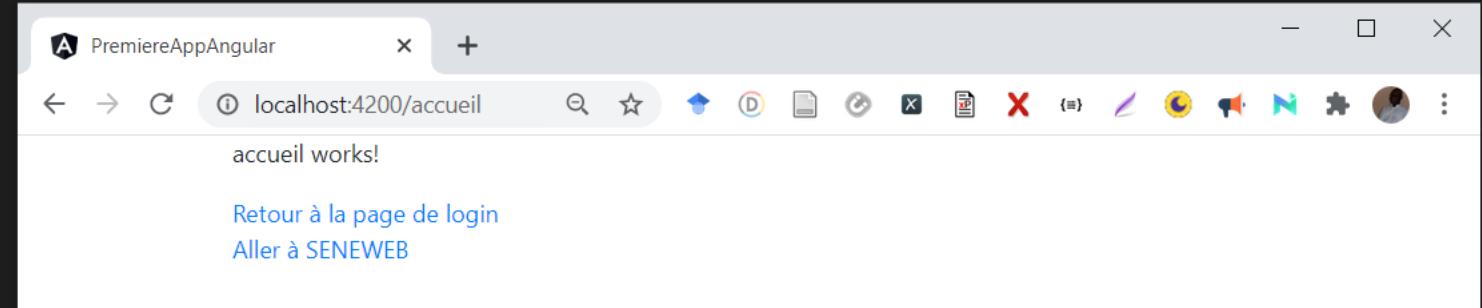
> TIMELINE

> DEPENDENCIES

## accueil.component.html X

src &gt; app &gt; accueil &gt; accueil.component.html &gt; ...

```
1 <div class="container">
2   <p>accueil works!</p>
3   <a href="/login">Retour à la page de login</a><br>
4   <a href="http://www.seneweb.com">Aller à SENEWEB</a> <br>
5 </div>
```



# Naviguer avec des Urls dynamiques

- Dans de nombreux cas, vous devez accéder à un chemin qui ressemble à **/items/:id** et vous ne connaissez pas la valeur de l'ID à l'avance. Configurons un itinéraire capable de gérer les paramètres dynamiques. Vous pouvez créer un segment d'URL dynamique en le préfixant avec :, technique utile pour le partage de données entre des composants ou des pages.
- Tout modèle de chaîne valide peut être utilisé pour le deuxième segment d'URL. Les deux rendront le même composant, mais passeront un ID différent à travers les paramètres.

```
const routes: Routes = [
  // Regular Route
  { path: 'items/:id', component: MyComponent },
]:
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANG... 48

- src > app > **app-routing.module.ts** M
- accueil
- login
- app-routing.module.ts M
- app.component.css M
- app.component.html M
- app.component.spec.ts A
- app.component.ts M
- app.module.ts M
- assets
- environments
- favicon.ico A
- index.html A
- main.ts TS

> OUTLINE

> TIMELINE

> DEPENDENCIES

**A** app-routing.module.ts Xsrc > app > **A** app-routing.module.ts > [?] routes > 🔗 path

```
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { LoginComponent } from './login/login.component';
4 import { AccueilComponent } from './accueil/accueil.component';
5 const routes: Routes = [
6   { path: '', component: LoginComponent },
7   { path: 'accueil/:nomUtilisateur', component: AccueilComponent },
8   { path: 'login', component: LoginComponent }
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule],
14 })
15 export class AppRoutingModule {}
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPA... 48

- > node\_modules
- src
  - app
    - accueil
  - login
    - login.component.css U
    - login.component.h... U
    - login.component.s... U
    - login.component.ts U

...

OUTLINE

TIMELINE

DEPENDENCIES

## login.component.ts X

src &gt; app &gt; login &gt; login.component.ts &gt; LoginComponent &gt; valider

```
7   styleUrls: ['./login.component.css'],
8 }
9 export class LoginComponent implements OnInit {
10   nomUtilisateur = 'osall@univ-thies.sn';
11   motDePasse = '';
12   parametresInvalides = true;
13
14 constructor(private router:Router) {}
15
16 ngOnInit(): void {}
17
18 valider() {
19   if (this.nomUtilisateur === 'osall@univ-thies.sn' &&this.motDePasse === '') {
20     this.parametresInvalides = false;
21     this.router.navigate(["accueil", this.nomUtilisateur]);
22   }
23   else this.parametresInvalides = true;
24 }
25
26 }
```

# Naviguer d'une page à l'autre avec transfert de données: **NavController** et **ActivatedRoute**

The image displays two screenshots of a web browser window titled "PremiereAppAngular".

**Screenshot 1 (Top): Login Page**

- The URL in the address bar is `localhost:4200/login`.
- A red error message box at the top states: "Vérifier les paramètres saisies, elles ne sont pas les bonnes."
- The email input field contains `osall@univ-thies.sn`.
- The password input field contains `*****`.
- A blue "Se Connecter" button is at the bottom.

**Screenshot 2 (Bottom): Success Page**

- The URL in the address bar is `localhost:4200/accueil/osall@univ-thies.sn`.
- The page header says "accueil works!".
- A green success message box displays "Bienvenue osall@univ-thies.sn".
- Navigation links at the bottom include "Retour à la page de login" and "Aller à SENEWEB".

Prof. Ousmane SALL, Univ. Thiès, SN

# Naviguer d'une page à l'autre avec transfert de données: **NavController** et **ActivatedRoute**

- **NavController** est la classe de base pour les composants de contrôleur de navigation tels que Nav et Tab.
- **NavController** est utilisé pour naviguer entre les pages d'une application. Au niveau de base, un contrôleur de navigation est un tableau de pages représentant un historique particulier (d'un onglet par exemple). Ce tableau peut être manipulé pour naviguer dans une application en poussant et en sautant des pages ou en les insérant et en les supprimant à des emplacements arbitraires de l'historique.

The screenshot shows a Visual Studio Code (VS Code) interface with the following components:

- Top Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Sidebar:** Includes icons for file operations (New, Open, Save, Find, Replace, etc.), a search bar, and a notifications section showing "48".
- Left Panel:** Shows the file structure: src > app > accueil > accueil.component.html.
- Code Editor:** The file "accueil.component.html" contains the following code:

```
<div class="container">
  <p>accueil works!</p>
  <div class="alert alert-success" role="alert">
    <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
```
- Output Panel:** Shows the browser output for "accueil.component.html" at "localhost:4200/accueil/osall@...". The page displays "accueil works!" and "Bienvenue osall@univ-thies.sn". It also includes links to "Retour à la page de login" and "Aller à SENEWEB".
- Bottom Panel:** Shows the file "accueil.component.ts" with the following code:

```
export class AccueilComponent implements OnInit {
  nomUtilisateur = '';
  constructor(private routeActive: ActivatedRoute) {}

  ngOnInit(): void {
    this.nomUtilisateur = this.routeActive.snapshot.paramMap.get("nomUtilisateur");
  }
}
```
- Bottom Status Bar:** master\*+, Compile Hero: Off, Ln 11, Col 49, Spaces: 2, UTF-8, LF, TS, 4.0.3, Prettier: ✓, and icons for file operations.

Autre possibilité:

[routerLink]=["'/path', routeParam"]>

```
<a [routerLink]="/path" [queryParams]={property:value}>  
    State Details  
</a>
```

# On peut créer un composant error et l'afficher en cas de chemin inexistant

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "PREMIEREAPPANGULAR".
  - src: app-routing.module.ts, error.component.ts
  - entete-page
  - error: error.component.css, error.component.html, error.component.spec.ts, error.component.ts
  - login
  - modeles
  - pied-page
  - pipes
  - services: authentication.service.spec.ts, authentication.service.ts, rest-api.service.spec.ts, rest-api.service.ts
  - app-routing.module.ts (selected)
  - app.component.css
  - app.component.html
- Terminal:** PowerShell command: `PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g c error` followed by component creation logs.
- Output:** Browser window titled "PremiereAppAngular" showing the URL `localhost:4200/testerreur`. The page displays the message "error works!".
- Status Bar:** master\*+, 0 △ 0, Compile Hero: Off, Ln 10, Col 42 (14 selected), Spaces: 2, UTF-8, LF, TS, 4.0.3, Prettier: ✓.

# Travaux Pratiques n°4

<https://codelab.fun/angular/router/intro>



# Les Directives structurelles et par attribut

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- **Directives**
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Les directives

- Les directives sont des instructions intégrées dans le DOM que vous utiliserez presque systématiquement quand vous créerez des applications Angular.
- Quand Angular lit votre template et rencontre une directive qu'il reconnaît, il suit les instructions correspondantes.
- Il existe deux types principaux de directive : les **directives structurelles**(modifient la structure du document: **\*ngIf** , **\*ngSwitch** , **\*ngFor**) et les **directives par attribut**(modifient le comportement d'un objet déjà existant: **ngModel**, **ngStyle**, **ngClass**).

# Les directives: `[(ngModel)]="name "`

- Directive qui lie la valeur des contrôles HTML (input, select, textarea) aux données de l'application.
- La directive ng-model peut également:
  - Fournir la validation de type pour les données d'application (numéro, email, requis).
  - Indiquer le statut des données d'application (non valide, sale, touché, erreur).
  - Fournir des classes CSS pour les éléments HTML.
  - Lier des éléments HTML aux formulaires HTML.
- Nécessaire d'importer **FormsModule** dans `app.module.ts`  
**`import { FormsModule } from "@angular/forms";`**

# Les directives: \*ngIf="expression"

Permet de supprimer ou recréer un élément suivant une expression passée en paramètre

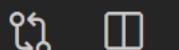
```
<div *ngIf="condition">  
    Content to render when condition is true.  
</div>
```

```
<div *ngIf="condition; else elseBlock">  
    Content to render when condition is true.  
</div>  
<ng-template #elseBlock>  
    Content to render when condition is false.  
</ng-template>
```



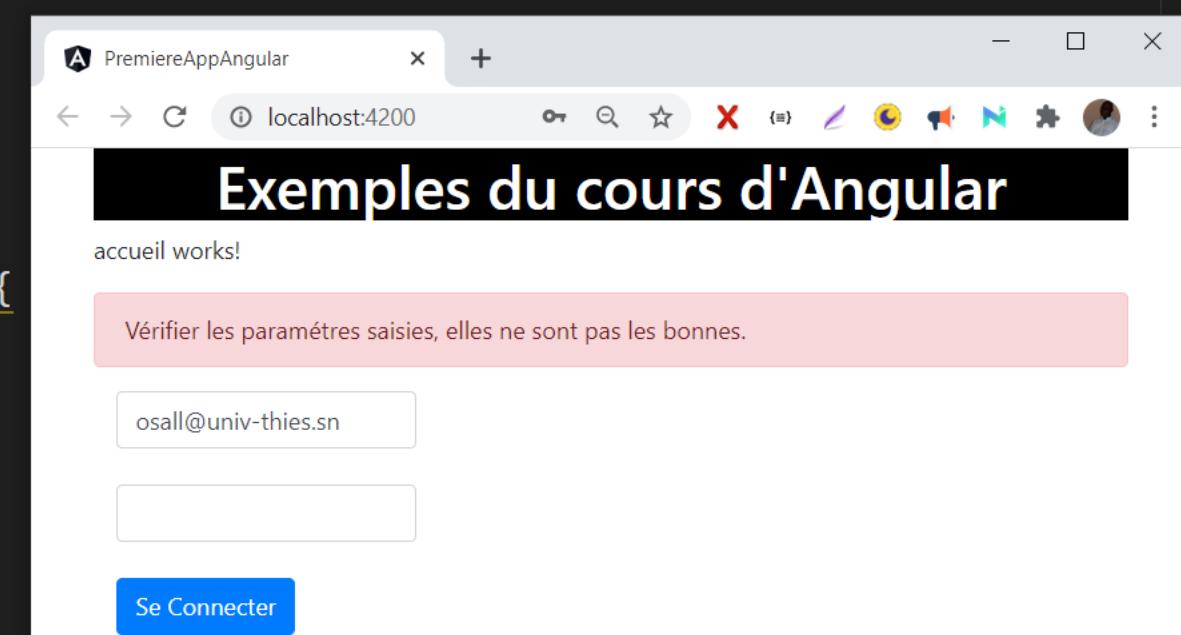
login.component.html

login.component.ts X



src &gt; app &gt; login &gt; login.component.ts &gt; LoginComponent &gt; parametresInvalides

```
3  @Component({
4    selector: 'app-login',
5    templateUrl: './login.component.html',
6    styleUrls: ['./login.component.css'],
7  })
8  export class LoginComponent implements OnInit {
9    nomUtilisateur = 'osall@univ-thies.sn';
10   motDePasse = '';
11   parametresInvalides=true;
12   constructor() {}
13
14   ngOnInit(): void {}
15
16   valider() {
17     if(this.nomUtilisateur==="osall@univ-thies.sn" && this.motDePasse==="passer123")
18       this.parametresInvalides=false;
19     else
20       this.parametresInvalides=true;
21   }
22 }
```



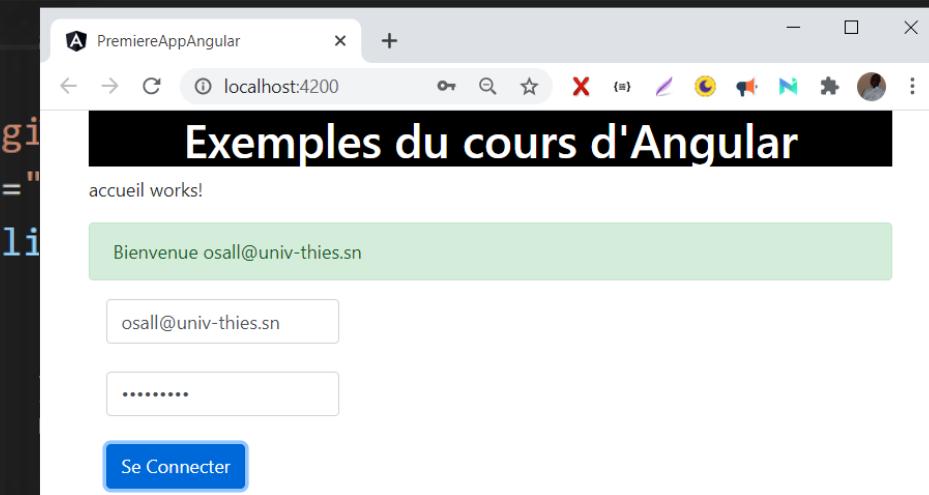
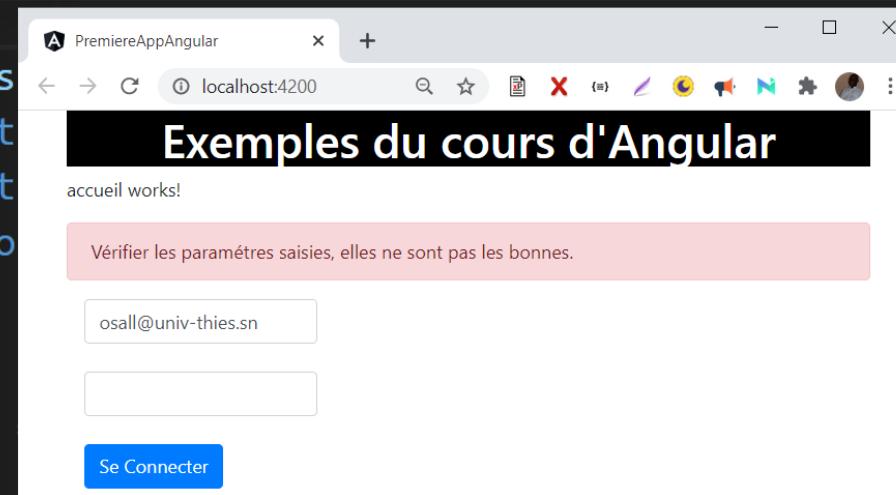


login.component.html X

login.component.ts

src &gt; app &gt; login &gt; login.component.html

```
1   <div class="container">
2     <ng-container *ngIf="!parametresInvalides; else elseTemplate">
3       <div class="alert alert-success" role="alert">
4         Bienvenue {{nomUtilisateur}}
5       </div>
6     </ng-container>
7     <ng-template #elseTemplate>
8       <div class="alert alert-danger" role="alert">
9         Vérifier les paramètres saisies, elles ne sont pas les bonnes.
10      </div>
11    </ng-template>
12
13   <div class="form-group">
14     <input type="text" placeholder="Email" value="osall@univ-thies.sn" />
15     <input type="password" placeholder="Mot de passe" value="*****" />
16     <button type="button" value="Se Connecter">Se Connecter</button>
17   </div>
18 </div>
```



# Les directives: **[ngSwitch]**, **\*ngSwitchCase** et **\*ngSwitchDefault**

**[ngSwitch]** ajoute ou supprime des templates lorsque l'expression correspond à l'expression du switch. Il affiche une section de HTML uniquement si elle correspond à une certaine valeur. Les expressions à rechercher sont fournies par la directive **\*ngSwitchCase**

- **[ngSwitch]** est utilisé comme liaison de propriété et **\*** est utilisé avec les directives **ngSwitchCase** et **ngSwitchDefault**
- La directive **\*ngSwitchCase** nous permet de définir une condition qui, lorsqu'elle est mise en correspondance avec l'expression, rendra l'élément
- Chaque vue qui correspond est rendue S'il n'y a pas de correspondance, "la directive **\*ngSwitchDefault** est rendue
- **\*ngSwitchCase** et **\*ngSwitchDefault** sont ceux qui créent le template



login.component.html X

login.component.ts



src &gt; app &gt; login &gt; login.component.html &gt; div.container

```
1   <div class="container">
2     <span [ngSwitch]="!parametresInvalides">
3       <div *ngSwitchCase="true" class="alert alert-success" role="alert">
4         Bienvenue {{nomUtilisateur}}
5       </div>
6       <div *ngSwitchCase="false" class="alert alert-danger" role="alert">
7         Vérifier les paramètres saisis, elles ne sont pas les bonnes.
8       </div>
9       <div *ngSwitchDefault>
10      </div>
11    </span>
13
14    <div class="col col-md-4">
15      <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
16      <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse"><br>
17      <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
18    </div>
19  </div>
```



47



...



# Les directives: \*ngFor="let item of items"

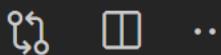
- Le \* est un raccourci pour utiliser la nouvelle syntaxe de modèle angular avec la balise de modèle. Cela s'appelle aussi une directive structurelle.
- Il est utile de savoir que \* n'est qu'un raccourci pour définir explicitement les liaisons de données sur une balise de modèle. La balise de modèle empêche le navigateur de lire ou d'exécuter le code qu'il contient.

## A noter:

- \*ngFor nous permet d'itérer sur un tableau, mais non pas sur un objet
- Il est possible d'itérer sur un objet après avoir défini un pipe qui convertit virtuellement un objet en tableau



accueil.component.ts X



src &gt; app &gt; accueil &gt; accueil.component.ts &gt; RendezVous &gt; (get) date

```
20  export class RendezVous {
21      constructor(private _id: number,  private _description: string,  private _lieu: string,
22                  private _date: Date) {}
23
24      public get id(): number {
25          return this._id;
26      }
27
28      public get description(): string {
29          return this._description;
30      }
31
32      public get lieu(): string {
33          return this._lieu;
34      }
35
36      public get date(): Date {
37          return this._date;
38      }
39  }
```



A accueil.component.ts X

```
src > app > accueil > A accueil.component.ts > 📄 AccueilComponent > 🔑 rendezVous
    7   |   styleUrls: ['./accueil.component.css'],
    8   })
    9   export class AccueilComponent implements OnInit {
10     nomUtilisateur = '';
11     rendezVous = [
12       new RendezVous(1, "Cours Angular", "UT-THIES", new Date('20/10/2020')),
13       new RendezVous(2, 'Cours Flutter', "UGB-Saint-Louis", new Date('10/11/2010')),
14       new RendezVous( 3, 'Cours Spring Boot', "UADB-Bambey", new Date('12/09/2021'))
15     ];
16
17
18   constructor(private routeActive: ActivatedRoute) {}
19
20   ngOnInit(): void {
21     this.nomUtilisateur = this.routeActive.snapshot.paramMap.get(
22       'nomUtilisateur'
23     );
24   }
25 }
26
27 export class RendezVous {
```



accueil.component.html X

accueil.component.ts



```
src > app > accueil > accueil.component.html > div.container > table.table.table-light > table.table.table-light > thead.thead-light > tr  
1   <div class="container">  
2     <div class="alert alert-success" role="alert">  
3       <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>  
4     </div>  
5     <table class="table table-light">  
6       <thead class="thead-light">  
7         <tr>  
8           <th>Id</th>  
9           <th>Description</th>  
10          <th>Lieu</th>  
11          <th>Date</th>  
12        </tr>  
13      </thead>  
14      <tbody>  
15        <tr>  
16          <td>{{rendezVous[0].id}}</td>  
17          <td>{{rendezVous[0].description}}</td>  
18          <td>{{rendezVous[0].lieu}}</td>  
19          <td>{{rendezVous[0].date}}</td>  
20        </tr>  
21      </tbody>  
22    </table>  
23  </div>
```



48



A PremiereAppAngular

localhost:4200/accueil/osall@...

Id	Description	Lieu	Date
1	Cours Angular	UT-THIES	Invalid Date

# Les directives: \*ngFor="let item of items"

\*ngFor demande à Angular de créer une ligne par rendez-vous

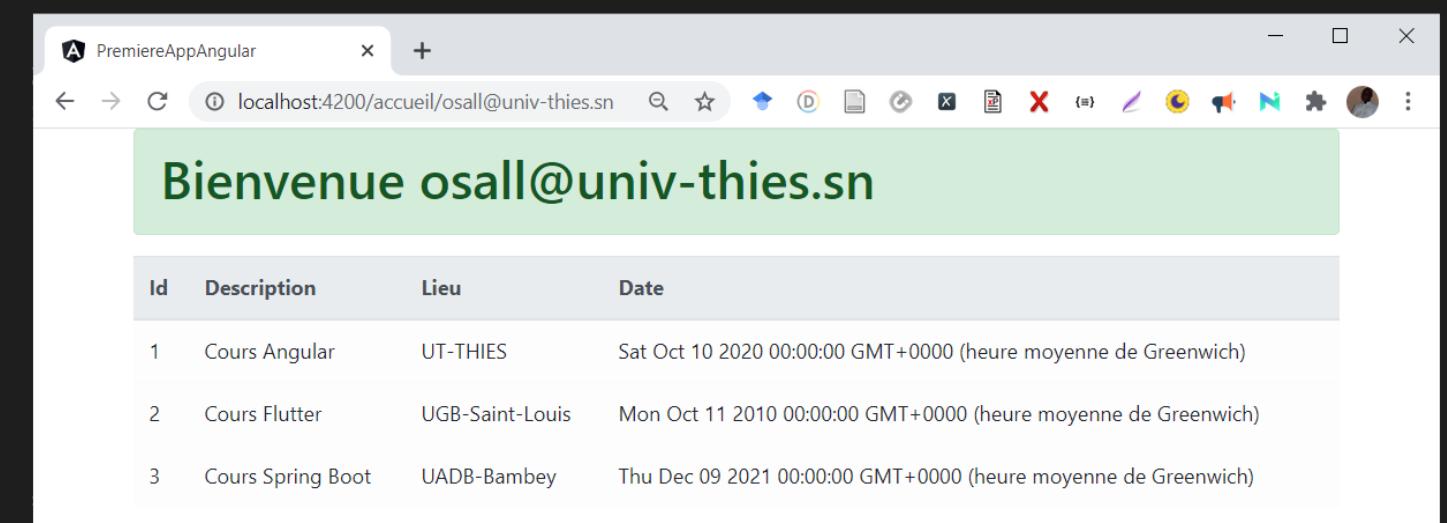
"Let rv" crée une variable locale qui sera disponible pour le template

```
<tr *ngFor="let rv of rendezVous">  
  ...  
</tr>
```

"of rendezVous" signifie que l'itération se fera sur la collection rendezVous

accueil.component.html X

```
src > app > accueil > accueil.component.html
1   <div class="container">
2     <div class="alert alert-success" role="alert">
3       <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
4     </div>
5     <table class="table table-light">
6       <thead class="thead-light">
7         <tr>
8           <th>Id</th>
9           <th>Description</th>
10          <th>Lieu</th>
11          <th>Date</th>
12        </tr>
13      </thead>
14      <tbody>
15
16        <tr *ngFor="let rv of rendezVous">
17          <td>{{rv.id}}</td>
18          <td>{{rv.description}}</td>
19          <td>{{rv.lieu}}</td>
20          <td>{{rv.date}}</td>
21        </tr>
22      </tbody>
23    </table>
24  </div>
```



# Les directives:

[ngClass] = "{'class' : true}"

- [ngClass] = "{'classname' : condition}"  
**Ou bien**
- [ngClass] = "(expr=='val')?'my-class1':'my-class2'"  
**Ou bien**
- [ngClass] = "getMaClasseCSS()"

# [Style.ObjectProperties] = "test? 'value1':'value2'"

The screenshot shows a web browser window displaying the w3schools.com website. The URL in the address bar is [https://www.w3schools.com/jsref/dom\\_obj\\_style.asp](https://www.w3schools.com/jsref/dom_obj_style.asp). The page title is "HTML DOM Style Object". On the left, there is a sidebar with a green header "DOM Style" containing a list of CSS properties: alignContent, alignItems, alignSelf, animation, animationDelay, animationDirection, animationDuration, animationFillMode, animationIterationCount, animationName, animationTimingFunction, animationPlayState, background, backgroundAttachment, backgroundColor, backgroundImage, backgroundPosition, backgroundRepeat, backgroundClip, backgroundOrigin, backgroundSize, backfaceVisibility, border, borderBottom, borderBottomColor. The main content area has a heading "Style object" with the subtext "The Style object represents an individual style statement." Below it is a section titled "Style Object Properties" with the subtext "The "CSS" column indicates in which CSS version the property is defined (CSS1, CSS2, or CSS3)." A table lists the properties and their descriptions:

Property	Description	CSS
<a href="#">alignContent</a>	Sets or returns the alignment between the lines inside a flexible container when the items do not use all available space	3
<a href="#">alignItems</a>	Sets or returns the alignment for items inside a flexible container	3
	Sets or returns the alignment for selected items inside a flexible container	3

At the bottom of the page, there is a "COLOR PICKER" button and a footer note "Prof. En attente de web.hb.ad.cpe.dotomi.com...".



# Les Services et Injection de dépendances

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- **Services et Injection de dépendances**
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



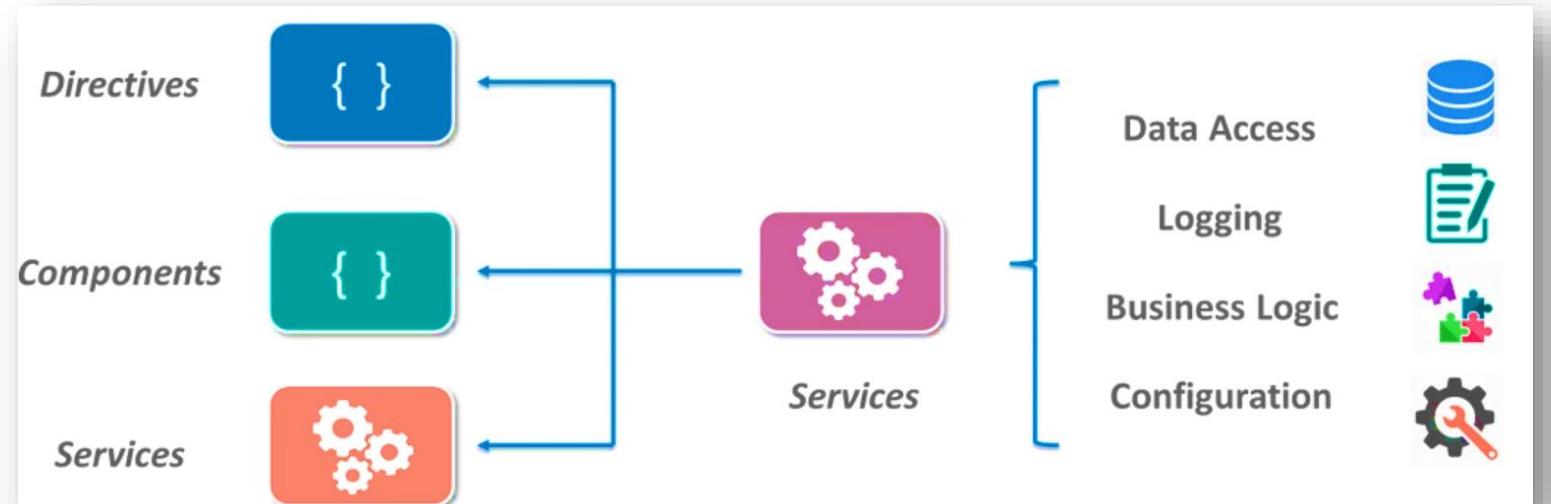
# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation

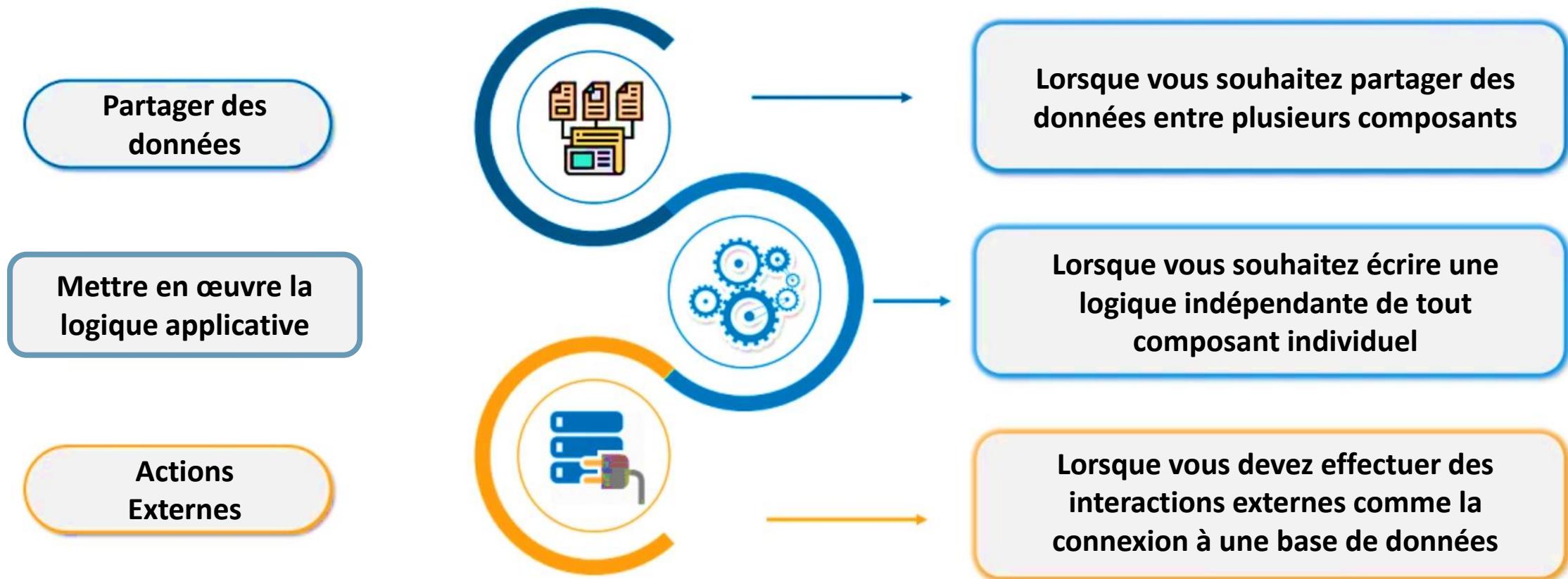


# Qu'est-ce qu'un service Angular

- Un service dans Angular est utilisé lorsque vous devez réutiliser des données ou une logique sur plusieurs composants. Il englobe toute valeur, fonction ou fonctionnalité dont une application a besoin. C'est un morceau de code qui agit comme un référentiel central ou une unité commerciale centrale où vous pouvez stocker et centraliser le code

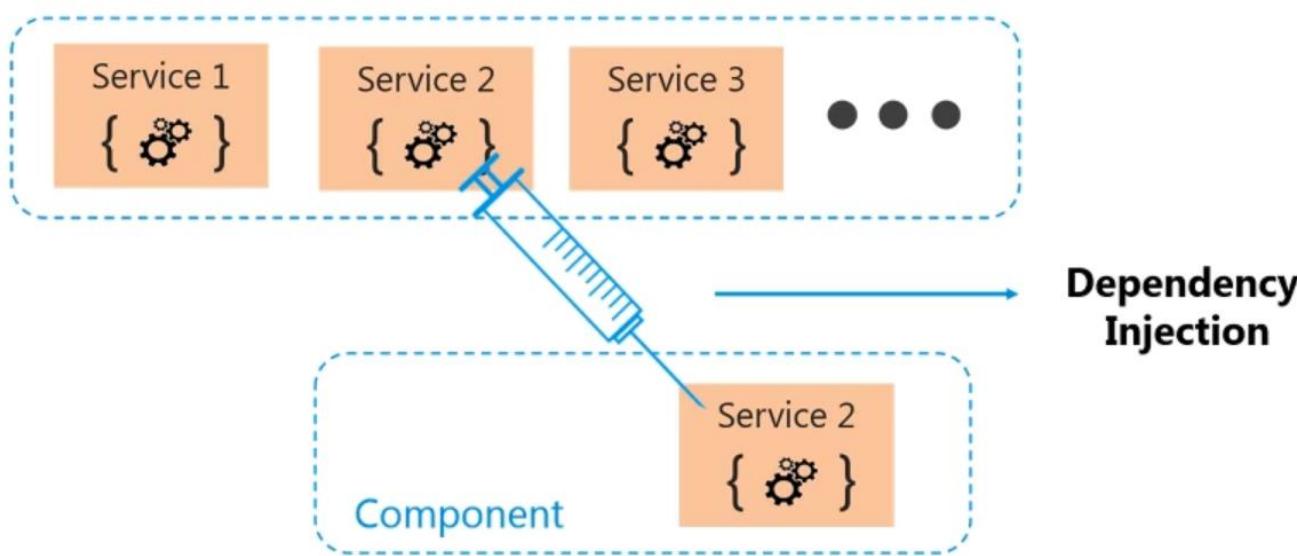


# Quand utiliser un Service ?



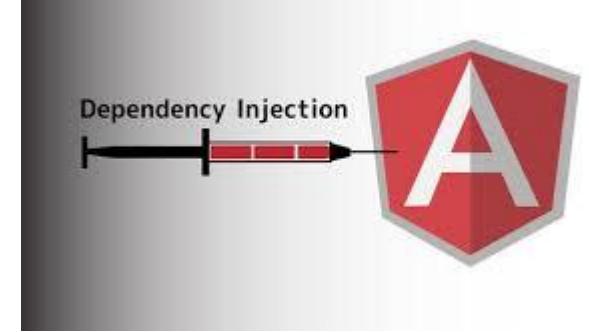
# Qu'est-ce que l'injection de dépendance ?

- L'injection de dépendances fait référence à un objet fournissant les dépendances d'un autre objet. Une classe reçoit ses dépendances de sources externes plutôt que de les créer elle-même

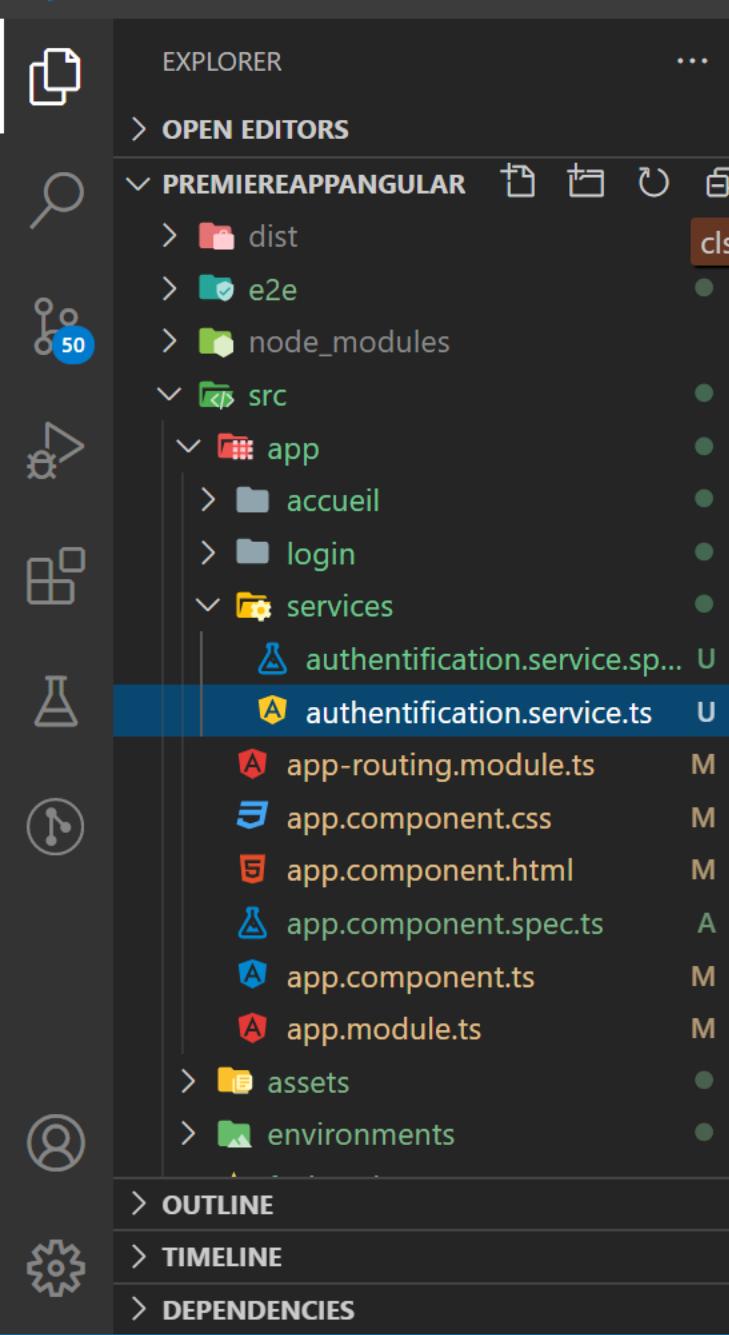


- Vous voulez qu'un service soit injecté dans un composant vous pouvez utiliser l'injection de dépendances
- Il créera une nouvelle instance d'une classe de service avec ses dépendances requises
- Vous pouvez ensuite utiliser la logique et les méthodes qui ont été définies dans cette classe de service

# Les Services



- Une classe TypeScript décorée par **@Injectable**
- Un singleton
- Intermédiaire avec la partie Back-end
- Peut être utilisé sans qu'il soit instancié : en utilisant l'injection de dépendance
- Doit être déclaré dans la section **providers** du module
- Un service peut faire appel à un ou plusieurs autres services
- Pour créer un service
  - **ng generate service nomService** ou aussi
  - **ng g s nomService**



The Explorer sidebar shows the project structure:

- > OPEN EDITORS
- ✓ PREMIEREAPPANGULAR    
- > dist
- > e2e
- > node\_modules
- ✓ src 
  - < app 
    - > accueil
    - > login
    - ✓ services 
      - authentification.service.sp... 
      - authentification.service.ts 
      - app-routing.module.ts 
      - app.component.css 
      - app.component.html 
      - app.component.spec.ts 
      - app.component.ts 
      - app.module.ts 
    - > assets
    - > environments
- > OUTLINE
- > TIMELINE
- > DEPENDENCIES

### authentification.service.ts

src > app > services > authentification.service.ts > ...

```
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class AuthentificationService {
7
8   constructor() { }
9 }
```

10



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

3: powershell      

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g s services/authentification
CREATE src/app/services/authentification.service.spec.ts (407 bytes)
CREATE src/app/services/authentification.service.ts (145 bytes)
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```



EXPLORER

app.module.ts



&gt; OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR



&gt; node\_modules



&gt; src



&gt; app



&gt; accueil



&gt; login



&gt; services



app-routing.module.ts M

app.component.css M

app.component.html M

app.component.spec.ts A

app.component.ts M

app.module.ts M

&gt; assets



&gt; environments



favicon.ico A

index.html A

main.ts A

polyfills.ts A

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

src &gt; app &gt; app.module.ts &gt; ...

```
8 import { FormsModule } from '@angular/forms';
9 import { AuthentificationService } from './services/authentification.service';
10
11 @NgModule({
12   declarations: [AppComponent, AccueilComponent, LoginComponent],
13   imports: [BrowserModule, AppRoutingModule, FormsModule],
14   providers: [AuthentificationService],
15   bootstrap: [AppComponent],
16 })
17 export class AppModule {}
```



## A authentication.service.ts X



src &gt; app &gt; services &gt; A authentication.service.ts &gt; ...

```
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class AuthentificationService {
7
8   constructor() {}
9
10  seLoguer(nomUtilisateur:string, motDePasse:string) {
11    if (nomUtilisateur === 'osall@univ-thies.sn' &&motDePasse === 'passer123'){
12      return true;
13    }
14    return false;
15  }
16}
```

Le décorateur **@Injectable** transforme une simple classe TypeScript en service



login.component.ts X



src &gt; app &gt; login &gt; login.component.ts &gt; LoginComponent &gt; constructor

```
8   styleUrls: ['./login.component.css'],
9 }
10 export class LoginComponent implements OnInit {
11   nomUtilisateur = 'osall@univ-thies.sn';
12   motDePasse = 'passer123';
13   parametresInvalides = true;
14
15   constructor(private router:Router, private authenticationService:AuthentificationService) {}
16
17   ngOnInit(): void {}
18
19   valider() {
20     if(this.authenticationService.seLoguer(this.nomUtilisateur, this.motDePasse)){
21       this.router.navigate(["accueil", this.nomUtilisateur]);
22       this.parametresInvalides = false;
23     }else{
24       this.parametresInvalides = true;
25     }
26   }
27 }
```

# Travaux Pratiques n°5

[https://codelab.fun/  
angular/dependency  
-injection/intro](https://codelab.fun/angular/dependency-injection/intro)



# Les Pipes

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- **Pipes**
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Un pipe

- Rôle de modifier une donnée dans le **.component.html**
- Une classe décorée par le décorateur **@pipe**
- Implémente la méthode **transform(value: any, args?: any)** de l'interface **PipeTransform**
- Peut prendre des paramètres
- **Exemple**

```
 {{ "bonjour" | uppercase }}  
 <!-- BONJOUR -->  
  
 Certains pipes peuvent prendre des paramètres  
 {{ maDate | date:'d MMM y' }}  
 <!-- affiche 19 Avril 2020 -->  
  
 Il est possible de chaîner les pipes  
 {{ maDate | date:'d MMM y' | uppercase }}  
 <!-- affiche 19 AVR 2020 -->
```

```
import { Pipe, PipeTransform } from '@angular/core';  
  
@Pipe({  
  name: 'pipeName'  
})  
  
export class pipeClass implements PipeTransform {  
  transform(parameters): returntype {}  
}
```

# Deux catégories de pipe

- pipes prédéfinis comme uppercase, lowercase, decimal, currency, percent
- pipes personnalisés

# Pipes prédéfinis

UpperCasePipe  
DecimalPipe

TitleCasePipe

CurrencyPipe

SlicePipe

I18nSelectPipe

PercentPipe

AsyncPipe

JsonPipe

I18nPluralPipe

LowerCasePipe



accueil.component.html X



src &gt; app &gt; accueil &gt; accueil.component.html &gt; div.container &gt; table.table-table-light &gt; tbody &gt; tr &gt; td

```
4      </div>
5      <table class="table table-light">
6          <thead class="thead-light">
7              <tr>
8                  <th>Id</th>
9                  <th>Description</th>
10                 <th>Lieu</th>
11                 <th>Date</th>
12             </tr>
13         </thead>
14         <tbody>
15             <tr *ngFor="let rv of rendezVous">
16                 <td>{{rv.id}}</td>
17                 <td>{{rv.description}}</td>
18                 <td>{{rv.lieu | uppercase }}</td>
19                 <td>{{rv.date | date:'dd MMMM y à hh:mm:ss' }}</td>
20             </tr>
21         </tbody>
22     </table>
23 </div>
```

ID	Description	Lieu	Date
1	Cours Angular	UT-THIES	10 October 2020 à 12:00:00
2	Cours Flutter	UGB-SAINT-LOUIS	11 October 2010 à 12:00:00
3	Cours Spring Boot	UADB-BAMBEY	09 December 2021 à 12:00:00

EXPLORER ...

> OPEN EDITORS

PREMIEREAPPANGU... 📁 ⚡ ⚡ ⚡

- > e2e
- > node\_modules
- ✓ src
  - ✓ app
    - > accueil
    - > login
  - ✓ pipes
    - get-string-avant-tiret.... U
    - A get-string-avant-tiret.... U
  - > services
  - A app-routing.module.ts M
  - E app.component.css M
  - S app.component.html M
  - A app.component.spec.ts A
  - A app.component.ts M
  - A app.module.ts M
- > assets
- > environments

> OUTLINE

> TIMELINE

> DEPENDENCIES

get-string-avant-tiret.pipe.ts X

src > app > pipes > A get-string-avant-tiret.pipe.ts > ...

```
1 import { Pipe, PipeTransform } from '@angular/core';
2
3 @Pipe({
4   name: 'getStringAvantTiret'
5 })
6 export class GetStringAvantTiretPipe implements PipeTransform {
7
8   transform(value: unknown, ...args: unknown[]): unknown {    'value' is declared but its v
9     return null;
10 }
11
12 }
```

Pour créer un pipe personnalisé  
ng generate pipe nomPipe

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: node

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g pipe pipes/getStringAvantTiret
CREATE src/app/pipes/get-string-avant-tiret.pipe.spec.ts (242 bytes)
CREATE src/app/pipes/get-string-avant-tiret.pipe.ts (243 bytes)
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> [ ]
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- > e2e
- > node\_modules
- ✓ src
  - ✓ app
    - > accueil
    - > login
  - ✓ pipes
    - get-string-avant-tiret.pipe.ts U
    - get-string-avant-tiret.pipe.spec.ts U
  - > services
    - app-routing.module.ts M
    - app.component.css M
    - app.component.html M
    - app.component.spec.ts A
    - app.component.ts M
    - app.module.ts M
  - > assets
  - > environments
- > OUTLINE
- > TIMELINE
- > DEPENDENCIES

get-string-avant-tiret.pipe.ts app.module.ts

```
src > app > app.module.ts > ...
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { AccueilComponent } from './accueil/accueil.component';
7  import { LoginComponent } from './login/login.component';
8  import { FormsModule } from '@angular/forms';
9  import { AuthenticationService } from './services/authentication.service';
10 import { GetStringAvantTiretPipe } from './pipes/get-string-avant-tiret.pipe';

12 @NgModule({
13   declarations: [AppComponent, AccueilComponent, LoginComponent, GetStringAvantTiretPipe],
14   imports: [BrowserModule, AppRoutingModule, FormsModule],
15   providers: [AuthenticationService],
16   bootstrap: [AppComponent],
17 })
18 export class AppModule {}
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- > e2e
- > node\_modules
- src
  - app
    - accueil
    - login
  - pipes
    - get-string-avant-tiret.pipe.spec.ts
    - get-string-avant-tiret.pipe.ts
  - services
  - app-routing.module.ts
  - app.component.css
  - app.component.html
  - app.component.spec.ts
  - app.component.ts
  - app.module.ts
- assets
- environments

> OUTLINE

> TIMELINE

> DEPENDENCIES

get-string-avant-tiret.pipe.ts X

src > app > pipes > A get-string-avant-tiret.pipe.ts > GetStringAvantTiretPipe > transform

```
1 import { Pipe, PipeTransform } from '@angular/core';
2
3 @Pipe({
4   name: 'getStringAvantTiret',
5 })
6 export class GetStringAvantTiretPipe implements PipeTransform {
7   transform(value: unknown, ...args: unknown[]): unknown {
8     const str = value + args.join(' ');
9     return str.substring(0, str.indexOf('-'));
10  }
11 }
12 }
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- > e2e
- > node\_modules
- src
  - app
    - accueil
      - accueil.component.css
      - accueil.component.html
      - accueil.component.spec.ts
      - accueil.component.ts
    - login
    - pipes
    - services
    - app-routing.module.ts
    - app.component.css
    - app.component.html
    - app.component.spec.ts
    - app.component.ts
    - app.module.ts

> OUTLINE

> TIMELINE

> DEPENDENCIES

accueil.component.html X

src > app > accueil > accueil.component.html

```
4   </div>
5   <table class="table table-light">
6     <thead class="thead-light">
7       <tr>
8         <th>Id</th>
9         <th>Description</th>
10        <th>Lieu</th>
11        <th>Date</th>
12      </tr>
13    </thead>
14    <tbody>
15      <tr *ngFor="let rv of rendezVous">
16        <td>{{rv.id}}</td>
17        <td>{{rv.description}}</td>
18        <td>{{rv.lieu | uppercase | getStringAvantTiret}}</td>
19        <td>{{rv.date | date:'dd MMMM y à hh:mm:ss' }}</td>
20      </tr>
21    </tbody>
22  </table>
23 </div>
```

Bienvenue osall@univ-thies.sn

ID	Description	Lieu	Date
1	Cours Angular	UT	10 October 2020 à 12:00:00
2	Cours Flutter	UGB	11 October 2010 à 12:00:00
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00

master\*+ 0 ▲ 0

Compile Hero: Off Ln 18, Col 64 (19 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter 1.22.0 Premiere. v



# Gestion des Sessions avec **localStorage** et **sessionStorage**

# Contenu Cours Angular

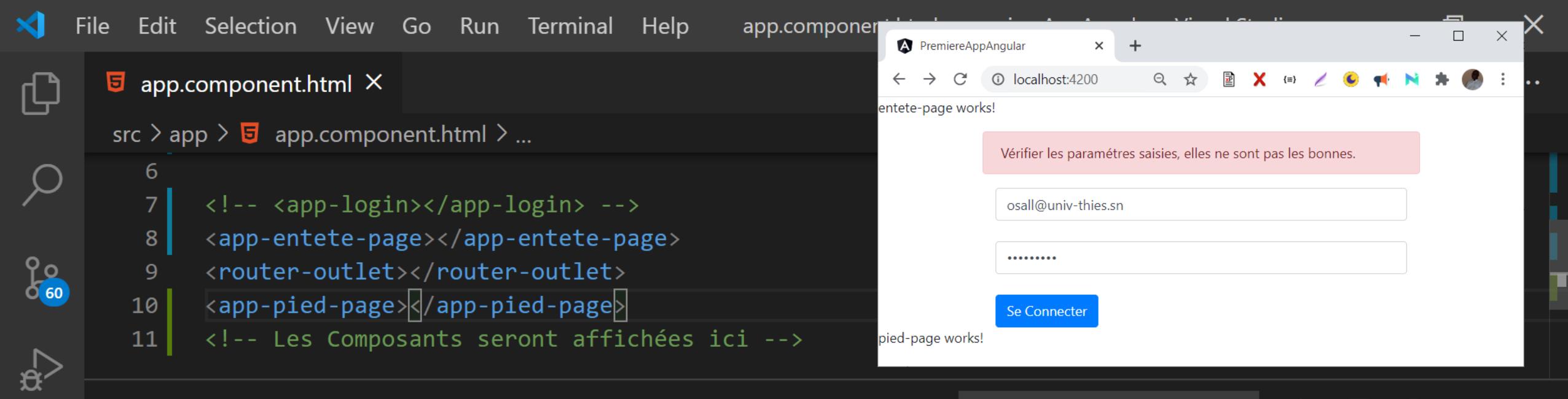
- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- **Gestion des Sessions avec localStorage et sessionStorage**



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation





PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: node

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g c entetePage
CREATE src/app/entete-page/entete-page.component.html (26 bytes)
CREATE src/app/entete-page/entete-page.component.spec.ts (655 bytes)
CREATE src/app/entete-page/entete-page.component.ts (294 bytes)
CREATE src/app/entete-page/entete-page.component.css (0 bytes)
UPDATE src/app/app.module.ts (890 bytes)

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> ng g c piedPage
CREATE src/app/pied-page/pied-page.component.html (24 bytes)
CREATE src/app/pied-page/pied-page.component.spec.ts (641 bytes)
CREATE src/app/pied-page/pied-page.component.ts (286 bytes)
CREATE src/app/pied-page/pied-page.component.css (0 bytes)
UPDATE src/app/app.module.ts (978 bytes)
```

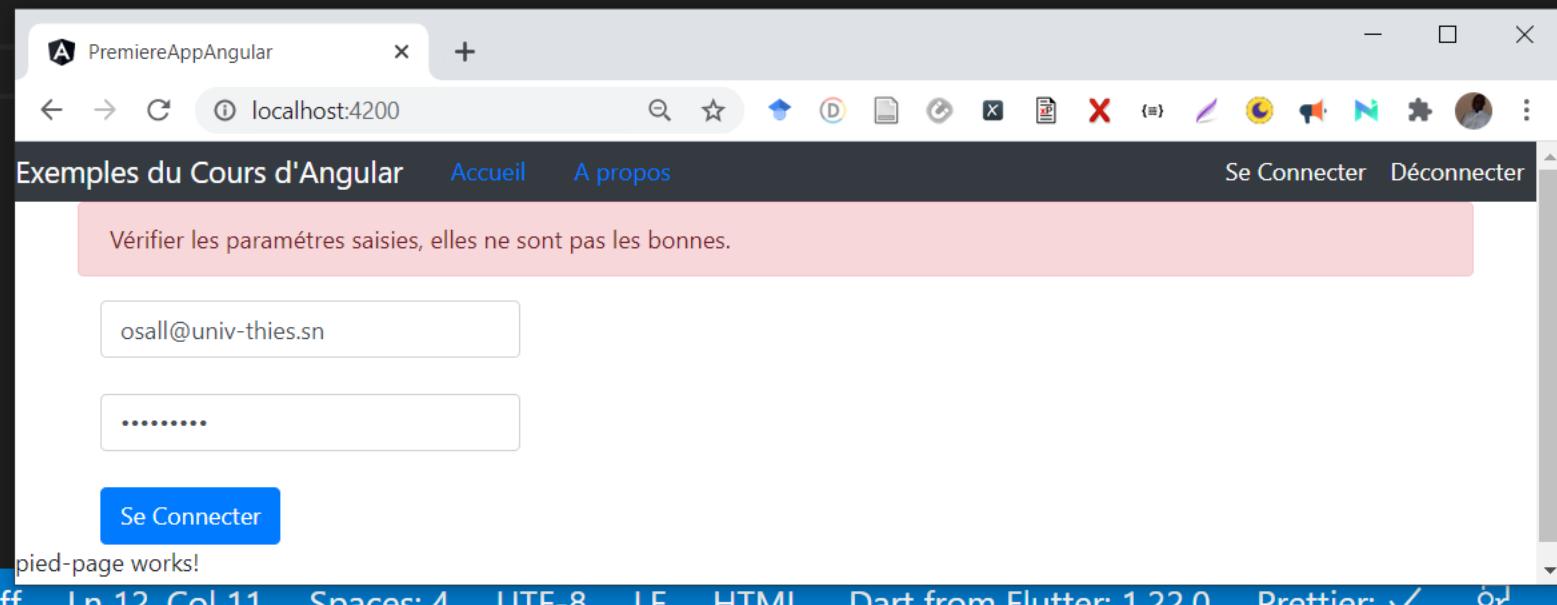


## entete-page.component.html X



```
> > entete-page.component.html > header > nav.nav.navbar-expand-md.navbar-dark.bg-dark > ul.navbar-nav.navbar-collapse.justify-content-end >
```

```
1  <header>
2    <nav class="nav navbar-expand-md navbar-dark bg-dark">
3      <div class="nav-item">
4        <a href="http://www.univ-thies.sn" class="navbar-brand">Exemples du Cours d'Angular</a>
5      </div>
6      <li class="nav-item"> <a class="nav-link active" href="/accueil">Accueil</a> </li>
7      <li class="nav-item"> <a class="nav-link active" href="/apropos">A propos</a> </li>
8      <ul class="navbar-nav navbar-collapse justify-content-end">
9        <li> <a class="nav-link active" href="/login">Se Connecter</a> </li>
10       <li> <a class="nav-link active" href="/logout">Déconnecter</a> </li>
11     </ul>
12   </nav>
13 </header>
```





EXPLORER

...

pied-page.component.css X

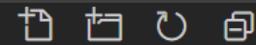
src &gt; app &gt; pied-page &gt; pied-page.component.css &gt; footer

```
1 footer{  
2     position: absolute;  
3     bottom: 0;  
4     width: 100%;  
5     height: 50px;  
6     background-color: #222222;  
7     color: white;  
8     text-align: center;  
9     text-decoration: solid;  
10    }
```

A screenshot of a web browser window titled "PremiereAppAngular". The URL is "localhost:4200/login". The page content includes a header with links for "Exemples du Cours d'Angular", "Accueil", "A propos", "Se Connecter", and "Déconnecter". Below the header is a form with two input fields and a "Se Connecter" button. A red error message at the top of the form area says "Vérifier les paramètres saisies, elles ne sont pas les bonnes.". At the bottom right of the page, there is a copyright notice "Tous Droits Réservés (C) 2020".



&gt; OPEN EDITORS



PREMIEREAPPANGULAR



e2e

node\_modules

src

app

- accueil
- entete-page
- login
- pied-page
  - pied-page.component.css
  - pied-page.component.html

pied-page.component.css

pied-page.component.ts

pipes

services

app-routing.module.ts

app.component.css

app.component.html

app.component.spec.ts

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES



pied-page.component.html X

src &gt; app &gt; pied-page &gt; pied-page.component.html

```
1 <footer>  
2     <div class="container">  
3         Tous Droits Réservés (C) 2020  
4     </div>  
5 </footer>
```

EXPLORER ...

> OPEN EDITORS

PREMIEREAPPANGULAR

- > e2e
- > node\_modules
- ✓ src
  - ✓ app
    - ✓ accueil
      - accueil.component.css
      - accueil.component.html
      - accueil.component.spec.ts
      - accueil.component.ts
    - entete-page
    - login
    - ✓ pied-page
      - pied-page.component.css
      - pied-page.component.html
      - pied-page.component.spec.ts
      - pied-page.component.ts
    - > pipes
    - > services

> OUTLINE

> TIMELINE

> DEPENDENCIES

accueil.component.html X

src > app > accueil > accueil.component.html

```
1 <div class="container">
2   <div class="alert alert-success" role="alert">
3     <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
4   </div>
5   <h1>Liste des Rendez-Vous</h1>
6   <table class="table">
7     <thead class="thead-dark">
8       <tr>
9         <th>Id</th>
10        <th>Description</th>
11        <th>Lieu</th>
12        <th>Date</th>
13      </tr>
14    </thead>
15    <tbody>
16      <tr>
17        <td>1</td>
18        <td>Cours Angular</td>
19        <td>UT</td>
20        <td>10 October 2020 à 12:00:00</td>
21      </tr>
22      <tr>
23        <td>2</td>
24        <td>Cours Flutter</td>
25        <td>UGB</td>
26        <td>11 October 2010 à 12:00:00</td>
27      </tr>
28      <tr>
29        <td>3</td>
30        <td>Cours Spring Boot</td>
31        <td>UADB</td>
32        <td>09 December 2021 à 12:00:00</td>
33      </tr>
34    </tbody>
35  </table>
36</div>
```



# Gestion des Sessions avec **localStorage** et **sessionStorage**

- **localStorage** et **sessionStorage**, qui font partie de l'API de stockage Web, sont deux excellents outils pour enregistrer localement des paires clé / valeur.
- **LocalStorage** et **sessionStorage** offrent tous deux des avantages par rapport à l'utilisation de cookies:
  - Les données sont enregistrées uniquement localement et ne peuvent pas être lues par le serveur, ce qui élimine le problème de sécurité que présentent les cookies.
  - Il permet d'enregistrer beaucoup plus de données (10 Mo pour la plupart des navigateurs).
  - Il est plus simple à utiliser et la syntaxe est très simple.
- Il est également pris en charge dans tous les navigateurs modernes, vous pouvez donc l'utiliser aujourd'hui sans problème. Évidemment, comme les données ne peuvent pas être lues sur le serveur, les cookies ont toujours une utilité, notamment en matière d'authentification.

# Gestion des Sessions avec **localStorage** et **sessionStorage**.

- Pour créer ou mettre à jour une variable session

- **localStorage.setItem("nom", "SALL");**
- On peut aussi faire **localStorage["nom"] = "SALL";**
- Ou encore **localStorage.nom = "SALL";**

- Pour récupérer une variable de localStorage

```
let nom = localStorage.getItem("nom");
```

- Pour supprimer une variable de localStorage

```
localStorage.removeItem("nom");
```

- Pour vider localStorage **localStorage.clear();**

**Même syntaxe et méthodes pour sessionStorage**

# Gestion des Sessions avec **localStorage** et **sessionStorage**

- Pour connaître le nombre de variables session

**localStorage.length**

- Pour stocker un objet dans localStorage

```
let perso = { nom: "SALL", prenom: "Ousmane" };
localStorage.setItem("person", JSON.stringify(perso));
```

- Pour récupérer un objet de localStorage

```
let p=JSON.parse(localStorage.getItem("person"));
```

Même syntaxe et méthodes pour **sessionStorage**

# Gestion des Sessions avec **localStorage** et **sessionStorage**

- Pour tester la compatibilité de **localStorage** avec le navigateur

```
if(window.localStorage) {  
    ...  
}
```

- Pour itérer sur **localStorage**

```
for(let i = 0; i < localStorage.length; i++){  
    let clef = localStorage.key(i);  
    let valeur = localStorage.getItem(clef);  
    console.log(clef, valeur);  
}
```

Même syntaxe et méthodes pour **sessionStorage**

# Gestion des Sessions

## **localStorage vs. sessionStorage**

- **localStorage** et **sessionStorage** accomplissent exactement la même chose et ont la même API, mais avec **sessionStorage**, les données ne sont conservées que jusqu'à la fermeture de la fenêtre ou de l'onglet
- Tandis qu'avec **localStorage**, les données sont conservées jusqu'à ce que l'utilisateur vide manuellement le cache du navigateur ou jusqu'à ce que votre application Web efface les données.



entete-page.component.html

entete-page.component.ts

authentication.service.ts



```
src > app > services > authentication.service.ts > AuthenticationService > seLoguer
      ,
6   < export class AuthenticationService {
7
8     constructor() {}
9
10    seLoguer(nomUtilisateur:string, motDePasse:string) {
11      if (nomUtilisateur === 'osall@univ-thies.sn' &&motDePasse === 'passer123'){
12        sessionStorage.setItem("nomUtilisateur",nomUtilisateur);
13        return true;
14      }
15      return false;
16    }
17
18    utilisateurConnecte(){
19      if(sessionStorage.getItem("nomUtilisateur")===null)
20        return false;
21      return true;
22    }
23  }
24
```

master\*



0 △ 0

Compile Hero: Off

Ln 12, Col 21 (14 selected)

Spaces: 2

UTF-8

LF

TS

4.0.3

Prettier: ✓



A PremiereAppAngular

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos Se Connecter Déconnecter

# Bienvenue osall@univ-thies.sn

## Liste des Rendez-Vous

Id	Description	Lieu	Date
1	Cours Angular	UT	10 October 2020 à 12:00:00
2	Cours Flutter	UGB	11 October 2010 à 12:00:00
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00

Tous Droits Réservés (C) 2020

Application  
Manifest  
Service Workers  
Clear storage

Storage  
Local Storage  
Session Storage  
http://localhost:4200  
IndexedDB  
Web SQL  
Cookies

Cache  
Cache Storage  
Application Cache

Background Services  
Background Fetch  
Background Sync  
Notifications  
Payment Handler  
Periodic Background Sync  
Push Messaging

Frames  
top

Console

Angular is running in development mode. Call enableProdMode() to enable production mode.  
[WDS] Live Reloading enabled.

Application

Key: nomUtilisateur Value: osall@univ-thies.sn

Line 1, Column 1

1 osall@univ-thies.sn

1 hidden core.js:26882 client:52



A entete-page.component.ts X

A authentication.service.ts

S entete-page.component.html

A accueil.co



src &gt; app &gt; entete-page &gt; A entete-page.component.ts &gt; EntetePageComponent &gt; nomUtilisateur

```
5   @Component({
6     selector: 'app-entete-page',
7     templateUrl: './entete-page.component.html',
8     styleUrls: ['./entete-page.component.css'],
9   })
10  export class EntetePageComponent implements OnInit {
11    nomUtilisateur=null;
12    constructor(public authenticationService: AuthenticationService, private router: Router) {}

14    ngOnInit(): void {
15      this.nomUtilisateur=sessionStorage.getItem("nomUtilisateur");
16    }

18    seDeconnecter() {
19      sessionStorage.removeItem('nomUtilisateur');
20      this.router.navigateByUrl('/login');
21    }
22  }
```



A entete-page.component.ts

A authentification.service.ts

S entete-page.component.html

A accueil.co



src &gt; app &gt; entete-page &gt; S entete-page.component.html

```
1   <header>
2     <nav class="nav navbar-expand-md navbar-dark bg-dark">
3       <div class="nav-item">
4         <a href="http://www.univ-thies.sn" class="navbar-brand">Exemples du Cours d'Angular</a>
5       </div>
6       <li class="nav-item">
7         <a *ngIf="authentificationService.utilisateurConnecte()">
8           class="nav-link active" [routerLink]="[ '/accueil', nomUtilisateur]">Accueil</a>
9       </li>
10      <li class="nav-item" *ngIf="authentificationService.utilisateurConnecte()">
11        <a class="nav-link active" href="/apropos">A propos</a>
12      </li>
13      <ul class="navbar-nav navbar-collapse justify-content-end">
14        <li *ngIf="!authentificationService.utilisateurConnecte()">
15          <a class="nav-link active" href="/login">Se Connecter</a>
16        </li>
17        <li *ngIf="authentificationService.utilisateurConnecte()">
18          <a class="nav-link active" (click)="seDeconnecter()">Déconnecter</a>
19        </li>
20      </ul>
21    </nav>
```





A entete-page.component.ts

S entete-page.component.html

A accueil.component.ts X



```
src > app > accueil > A accueil.component.ts > AccueilComponent > constructor
12
13     rendezVous = [
14         new RendezVous(1, "Cours Angular", "UT-THIES", new Date('10/10/2020')),
15         new RendezVous(2, 'Cours Flutter', "UGB-Saint-Louis", new Date('10/11/2010')),
16         new RendezVous( 3, 'Cours Spring Boot', "UADB-Bambey", new Date('12/09/2021'))
17     ];
18
19     constructor(private authenticationService: AuthenticationService,
20                 private routeActive: ActivatedRoute, private router: Router) {}
21
22     ngOnInit(): void {
23         if(this.authenticationService.utilisateurConnecte())
24             this.nomUtilisateur = this.routeActive.snapshot.paramMap.get('nomUtilisateur');
25         else
26             this.router.navigateByUrl('/login');
27     }
28 }
29
30 export class RendezVous {
31     constructor(private _id: number,  private _description: string,  private _lieu: string,
32                 private date: Date) {}}
```



...



A PremiereAppAngular

localhost:4200/login

# Exemples du Cours d'Angular

## Se Connecter

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

osall@univ-thies.sn

.....

Se Connecter

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
  - http://localhost:4200
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Console

Select a value to preview

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

core.js:26882 client:52

Tous Droits Réservés (C) 2020

A PremiereAppAngular

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos Déconnecter

# Bienvenue osall@univ-thies.sn

## Liste des Rendez-Vous

Id	Description	Lieu	Date
1	Cours Angular	UT	10 October 2020 à 12:00:00
2	Cours Flutter	UGB	11 October 2010 à 12:00:00
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00

Tous Droits Réservés (C) 2020

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
  - http://localhost:4200
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Console

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

Select a value to preview

Filter

Key	Value
nomUtilisateur	osall@univ-thies.sn

Default levels ▾

3 hidden

core.js:26882

client:52

A PremiereAppAngular

localhost:4200/accueil/null

Exemples du Cours d'Angular Accueil A propos Déconnecter

# Bienvenue osall@univ-thies.sn

## Liste des Rendez-Vous

Id	Description	Lieu	Date
1	Cours Angular	UT	10 October 2020 à 12:00:00
2	Cours Flutter	UGB	11 October 2010 à 12:00:00
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00

Tous Droits Réservés (C) 2020

localhost:4200/accueil/null

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
  - http://localhost:4200
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Console

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

Select a value to preview

Filter

Default levels ▾

3 hidden

core.js:26882

client:52

A PremiereAppAngular

localhost:4200/login

# Exemples du Cours d'Angular

## Se Connecter

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

osall@univ-thies.sn

.....

Se Connecter

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
  - http://localhost:4200
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Console

Select a value to preview

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

Tous Droits Réservés (C) 2020



# Interaction entre composants **@input** et **@output**

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage

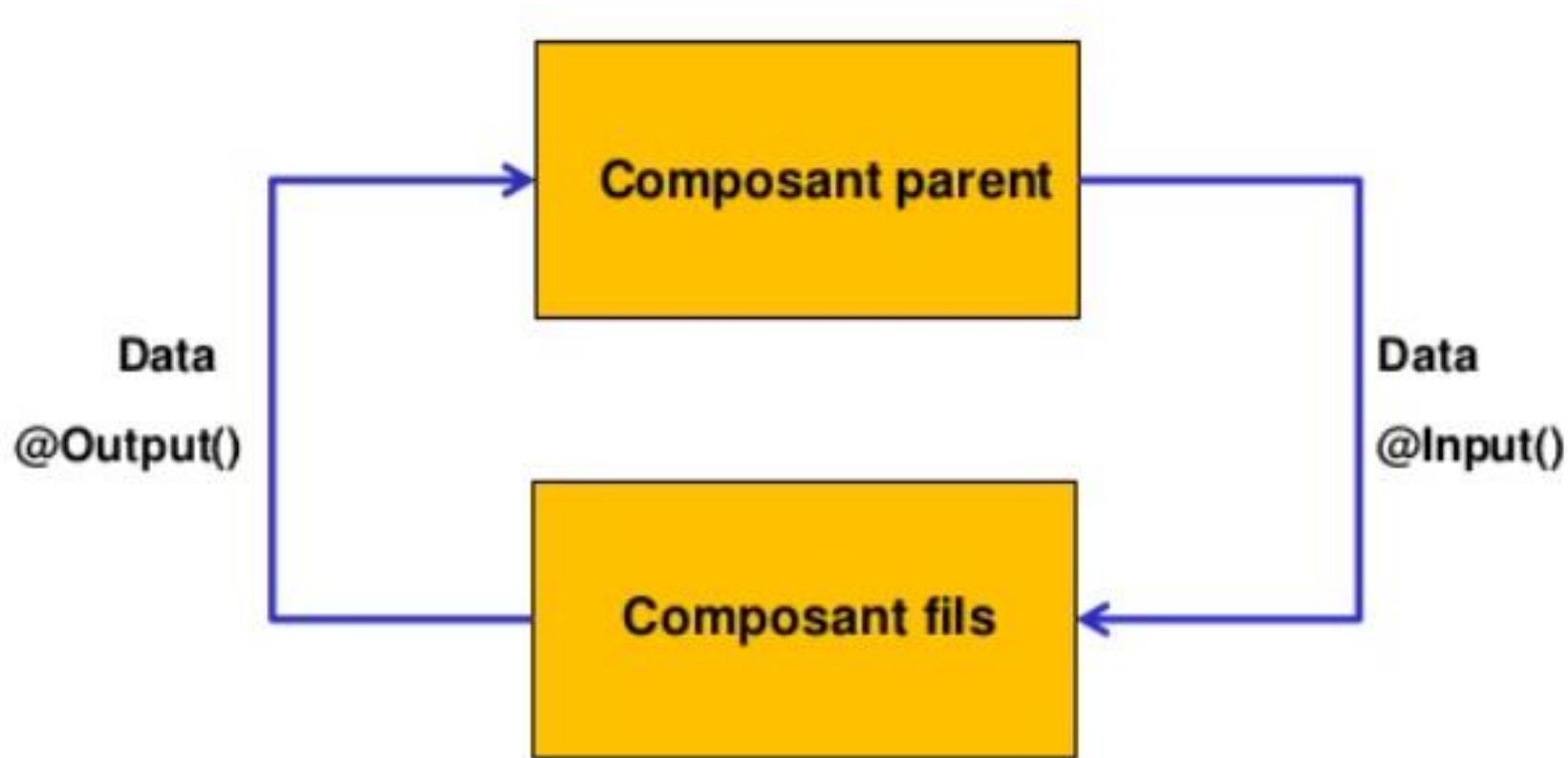


# Contenu Cours Angular

- **Interaction entre composants @input et @output**
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Interaction entre composants



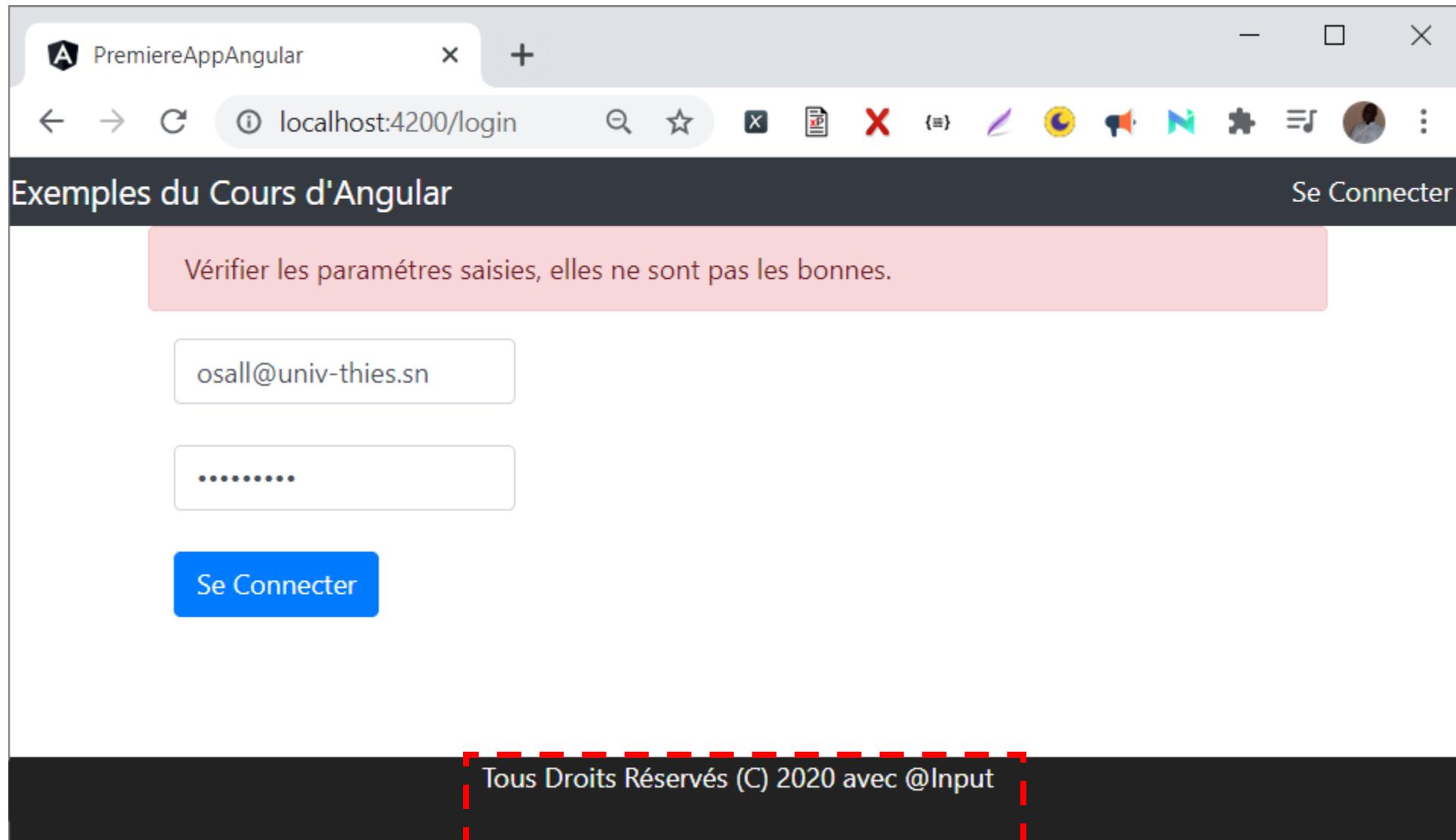
# Interaction entre composants

- Première forme d'interaction
  - Une application Angular est composée de plusieurs composants
  - En utilisant des formulaires et des liens, on peut envoyer des données d'un composant à un autre
- Autre forme d'interaction
  - On peut ajouter le sélecteur d'un premier composant dans la template d'un deuxième composant
    - on appelle le premier composant : **composant fils**
    - on appelle le deuxième composant : **composant parent**
  - En utilisant les décorateurs **@Input()** et **@Output()** les deux composants peuvent échanger de données

# Le décorateur **@Input()**

- **@Input()** : permet à un **composant fils** de récupérer des données de son **composant parent**
- **Cas d'études:** Supposons que l'on souhaite afficher dans le composant fils **PiedPageComponent** le texte du pied de page. Le décorateur **@input** permet de créer une propriété qu'on peut fixer depuis la balise **<app-pied-page>** du composant parent.

```
<app-pied-page [messageParent]="messagePiedPage"></app-pied-page>
```



EXPLORER

> OPEN EDITORS

PREMIEREAPPAN... 60

- app
  - accueil
  - entete-page
  - login
  - pied-page
  - pipes
  - services
- app-routing.module.ts M
- app.component.css M
- app.component.html M
- app.component.spec.ts A
- app.component.ts M
- app.module.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

A app.component.ts X

src > app > A app.component.ts > AppComponent > messagePiedPage

1 import { Component, Input } from '@angular/core';  
2  
3 @Component({  
4 selector: 'app-root',  
5 templateUrl: './app.component.html',  
6 // template:<h1 class='text-center'>Titre: {{title}}</h1>,  
7 styleUrls: ['./app.component.css']  
8 })  
9 export class AppComponent {  
10 title = "Exemples du cours d'Angular";  
11  
12 messagePiedPage="Tous Droits Réservés (C) 2020 avec @Input";  
13 }  
14

EXPLORER

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPAN... 60

- > pipes
- > services
- A app-routing.module.ts M
- E app.component.css M
- S app.component.html M
- A app.component.spec.ts A
- A app.component.ts M
- A app.module.ts M
- > assets
- > environments
- ★ favicon.ico A
- E index.html A
- TS main.ts A
- TS polyfills.ts A

> OUTLINE

> TIMELINE

> DEPENDENCIES

app.component.html

```
src > app > app.component.html
1 | <!-- <div class="container">
2 |   <h1 class="text-center">{{title}}</h1>
3 | </div> -->
4 |
5 | <!-- <app-accueil></app-accueil> -->
6 |
7 | <!-- <app-login></app-login> -->
8 | <app-entete-page></app-entete-page>
9 | <router-outlet></router-outlet>
10| <app-pied-page [messageParent]="messagePiedPage"></app-pied-page>
11| <!-- Les Composants seront affichées ici -->
```



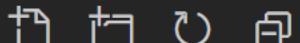
EXPLORER

...

A pied-page.component.ts X

...

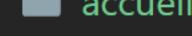
&gt; OPEN EDITORS



&lt; PREMIEREAPPAN...



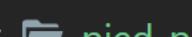
app



accueil



entete-page



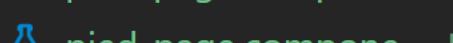
login



pied-page



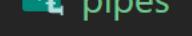
pied-page.compone...



pied-page.compone...



pied-page.compone...



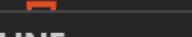
pipes



services



app-routing.module.ts M



app.component.css M

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

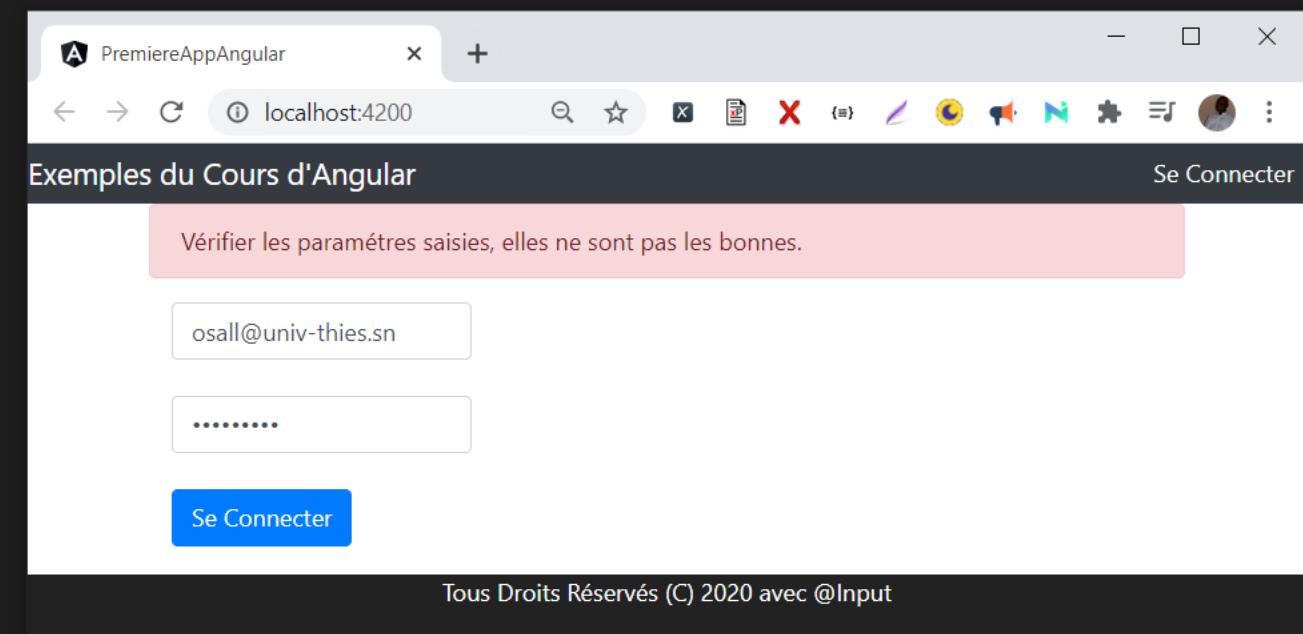
```
1 import { Component, Input, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-pied-page',
5   templateUrl: './pied-page.component.html',
6   styleUrls: ['./pied-page.component.css']
7 })
8 export class PiedPageComponent implements OnInit {
9   @Input("messageParent")
10  messagePiedPage:string
11
12  constructor() { }
13
14  ngOnInit(): void {
15
16
17 }
18 }
```

- EXPLORER
- > OPEN EDITORS
- PREMIEREAPPANGULAR
  - > app
    - accueil
    - entete-page
    - login
  - > pied-page
    - pied-page.component.css
    - pied-page.component.html
    - pied-page.component.spec.ts
    - pied-page.component.ts
  - > pipes
  - > services
  - app-routing.module.ts
  - app.component.css
  - app-component.html
- > OUTLINE
- > TIMELINE
- > DEPENDENCIES

5 pied-page.component.html X

src > app > pied-page > 5 pied-page.component.html

```
1 <footer>
2   <div class="container">
3     {{messagePiedPage}}
4   </div>
5 </footer>
```



# Le décorateur `@Output()`

- `@Output()` : permet à un **composant parent** de récupérer des données de son **composant fils**
- **Cas d'études:** Jusqu'à présent, il est très simple d'utiliser la liaison d'événement pour obtenir le bouton pour appeler la fonction dans le composant. Maintenant, modifions un peu l'exigence. Que faire si vous souhaitez exécuter une fonction du composant parent **AppComponent** sur l'événement click d'un bouton dans le composant fils **AppEntetePageComponent** ? Pour ce faire, vous devrez émettre l'événement de clic de bouton depuis le composant fils **AppEntetePageComponent**. Importez **EventEmitter** et la sortie de **@angular/core**.

A PremiereAppAngular

localhost:4200/login

# Exemples du Cours d'Angular

Se Connecter

## @Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

osall@univ-thies.sn

.....

Se Connecter

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

1  
2  
3  
4  
5  
6  
7  
8

Tous Droits Réservés (C) 2020 avec @Input

Console



entete-page.component.ts X

```
src > app > entete-page > A entete-page.component.ts > EntetePageComponent
1 import { Component, EventEmitter, OnInit, Output } from '@angular/core';
2 import { Router } from '@angular/router';
3 import { AuthentificationService } from '../services/authentification.service';
4
5 @Component({
6   selector: 'app-entete-page',
7   templateUrl: './entete-page.component.html',
8   styleUrls: ['./entete-page.component.css'],
9 })
10 export class EntetePageComponent implements OnInit {
11   nomUtilisateur=null;
12
13   @Output() changementValeurEvent=new EventEmitter();
14   compteur=0;
15
16   compteurMethode(){
17     this.compteur++;
18     this.changementValeurEvent.emit(this.compteur);
19   }
20
21   constructor(public authentificationService: AuthentificationService, private router: Router) {}
22
23   ngOnInit(): void {
24     this.nomUtilisateur=sessionStorage.getItem("nomUtilisateur");
25   }
26 }
```

À l'heure actuelle, nous effectuons les tâches suivantes dans la classe **EntetePageComponent**:

1. Création d'une variable appelée **compteur**, qui sera passée en paramètre de l'événement émis.
2. Création d'un **EventEmitter**, **changementValeurEvent**, qui sera émis vers le composant parent lors de l'événement click du bouton.
3. Création d'une fonction nommée **compteurMethode()**. Cette fonction est appelée sur l'événement click du bouton, et à l'intérieur de l'événement de fonction **changementValeurEvent** est émis.
4. Lors de l'émission de l'événement **changementValeurEvent**, la valeur du **compteur** est passée en paramètre.

Dans le composant parent, **AppComponent**, le composant enfant, **EntetePageComponent**, peut être utilisé comme indiqué dans le code.



## entete-page.component.html X

```
src > app > entete-page > entete-page.component.html
  4   |         <a href="http://www.univ-thies.sn" class="nav-link active" [routerLink]=[ 'home' ]>
  5   |     </div>
  6   |     <li class="nav-item">
  7   |       <a *ngIf="authentificationService.utilisateurs.length > 0" href="#" class="nav-link active" [routerLink]=[ 'profil' ]>
  8   |     </a>
  9   |   </li>
 10   |   <li class="nav-item" *ngIf="authentifications.length > 0">
 11   |     <a class="nav-link active" href="/apropos">
 12   |   </li>
 13   |   <ul class="navbar-nav navbar-collapse justify-content-end">
 14   |     <li *ngIf="!authentificationService.utilisateurs.length > 0">
 15   |       <a class="nav-link active" href="/login">
 16   |     </li>
 17   |     <li *ngIf="authentificationService.utilisateurs.length > 0">
 18   |       <a class="nav-link active" (click)="sousMenuAfficher()">
 19   |     </li>
 20   |   </ul>
 21   |   <ul>
 22   |     <li>
 23   |       <button class="btn btn-primary" type="button" (click)="compteurMethode()">
 24   |         @Output- Incrémenter le compteur
 25   |       </button>
 26   |     </li>
 27   |   </ul>
 28   | </nav>
 29 </header>
```

À l'heure actuelle, nous effectuons les tâches suivantes dans la classe **EntetePageComponent**:

1. Création d'une variable appelée **compteur**, qui sera passée en paramètre de l'événement émis.
2. Création d'un **EventEmitter**, **changementValeurEvent**, qui sera émis vers le composant parent lors de l'événement click du bouton.
3. Création d'une fonction nommée **compteurMethode()**. Cette fonction est appelée sur l'événement click du bouton, et à l'intérieur de l'événement de fonction **changementValeurEvent** est émis.
4. Lors de l'émission de l'événement **changementValeurEvent**, la valeur du compteur est passée en paramètre.

Dans le composant parent, **AppComponent**, le composant enfant, **EntetePageComponent**, peut être utilisé comme indiqué dans le code.



EXPLORER

3

A app.component.ts X



## > OPEN EDITORS

#### ▼ PREMIEREAPPANGULAR

src

✓  app

>  accueil

> entete-page

login

5

1

## File pipes

## A app-routing.module.ts

### 3 app.component.css

5 app.component.html

## App component specs

## A app.component.ts

## A app module ts

src > app > A app.component.ts > 📥 AppComponent > ⚙️ afficherCompteur

```
1 import { Component, Input} from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   // template:<h1 class='text-center'>Titre: {{title}}</h1>,
7   styleUrls: ['./app.component.css']
8 })
9 export class AppComponent {
10   title = "Exemples du cours d'Angular";
11
12   messagePiedPage="Tous Droits Réservés (C) 2020 avec @Input";
13
14   afficherCompteur(compteur:number){
15     console.log(compteur);
16
17 }
```

Comme vous pouvez le voir, la fonction AppComponent est appelée sur l'événement click du bouton placé sur l'EntetePageComponent. Ceci peut être fait avec `@Output` et `EventEmitter`. Lorsque vous exécutez l'application et cliquez sur le bouton, vous pouvez voir la valeur du compteur dans la console du navigateur. Chaque fois que vous cliquez sur le bouton, la valeur du compteur est augmentée de 1.



EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- src
  - app
    - accueil
    - entete-page
    - login
    - pied-page
    - pipes
  - services
  - app-routing.module.ts M
  - app.component.css M
  - app.component.html M
  - app.component.spec.ts A
  - app.component.ts M
  - app.module.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

app.component.html X

À l'heure actuelle, nous exécutons les tâches suivantes dans la classe **AppComponent**:

1. Utilisation de `<app-entete-page>` dans le modèle.
2. Dans l'élément `<app-entete-page>`, utilisez la liaison d'événement pour utiliser l'événement `changementValeurEvent`.
3. Appel de la fonction `afficherCompteur` sur l'événement `changementValeurEvent`.
4. Dans la fonction `afficherCompteur`, impression de la valeur du compteur passé depuis `EntetePageComponent`.

```
6
7 | <!-- <app-login></app-login> -->
8 | <app-entete-page (changementValeurEvent)="afficherCompteur($event)"><!--
9 | <router-outlet></router-outlet>
10| <app-pied-page [messageParent]="messagePiedPage"></app-pied-page>
11| <!-- Les Composants seront affichées ici -->
```

Comme vous pouvez le voir, la fonction `AppComponent` est appelée sur l'événement `click` du bouton placé sur l' **EntetePageComponent**. Ceci peut être fait avec `@Output` et `EventEmitter`. Lorsque vous exécutez l'application et cliquez sur le bouton, vous pouvez voir la valeur du compteur dans la console du navigateur. Chaque fois que vous cliquez sur le bouton, la valeur du compteur est augmentée de 1.

A PremiereAppAngular

localhost:4200/login

Exemples du Cours d'Angular

Se Connecter

@Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

osall@univ-thies.sn

.....

Se Connecter

Angular is running in development mode. Call enableProdMode() to enable production mode.

[WDS] Live Reloading enabled.

1  
2  
3  
4  
5  
6  
7  
8

Tous Droits Réservés (C) 2020 avec @Input

Console

# Travaux Pratiques n°6

<https://codelab.unangular.com/ponent-tree/>



# Observables et Services HTTP

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- **Observables et Services HTTP**
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



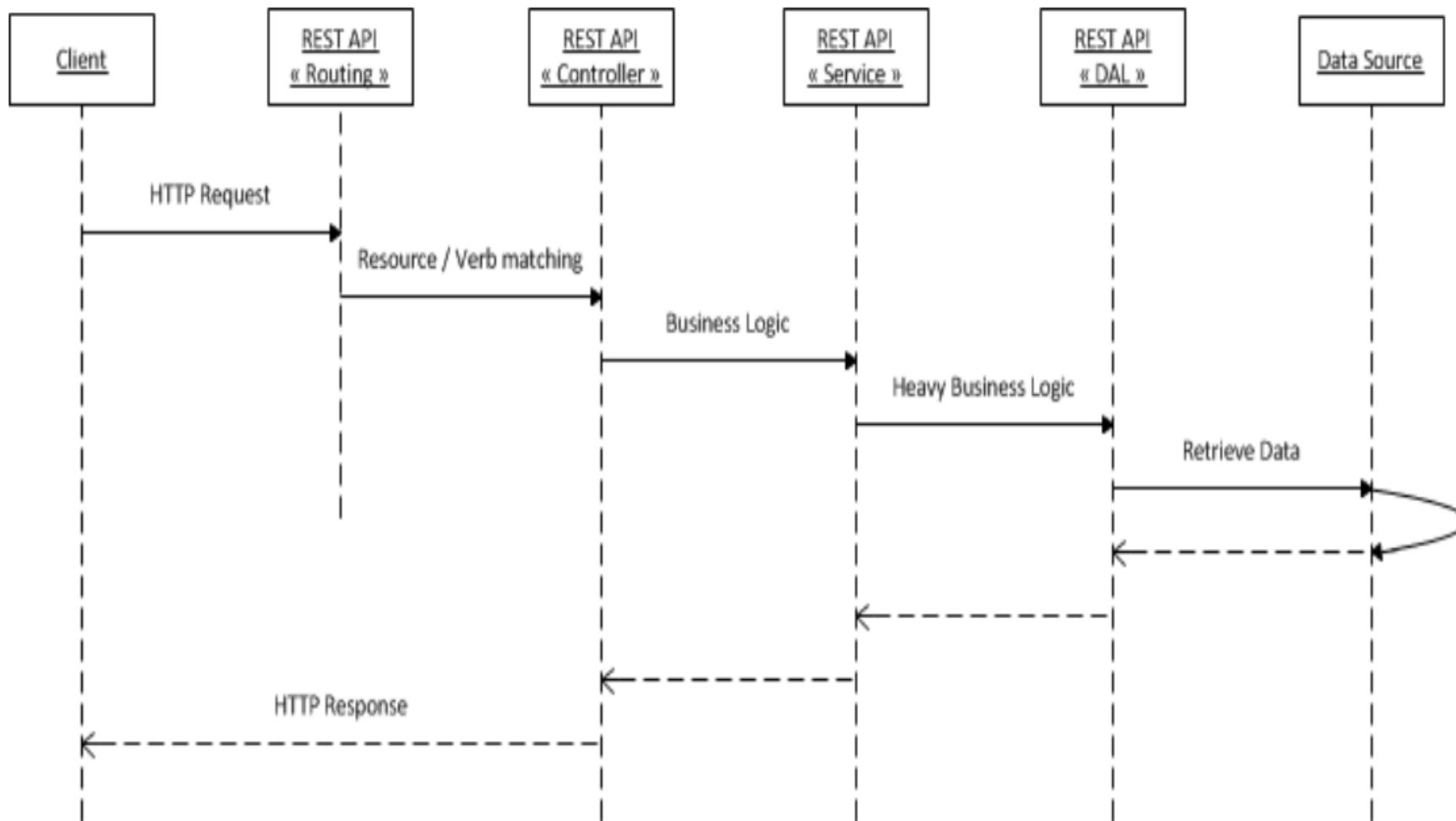
# Introduction aux services web

- Les services web sont des services informatiques de la famille des technologies web permettant la communication entre des applications hétérogènes dans des environnements distribués
- Les services web sont un moyen pour une application d'exposer son API au moyen du protocole http.
- Ils ont été proposés à la base comme solution d'intégrations de différents logiciels développés par des entreprises leur permettant de communiquer entre eux. Il existe deux principaux types de services web :
  - Les services web SAOP
  - Les services web REST
- Cette présentation met l'accent sur les services web REST ainsi que sur l'intégration d'un API REST avec une application Angular

# Qu'est-ce qu'un service web REST

- Acronyme de REpresentational State Transfer, REST est une approche plus simple de création de services web basée sur les URL et les méthodes HTTP (GET, POST, PUT, DELETE)
- Contrairement à HTTP, REST n'est pas un protocole ou un format, mais plutôt un style d'architecture inspiré de l'architecture du web fortement basé sur le protocole http
- La simplicité de l'architecture REST fait que les API "REST", sont propres et polyvalentes pour beaucoup de types d'échange d'information entre logiciels.

# Principe de fonctionnement



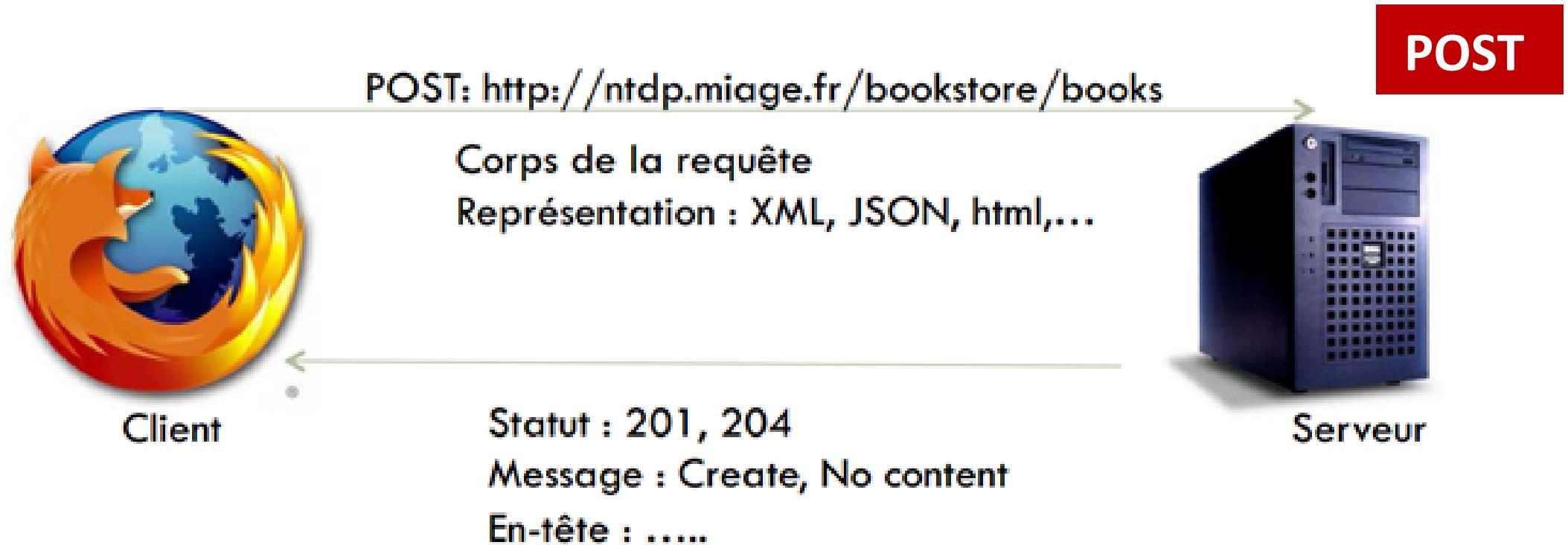
# Principe de fonctionnement

- L'accès aux ressources du serveur s'effectue via des URL servant de point d'entrée et des méthodes permettant de manipuler ces ressources
- Une ressource peut subir quatre opérations de bases CRUD correspondant aux quatre principaux types de requêtes HTTP (GET, PUT, POST, DELETE)
  - RETRIEVE → GET
  - CREATE → POST
  - UPDATE → PUT
  - DELETE → DELETE

# La méthode GET renvoie une représentation de la ressource telle qu'elle est sur le système



# La méthode POST crée une nouvelle ressource sur le système



# PUT Mise à jour de la ressource sur le système

PUT



# DELETE Supprime la ressource identifiée par l'URI sur le serveur

DELETE



# API REST

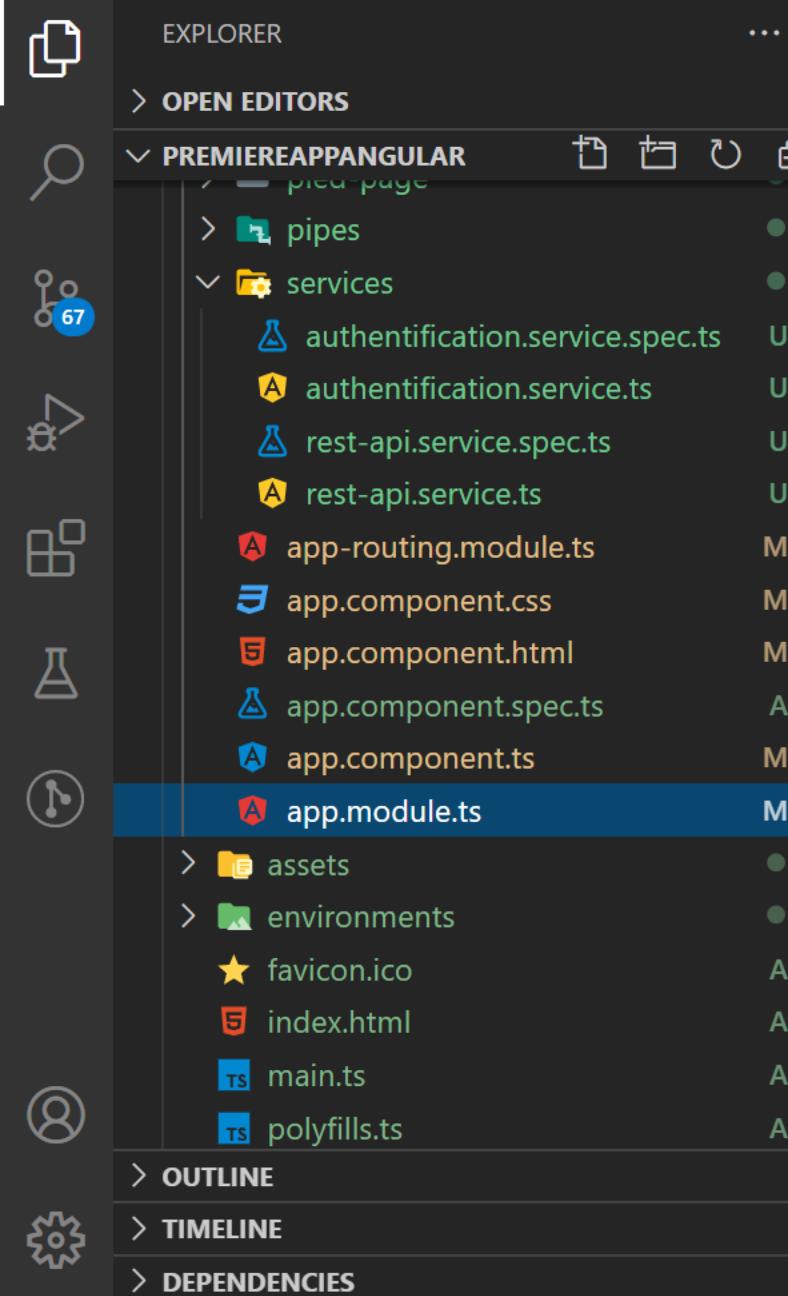
- Angular est un Framework JavaScript qui permet de réaliser des applications web qui s'exécutent côté client
- Angular possède un module, **HttpModule** puis **HttpClientModule**, qui facilite la réalisation de requête **HTTP** vers n'importe quel serveur
- Pour le côté serveur : on peut utiliser un serveur **JSON** qui recevra des requêtes HTTP envoyées par Angular et retourner une réponse
- **json-server** est un serveur Http de test, open-source, pour les développeurs Front-End. Documentation :  
<https://github.com/typicode/json-server>

# API REST

- Certaines des opérations de base qui retournent toujours un Observable sont les suivantes
  - Create: `http://localhost:3000/rv` POST (verb)
  - Read All: `http://localhost:3000/rv` GET (verb)
  - Read One: `http://localhost:3000/rv/: id` GET (verb)
  - Update: `http://localhost:3000/rv/: id` PUT (verb)
  - Delete: `http://localhost:3000/rv/: id` DELETE (verb)
  - **request()**
  - **head()**
  - **jsonp()**
  - **options()**
  - **patch()**

# API Rest

- La récupération des données sur Internet est nécessaire pour la plupart des applications. Heureusement, Angular et TypeScript fournissent des outils, pour ce type de travail.
- L'exemple utilise les étapes suivantes:
  - Ajoutez **HttpClientModule** dans **app.module.ts** et installer le service **HttpClient**
  - Configurer un **serveur JSON**
  - Créer un **modèle** pour les rendez-vous
  - Créer un **service REST** pour les **opérations CRUD**
  - Utiliser ces services dans **accueil.component.ts**



## app.module.ts ×

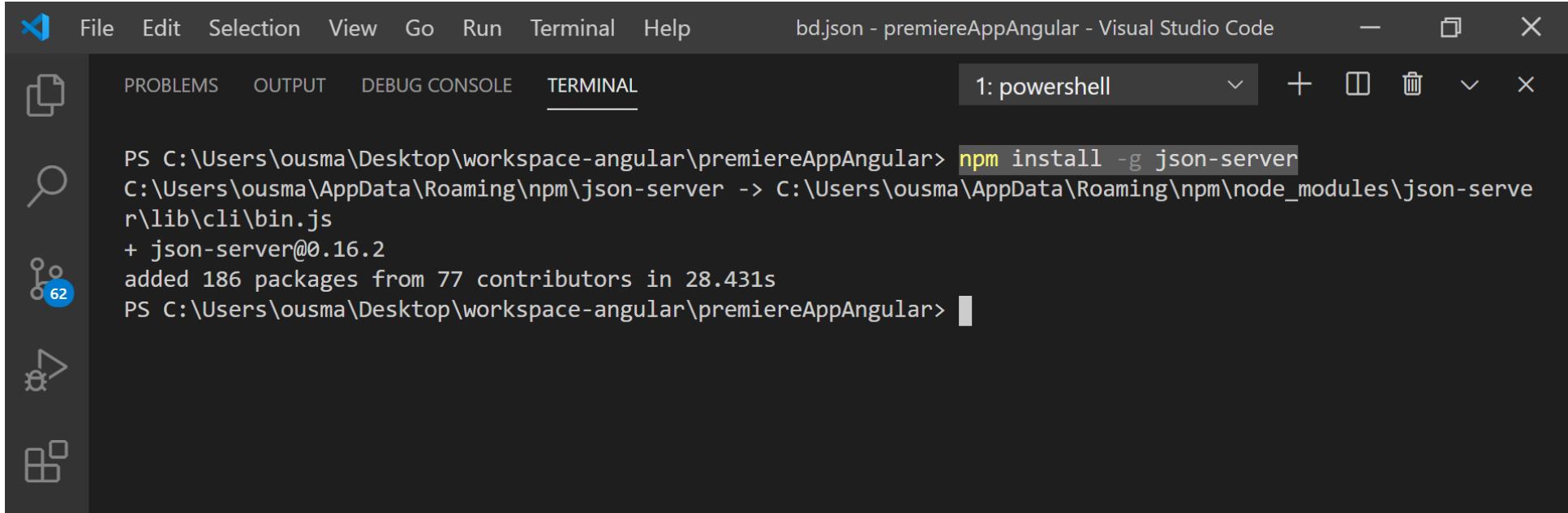
src &gt; app &gt; app.module.ts &gt; AppModule

```
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { AccueilComponent } from './accueil/accueil.component';
7 import { LoginComponent } from './login/login.component';
8 import { FormsModule } from '@angular/forms';
9 import { AuthenticationService } from './services/authentication.service';
10 import { GetStringAvantTiretPipe } from './pipes/get-string-avant-tiret.pipe';
11 import { EntetePageComponent } from './entete-page/entete-page.component';
12 import { PiedPageComponent } from './pied-page/pied-page.component';
13 import { HttpClientModule } from '@angular/common/http';

14
15 @NgModule({
16   declarations: [AppComponent, AccueilComponent, LoginComponent, GetStringAvantTiretP,
17   imports: [BrowserModule, AppRoutingModule, FormsModule, HttpClientModule],
18   providers: [AuthenticationService],
19   bootstrap: [AppComponent],
20 })
21 export class AppModule {}
```

# Configurer un serveur JSON

<https://github.com/typicode/json-server>



A screenshot of the Visual Studio Code interface. The title bar shows "bd.json - premiereAppAngular - Visual Studio Code". The left sidebar has icons for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the TERMINAL icon being blue and having a "62" badge. The main area is the terminal tab, titled "1: powershell". The output shows the command "npm install -g json-server" being run, along with its dependencies and execution details.

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> npm install -g json-server
C:\Users\ousma\AppData\Roaming\npm\json-server -> C:\Users\ousma\AppData\Roaming\npm\node_modules\json-server\lib\cli\bin.js
+ json-server@0.16.2
added 186 packages from 77 contributors in 28.431s
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

JSON Server est un projet simple qui vous aide à configurer très rapidement une API REST avec des opérations CRUD. Le site Web du projet peut être trouvé à <https://github.com/typicode/json-server>. Dans ce qui suit, vous apprendrez à configurer le serveur JSON et à publier un exemple d'API REST. En outre, vous verrez comment utiliser une autre bibliothèque, Faker.js, pour générer de fausses données pour l'API REST qui est exposée à l'aide du serveur JSON.



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell

+ □ └ ×

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular&gt; mkdir serveur

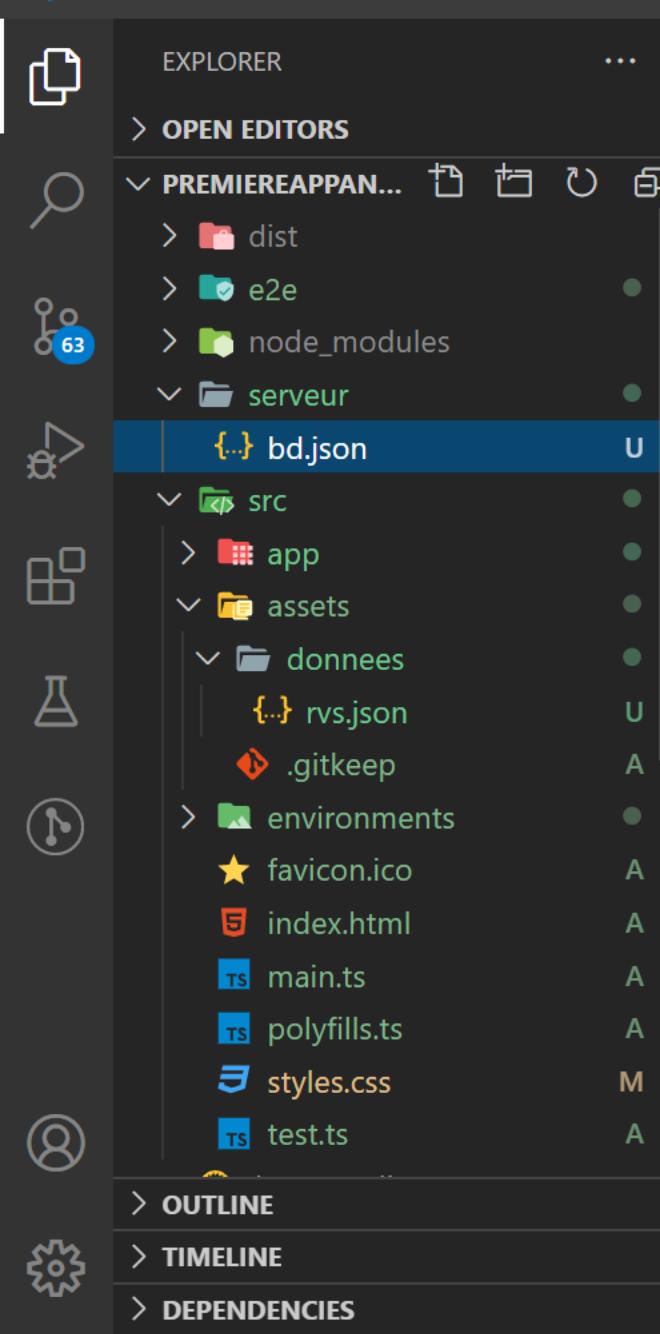
Répertoire : C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
d----	10/21/2020 3:09 PM		serveur

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> cd .\serveur\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\serveur>

...





bd.json

```
serveur > {...} bd.json > ...
```

```
1 [ "rv": [ 2 { 3 "id": 1, 4 "description": "Cours Angular", 5 "lieu": "UT-THIES", 6 "date": "10/10/2020" 7 }, 8 { 9 "id": 2, 10 "description": "Cours Flutter", 11 "lieu": "UGB-Saint-Louis", 12 "date": "10/11/2010" 13 }, 14 { 15 "id": 3, 16 "description": "Cours Spring Boot", 17 "lieu": "UADB-Bambey", 18 "date": "12/09/2021" 19 } 20 ] 21 ] 22 }
```



PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: node

PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> **json-server** --watch serveur/bd.json

\{^\_^}/ hi!

Loading serveur/bd.json

Done

**Resources**<http://localhost:3000/rv>**Home**<http://localhost:3000>

... Type s + enter at any time to create a snapshot of the database

Watching...



A PremiereAppAngular × JSON Server × +

localhost:3000

🔍 ⭐ 📤 📄 🗃 🖊️ 🖼️ 🎧 🎵 🎨 🎯 🎩

**JSON Server** ❤️ GitHub Sponsors 💧 My JSON Server 😊 Supporters

# Congrats!

You're successfully running JSON Server  
✧\*.٩(‘▽’\*)ጀ✧\*.

## Resources

/rv 3x

To access and modify resources, you can use any HTTP method:

GET POST PUT PATCH DELETE OPTIONS

undefined

## Documentation

[README](#)

To replace this page, create a `./public/index.html` file.

```
1 // 20201021152357
2 // http://localhost:3000/rv
3
4 [
5   {
6     "id": 1,
7     "description": "Cours Angular",
8     "lieu": "UT-THIES",
9     "date": "10/10/2020"
10 },
11 {
12   {
13     "id": 2,
14     "description": "Cours Flutter",
15     "lieu": "UGB-Saint-Louis",
16     "date": "10/11/2010"
17 },
18 {
19   "id": 3,
20   "description": "Cours Spring Boot",
21   "lieu": "UADB-Bambey",
22   "date": "12/09/2021"
23 }
```

```
1 // 20201021152416
2 // http://localhost:3000/rv/1
3
4 {
5   "id": 1,
6   "description": "Cours Angular",
7   "lieu": "UT-THIES",
8   "date": "10/10/2020"
9 }
```

```
ng serve -o
```

```
3 unchanged chunks
chunk {main} main.js, main.js.map (main) 48.5 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 2.99 MB [initial] [rendered]
Time: 2182ms
: Compiled successfully.
```

ERROR in node\_modules/@angular/common/http/http.d.ts:2822:22 - error NG6002: Appears in the NgModule.imports of AppModule, but could not be resolved to an NgModule class.

This likely means that the library (@angular/common/http) which declares HttpClientModule has not been processed correctly by ngcc, or is not compatible with Angular Ivy. Check if a newer version of the library is available, and update if so. Also consider checking with the library's authors to see if the library is expected to be compatible with Ivy.

```
2822 export declare class HttpClientModule {  
~~~~~
```

^CTerminer le programme de commandes (O/N) ? o

```
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>ng serve -o
```

dule has not been processed correctly by ngcc, or is not compatible with Angular Ivy. Check if a newer version of the library is available, and update if so. Also consider checking with the library's authors to see if the library is expected to be compatible with Ivy.

```
2822 export declare class HttpClientModule {
```

```
~~~~~
```

^CTerminer le programme de commandes (0/N) ? o

```
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>ng serve -o
```

```
chunk {main} main.js, main.js.map (main) 48.5 kB [initial] [rendered]
```

```
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
```

```
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
```

```
chunk {styles} styles.js, styles.js.map (styles) 686 kB [initial] [rendered]
```

```
chunk {vendor} vendor.js, vendor.js.map (vendor) 2.99 MB [initial] [rendered]
```

```
Date: 2020-10-21T15:33:33.708Z - Hash: 1a6c20d5951267345266 - Time: 19341ms
```

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

```
: Compiled successfully.
```

EXPLORER ...

> OPEN EDITORS

PREMIEREAPPANGULAR

- {...} bd.json U
- src ●
- app ●
- accueil ●
  - accueil.component.css U
  - accueil.component.h... U
  - accueil.component.s... U
  - accueil.component.ts U
- entete-page ●
- login ●
- pied-page ●
- pipes ●
- services ●
  - app-routing.module.ts M
  - app.component.css M
  - app.component.html M
  - app.component.spec.ts A
  - app.component.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

accueil.component.ts X

src > app > accueil > accueil.component.ts > RendezVous

```
27    }
28 }
29
30 export class RendezVous {
31   constructor(private _id: number, private _description: string, private _lieu: string,
32   private _date: Date) {}
33
34   public get id(): number {
35     return this._id;
36   }
37
38   public get description(): string {
39     return this._description;
40   }
41
42   public get lieu(): string {
43     return this._lieu;
44   }
45
46   public get date(): Date {
47     return this._date;
48   }
49 }
```

Compile Hero: Off Ln 49, Col 2 (389 selected) Spaces: 2 UTF-8 LF TS 4.0.3 Prettier: ✓

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- > serveur
- ✓ src
  - > app
    - > accueil
    - > entete-page
    - > login
    - > modeles
    - > pied-page
    - > pipes
    - > services
      - A app-routing.module.ts M
      - E app.component.css M
      - S app.component.html M
      - A app.component.spec.ts A
      - A app.component.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

accueil.component.ts X

src > app > accueil > accueil.component.ts > RendezVous

27 }  
28 }  
29 |  
30 export class RendezVous {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: powershell + - X

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> cd .\src\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src> cd .\app\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app> mkdir modeles
```

Répertoire : C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
d----	10/21/2020 3:40 PM		modeles

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app> cd .\modeles\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\modeles> █
```

Compile Hero: Off Ln 49, Col 2 (389 selected) Spaces: 2 UTF-8 LF TS 4.0.3 Prettier: ✓

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPAN... ⚡ ⚡ ⚡ ⚡

- > dist
- > e2e
- > node\_modules
- > serveur
- ✓ src
  - ✓ app
    - > accueil
    - > entete-page
    - > login
    - ✓ modeles
      - rendez-vous.spec.ts
      - rendez-vous.ts
    - > pied-page
    - > pipes
    - > services
    - app-routing.module.ts M
    - app.component.css M
    - app.component.html M

> OUTLINE

> TIMELINE

> DEPENDENCIES

ts rendez-vous.ts X

src > app > modeles > ts rendez-vous.ts > RendezVous > date

```
1  export class RendezVous {  
2      id: number;  
3      description: string;  
4      lieu: string;  
5      date: Date;  
6  }  
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

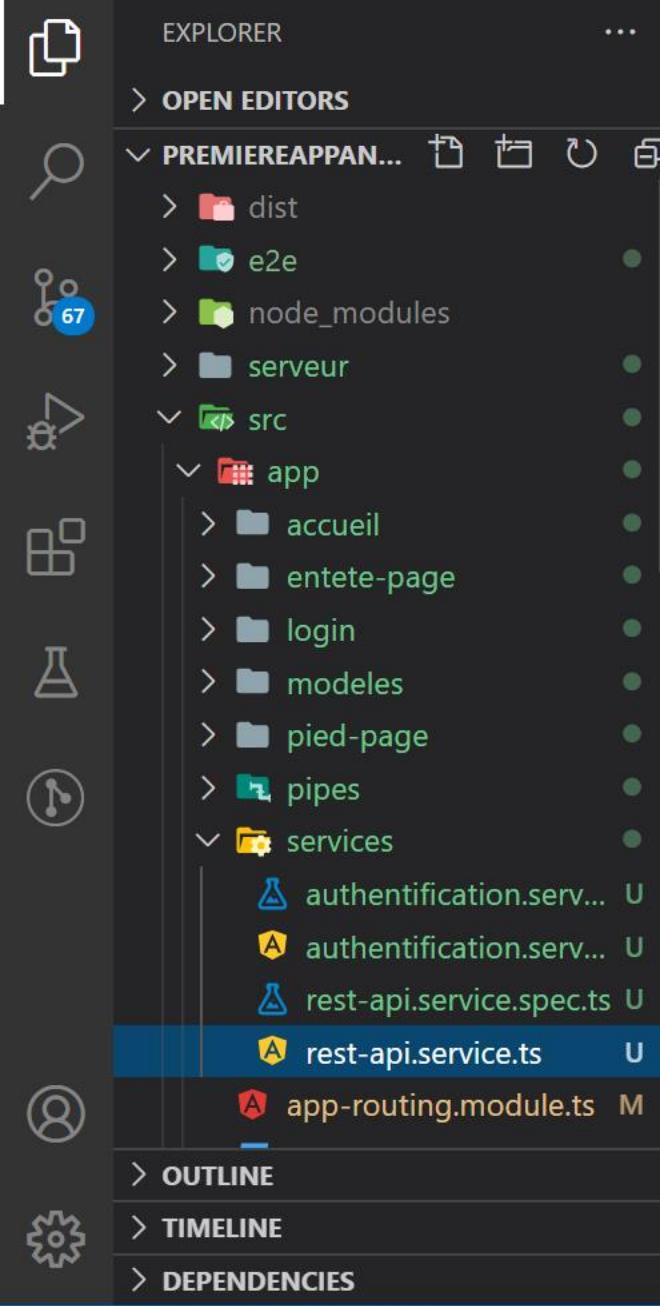
2: node

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src> cd .\app\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app> mkdir modeles
```

Répertoire : C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
d----	10/21/2020 3:40 PM		modeles

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app> cd .\modeles\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\modeles> ng g cl RendezVous  
CREATE src/app/modeles/rendez-vous.spec.ts (171 bytes)  
CREATE src/app/modeles/rendez-vous.ts (28 bytes)  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\modeles>
```



### rest-api.service.ts X

src > app > services > A rest-api.service.ts > ...

```
1 import { Injectable } from '@angular/core'; Calculating...
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class Rest ApiService {
7
8   constructor() { }
9 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: node

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app> cd .\services\
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services> ng g s rest-api
CREATE src/app/services/rest-api.service.spec.ts (363 bytes)
CREATE src/app/services/rest-api.service.ts (136 bytes)
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services>
```

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPAN... 📁 ⚡ ⚡ ⚡ ⚡

- > dist
- > e2e
- > node\_modules
- > serveur
- ✓ src
  - > app
    - > accueil
    - > entete-page
    - > login
    - > modeles
    - > pied-page
    - > pipes
  - > services
    - authentification.serv... U
    - authentification.serv... U
    - rest-api.service.spec.ts U
    - rest-api.service.ts U
  - > app-routing.module.ts M
- > OUTLINE
- > TIMELINE
- > DEPENDENCIES

A rest-api.service.ts X

src > app > services > A rest-api.service.ts > ...

```
1 import { HttpClient, HttpHeaders } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root',
6 })
7 export class Rest ApiService {
8   // Define API
9   apiURL = 'http://localhost:3000';
10
11 constructor(private http: HttpClient) {}    Property 'http' is declared but its value is never used
12
13 =====
14 | CRUD Methods for consuming RESTful API
15 =====
16
17 // Http Options
18 httpOptions = {
19   headers: new HttpHeaders({
20     'Content-Type': 'application/json',
21   }),
22 }
23
24 }
```

# get ()

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPAN... ⚡ ⚡ ⚡ ⚡

- > dist
- > e2e
- > node\_modules
- > serveur
- ✓ src 67
  - > app
    - > accueil
    - > entete-page
    - > login
    - > modeles
    - > pied-page
    - > pipes
  - > services
    - authentification.serv... U
    - authentification.serv... U
    - rest-api.service.spec.ts U
    - rest-api.service.ts U
  - > app-routing.module.ts M

> OUTLINE

> TIMELINE

> DEPENDENCIES

A rest-api.service.ts X

src > app > services > A rest-api.service.ts > Rest ApiService

```
5
6  @Injectable({
7    providedIn: 'root',
8  })
9  export class Rest ApiService {
10    // Define API
11    apiURL = 'http://localhost:3000';
12
13    constructor(private http: HttpClient) {}
14
15    /*=====
16     * CRUD Methods for consuming RESTful API
17     =====*/
18
19    // Http Options
20    httpOptions = {
21      headers: new HttpHeaders({
22        'Content-Type': 'application/json',
23      }),
24    };
25    // HttpClient API get() methode => Une liste des rendez-vous
26    getRendezVous(): Observable<RendezVous> {
27      return this.http.get<RendezVous>(this.apiURL + '/rv');
28    }
29}
```

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPAN... e2e node\_modules serveur

✓ src app accueil accueil.component.css U accueil.component.h... U accueil.component.s... U accueil.component.ts U

> entete-page login modeles pied-page pipes

✓ services authentification.serv... U authentification.serv... U rest-api service specs U

> OUTLINE

> TIMELINE

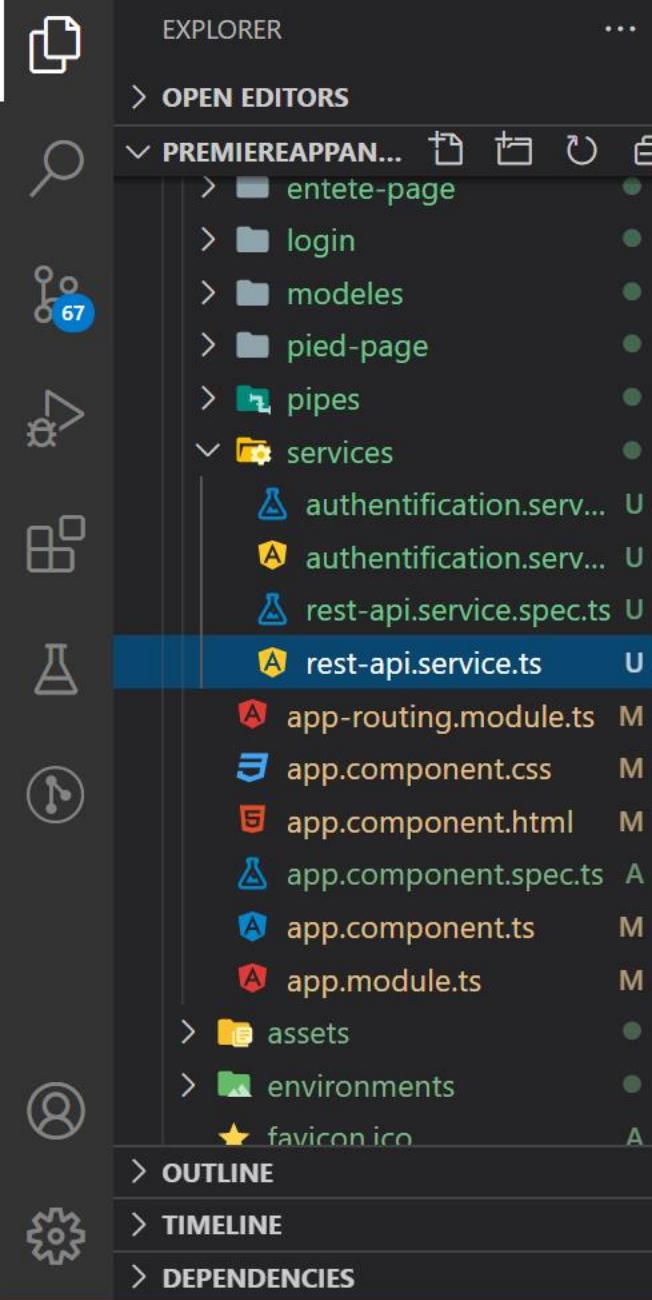
> DEPENDENCIES

rest-api.service.ts accueil.component.ts X

src > app > accueil > accueil.component.ts > AccueilComponent > rendezVous

```
12 export class AccueilComponent implements OnInit {
13   nomUtilisateur = '';
14
15   rendezVous: any = [];
16
17   constructor(
18     private authenticationService: AuthenticationService,
19     private routeActive: ActivatedRoute,
20     private router: Router,
21     private rest ApiService: Rest ApiService
22   ) {}
23
24   chargerLesRvs() {
25     this.rest ApiService.getRendezVous().subscribe(data: {}) => {
26       this.rendezVous = data;
27     });
28   }
29
30   ngOnInit(): void {
31     this.rendezVous = this.chargerLesRvs();
32     if (this.authenticationService.utilisateurConnecte())
33       this.nomUtilisateur = this.routeActive.snapshot.paramMap.get(
34         'nomUtilisateur'
35       );
36     else this.router.navigateByUrl('/login');
37   }
}
```

# delete()



```
rest-api.service.ts X accueil.component.ts
src > app > services > rest-api.service.ts > Rest ApiService > deleteRendezVous
13     constructor(private http: HttpClient) {}
14
15     /**
16      =====
17      CRUD Methods for consuming RESTful API
18      =====
19
20     // Http Options
21     httpOptions = {
22         headers: new HttpHeaders({
23             'Content-Type': 'application/json',
24         }),
25     };
26     // HttpClient API get() methode => Une liste des rendez-vous
27     getRendezVous(): Observable<RendezVous> {
28         return this.http.get<RendezVous>(this.apiURL + '/rv');
29     }
30
31     // HttpClient API delete() methode => Supprimer un rendez-vous par son Id
32     deleteRendezVous(id:number){
33         return this.http.delete<RendezVous>(this.apiURL + '/rv/' + id, this.httpOptions);
34     }
35 }
```

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPAN... 📁 ⏪ ⏴ ⏵ ⏷

- > dist
- > e2e
- > node\_modules
- > serveur
- ✓ src
  - ✓ app
    - ✓ accueil
      - accueil.component.css U
      - accueil.component.h... U
      - accueil.component.s... U
- > entete-page
- > login
- > modeles
- > pied-page
- > pipes
- ✓ services
  - authentification.serv... U

- > OUTLINE
- > TIMELINE
- > DEPENDENCIES

rest-api.service.ts accueil.component.ts ✖ accueil.component.html

src > app > accueil > accueil.component.ts > AccueilComponent > deleteRv

```
23
24     chargerLesRvs() {
25         this.rest ApiService.getRendezVous().subscribe((data: {}) => {
26             this.rendezVous = data;
27         });
28     }
29
30     ngOnInit(): void {
31         this.rendezVous = this.chargerLesRvs();
32         if (this.authenticationService.utilisateurConnecte())
33             this.nomUtilisateur = this.routeActive.snapshot.paramMap.get(
34                 'nomUtilisateur'
35             );
36         else this.router.navigateByUrl('/login');
37     }
38
39     // Delete Rendez-vous
40     deleteRv(id:number) {
41         if (window.confirm('Etes-vous sûr que vous voulez supprimer cet élément ?'))
42             this.rest ApiService.deleteRendezVous(id).subscribe(data => {
43                 this.chargerLesRvs()
44             });
45     }
46 }
47
48 }
```

EXPLORER ... A rest-api.service.ts A accueil.component.ts F accueil.component.html X

> OPEN EDITORS

src > app > accueil > F accueil.component.html

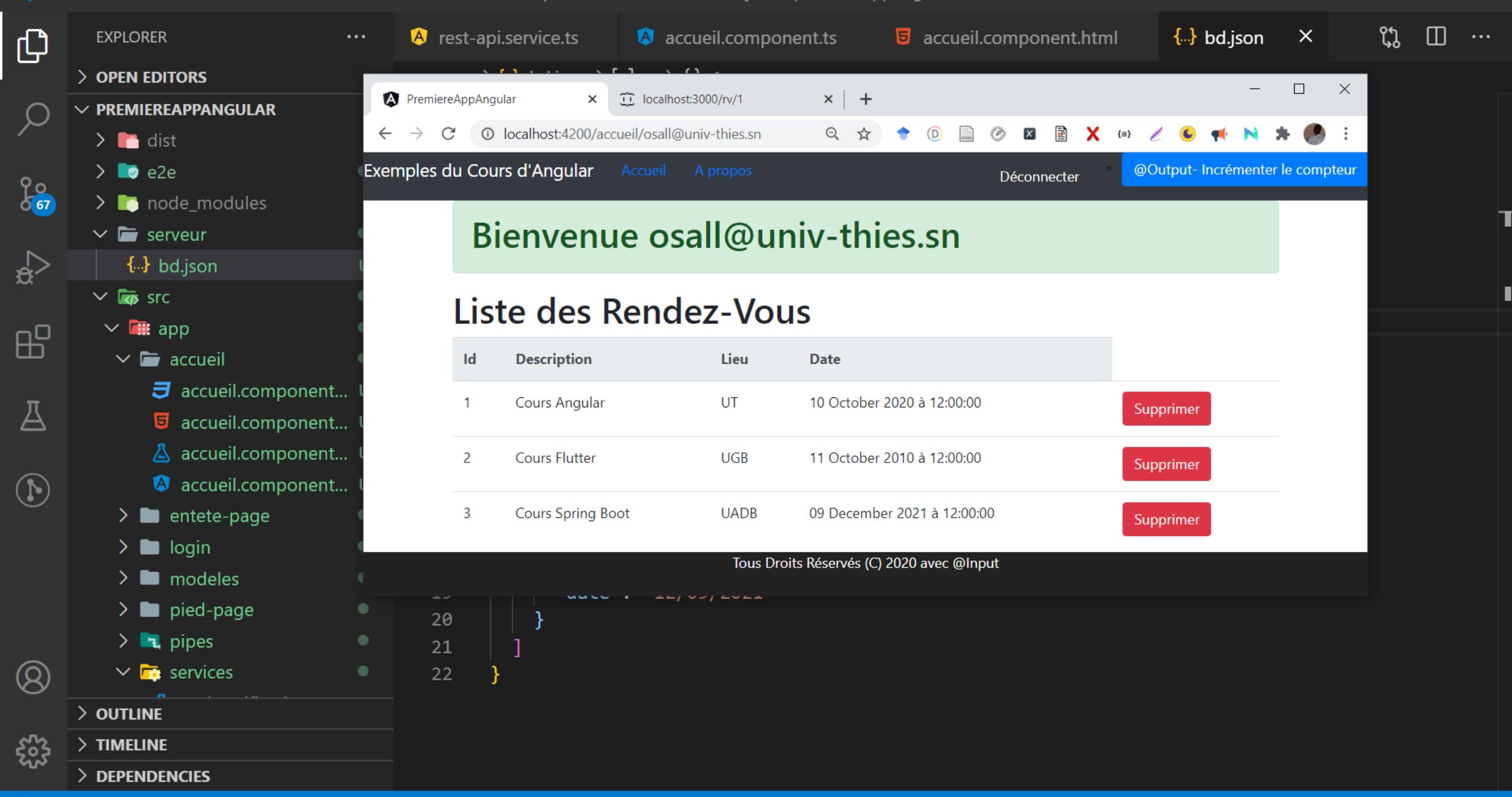
PREMIEREAPPANGULAR 67

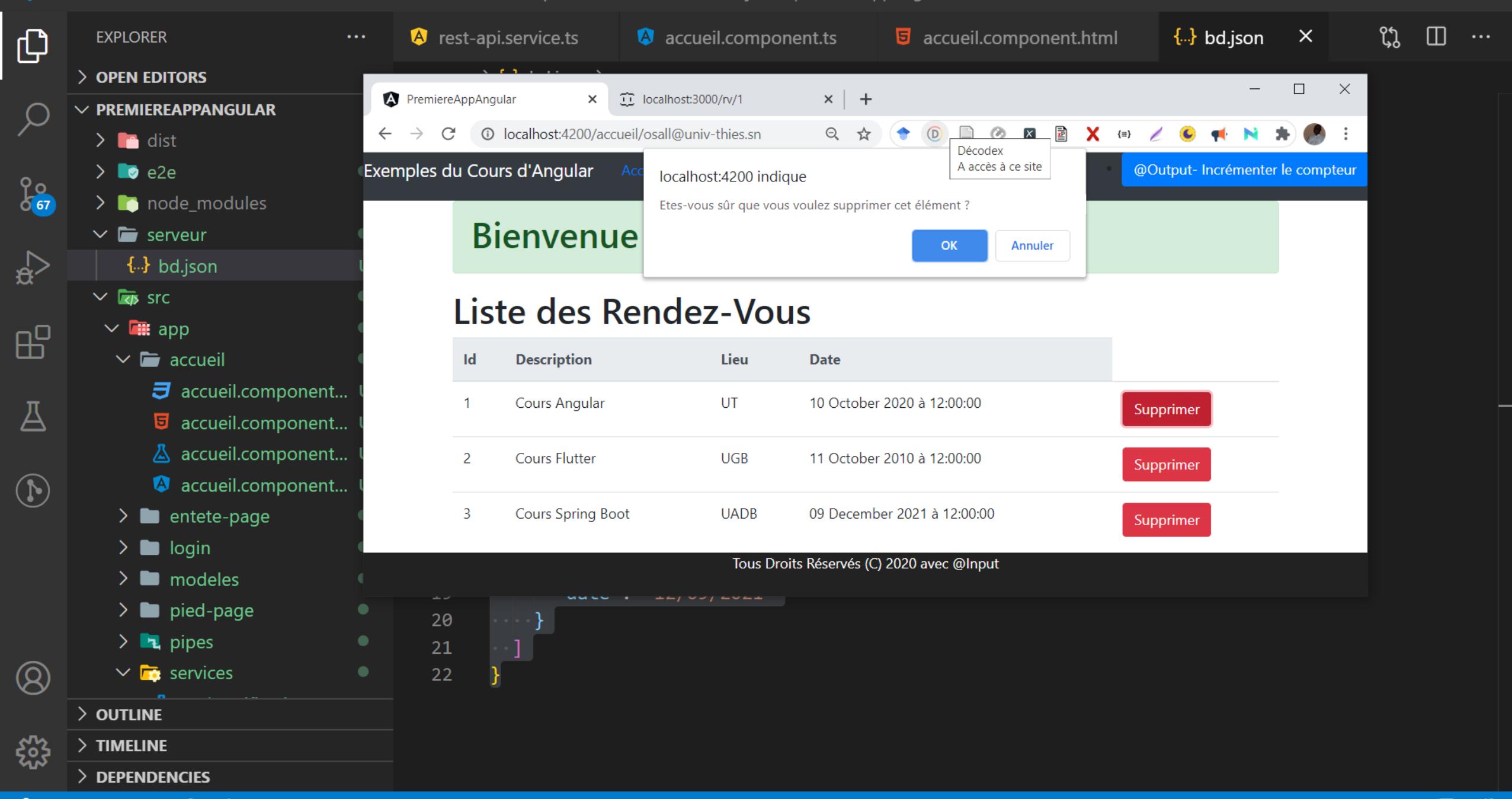
- > dist
- > e2e
- > node\_modules
- > serveur
- > src
  - > app
    - > accueil
      - F accueil.component... U
      - F accueil.component... U
      - A accueil.component... U
      - A accueil.component... U
    - > entete-page
    - > login
    - > modeles
    - > pied-page
    - > pipes
    - > services
      - A authentication.ser... U

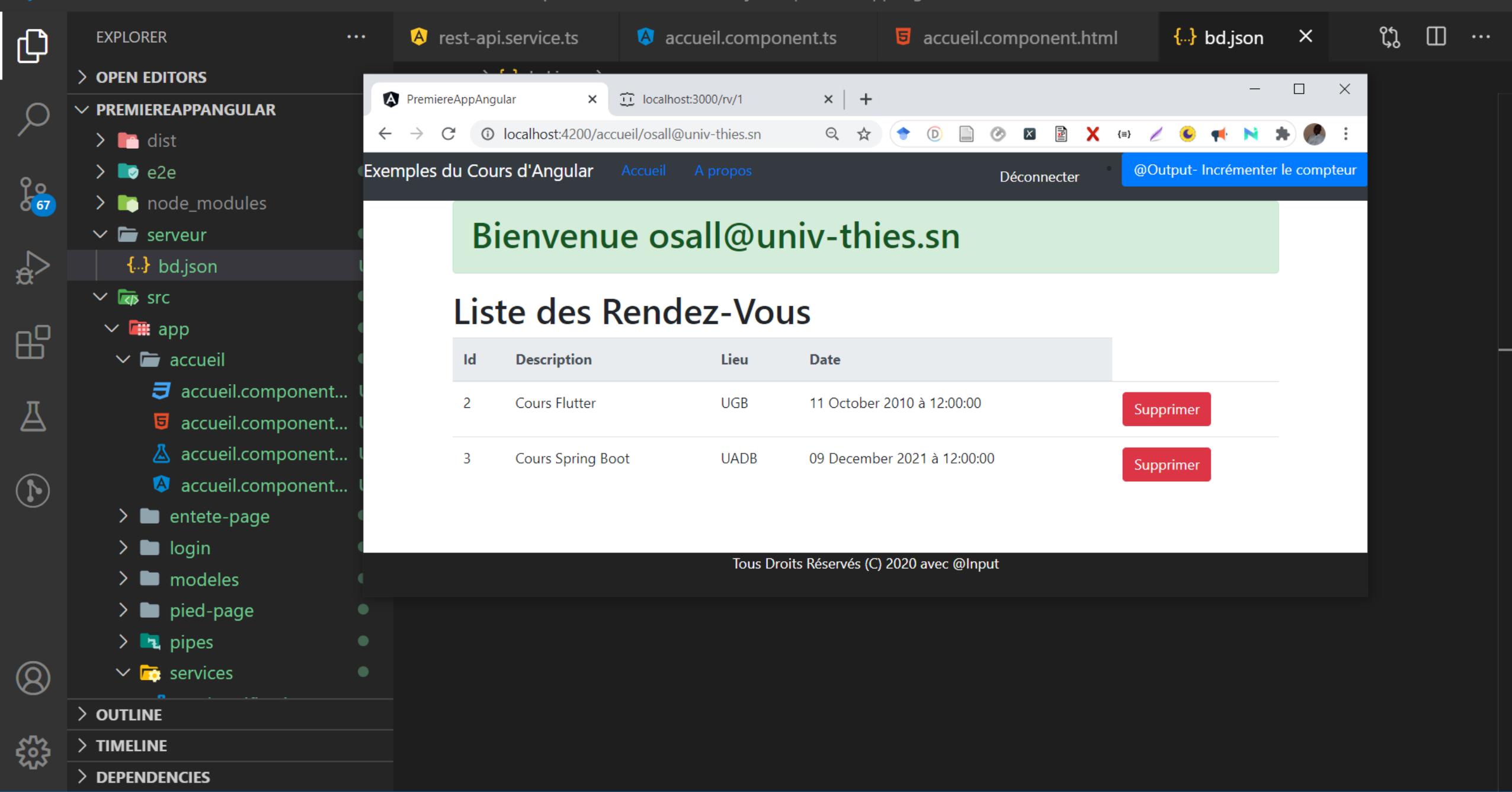
```
6   <table class="table">
7     <thead class="thead-light">
8       <tr>
9         <th>Id</th>
10        <th>Description</th>
11        <th>Lieu</th>
12        <th>Date</th>
13      </tr>
14    </thead>
15    <tbody>
16      <tr *ngFor="let rv of rendezVous">
17        <td>{{rv.id}}</td>
18        <td>{{rv.description}}</td>
19        <td>{{rv.lieu | uppercase | getStringAvantTiret}}</td>
20        <td>{{rv.date | date:'dd MMMM y à hh:mm:ss' }}</td>
21        <td>
22          <button class="btn btn-danger" type="button"(click)="deleteRv(rv.id)">
23            Supprimer
24          </button>
25        </td>
26      </tr>
27    </tbody>
28  </table>
29 </div>
```

master\*+ ↻ 0 △ 0

Compile Hero: Off Ln 22, Col 28 (6 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓







# post ()

EXPLORER ...

> OPEN EDITORS

PREMIEREAPPANGULAR 67

- > entete-page
- > login
- > modeles
- > pied-page
- > pipes
- ✓ services
  - authentification.service.spec.ts
  - A authentification.service.ts
  - rest-api.service.spec.ts
- A rest-api.service.ts
- A app-routing.module.ts
- app.component.css
- S app.component.html
- A app.component.spec.ts
- A app.component.ts
- A app.module.ts

> assets

> environments

★ favicon.ico

> OUTLINE

> TIMELINE

> DEPENDENCIES

rest-api.service.ts X accueil.component.ts accueil.component.html bd.json

src > app > services > rest-api.service.ts > Rest ApiService

```
27 |     return this.http.get<RendezVous>(this.apiURL + '/rv');
28 |
29 |
30 // HttpClient API delete() methode => Supprimer un rendez-vous par son Id
31 deleteRendezVous(id: number) {
32     return this.http.delete<RendezVous>
33     (
34         this.apiURL + '/rv/' + id,
35         this.httpOptions
36     );
37
38 // HttpClient API post() method => Create un rendez-vous
39 createRendezVous(rv: RendezVous): Observable<RendezVous> {
40     return this.http.post<RendezVous>
41     (
42         this.apiURL + '/rv',
43         JSON.stringify(rv),
44         this.httpOptions
45     );
46 }
47 }
```

## A rest-api.service.ts

A accueil.component.ts X

5 accueil.component.htm

{...} bd.json



src > app > accueil >  accueil.component.ts >  AccueilComponent >  rvSaisie

```
8     selector: 'app-accueil',
9     templateUrl: './accueil.component.html',
10    styleUrls: ['./accueil.component.css'],
11  })
12 export class AccueilComponent implements OnInit {
13   nomUtilisateur = '';
```

```
14
15 rvSaisie| = {
16   id: 0,
17   description: "",
18   lieu: "",
19   date: new Date()
20 };
```

```
23     this.restApiService.c  
24     this.chargerLesRvs()  
25   })  
26 }
```

```
26
27
28 rendezVous: any = [ ]
```

29

```
29  
30   constructor(  
31     private authenticationService: AuthenticationService,  
32     private routeActive: ActivatedRoute,  
33     private router: Router,
```



rest-api.service.ts

accueil.component.ts

accueil.component.html X

{..} bd.json

≡ ⌂ ⌂ ...

src > app > accueil > accueil.component.html

```
1  <div class="container">
2      <div class="alert alert-success" role="alert">
3          <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
4      </div>
5
6      <div class="container custom-container">
7          <div class="col-md-12">
8              <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
9              <div class="form-group">
10                 <input type="text" [(ngModel)]="rvSaisie.id" class="form-control" placeholder="Id">
11             </div>
12             <div class="form-group">
13                 <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
14             </div>
15             <div class="form-group">
16                 <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
17             </div>
18             <div class="form-group">
19                 <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
20             </div>
21             <div class="form-group">
22                 <button class="btn btn-success btn-lg btn-block" (click)="addRv()>Ajouter un nouveau RV </button>
23             </div>
24         </div>
25     </div>
```



67



PremiereAppAngular x localhost:3000/rv1 x | +

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémente le compteur

## Bienvenue osall@univ-thies.sn

### Créer un Rendez-Vous

4  
Examen Développement WEB  
UASZ-Zig  
Wed Oct 21 2020 16:49:25 GMT+0000 (heure moyenne de Greenwich)

Ajouter un nouveau RV

### Liste des Rendez-Vous

ID	Description	Lieu	Date	
2	Cours Flutter	UGB	11 October 2010 à 12:00:00	<button>Supprimer</button>
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00	<button>Supprimer</button>

Tous Droits Réservés (C) 2020 avec @Input

PremiereAppAngular x localhost:3000/rv1 x | +

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémente le compteur

## Bienvenue osall@univ-thies.sn

### Créer un Rendez-Vous

4  
Examen Développement WEB  
UASZ-Zig  
Wed Oct 21 2020 16:49:25 GMT+0000 (heure moyenne de Greenwich)

Ajouter un nouveau RV

### Liste des Rendez-Vous

ID	Description	Lieu	Date	
2	Cours Flutter	UGB	11 October 2010 à 12:00:00	<button>Supprimer</button>
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00	<button>Supprimer</button>
4	Examen Développement WEB	UASZ	21 October 2020 à 04:49:25	<button>Supprimer</button>

Tous Droits Réservés (C) 2020 avec @Input

Prof. Ousmane SALL, Univ. Thiès, SN Program

put()

EXPLORER ...

> OPEN EDITORS

PREMIEREAPPANGULAR 67

- > entete-page
- > login
- > modeles
- > pied-page
- > pipes
- < services
  - authentification.service.spec.ts
  - authentification.service.ts
  - rest-api.service.spec.ts
  - rest-api.service.ts
- app-routing.module.ts
- app.component.css
- app.component.html
- app.component.spec.ts
- app.component.ts
- app.module.ts
- > assets
- > environments
- favicon.ico

> OUTLINE

> TIMELINE

> DEPENDENCIES

A rest-api.service.ts X A accueil.component.ts S accueil.component.html {..} bd.json ⌂ ⌂ ...

src > app > services > A rest-api.service.ts > Rest ApiService

```
31 deleteRendezVous(id: number) {
32   return this.http.delete<RendezVous>(
33     this.apiURL + '/rv/' + id,
34     this.httpOptions
35   );
36 }
37
38 // HttpClient API post() method => Create un rendez-vous
39 createRendezVous(rv: RendezVous): Observable<RendezVous> {
40   return this.http.post<RendezVous>(
41     this.apiURL + '/rv',
42     JSON.stringify(rv),
43     this.httpOptions
44   );
45 }
46
47 // HttpClient API put() method => Update employee
48 updateRendezVous(id: number, rv: RendezVous): Observable<RendezVous> {
49   return this.http.put<RendezVous>(
50     this.apiURL + '/rv/' + id,
51     JSON.stringify(rv),
52     this.httpOptions
53   );
54 }
55
56 }
```



rest-api.service.ts

accueil.component.ts

accueil.component.html

bd.json

≡ ⌂ ⌂ ⌂

src > app > accueil > accueil.component.ts > AccueilComponent > update

```
25     .subscribe(data: {}) => {    'data' is declared but its value is never read.
26       this.chargerLesRvs();
27     );
28   }
29
30   rendezVous: any = [];
31
32   update=false;
33   // Update Rendez-Vous
34   updateRv(rv: RendezVous) {
35     this.rvSaisie = rv;
36     this.update=true;
37 } // Update Rendez-Vous
38
39   mettreAJour() {
40     if(window.confirm('Êtes-vous sûr de vouloir mettre à jour?')){
41       this.rest ApiService.updateRendezVous(this.rvSaisie.id, this.rvSaisie).subscribe(data => {    'data' is declared but it
42
43     })
44   }
45   this.update=false;
46   this.chargerLesRvs();
47 }
48
49   constructor(
50     private authenticationService: AuthenticationService.
```

rest-api.service.ts

accueil.component.ts

accueil.component.html X

bd.json



```
src > app > accueil > accueil.component.html
30   <h1>Liste des Rendez-Vous</h1>
31   <table class="table">
32     <thead class="thead-light">
33       <tr>
34         <th>Id</th>
35         <th>Description</th>
36         <th>Lieu</th>
37         <th>Date</th>
38       </tr>
39     </thead>
40     <tbody>
41       <tr *ngFor="let rv of rendezVous">
42         <td>{{rv.id}}</td>
43         <td>{{rv.description}}</td>
44         <td>{{rv.lieu | uppercase | getStringAvantTiret}}</td>
45         <td>{{rv.date | date:'dd MMMM y à hh:mm:ss' }}</td>
46         <td>
47           <button class="btn btn-success" (click)="updateRv(rv)">Editer </button>
48           <button class="btn btn-danger" type="button" (click)="deleteRv(rv.id)">
49             Supprimer
50           </button>
51         </td>
52       </tr>
53     </tbody>
54   </table>
55 </div>
```



rest-api.service.ts

accueil.component.ts

accueil.component.html X

{..} bd.json

≡ ⌂ ⌂ ...

src &gt; app &gt; accueil &gt; accueil.component.html

```
5
6      <div class="container custom-container">
7          <div class="col-md-12">
8              <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
9              <div class="form-group">
10                 <input type="text" [(ngModel)]="rvSaisie.id" class="form-control" placeholder="Id">
11             </div>
12             <div class="form-group">
13                 <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
14             </div>
15             <div class="form-group">
16                 <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
17             </div>
18             <div class="form-group">
19                 <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
20             </div>
21             <div class="form-group" *ngIf="!update">
22                 <button class="btn btn-success btn-lg btn-block" (click)="addRv()">Ajouter un nouveau RV </button>
23             </div>
24             <div class="form-group" *ngIf="update">
25                 <button class="btn btn-success btn-lg btn-block" (click)="mettreAjour()">Mettre à jour un nouveau RV </button>
26             </div>
27         </div>
28     </div>
29
30     <h1>Liste des Rendez-Vous</h1>
```



67



A PremiereAppAngular x localhost:3000/rv/1 x | +

localhost:4200/accueil/osall@... 🔍 ⭐ ⚡ 📁 🗂️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

0

Description

Lieu

Wed Oct 21 2020 17:22:26 GMT+0000 (heure moyenne de Greenwich)

Ajouter un nouveau RV

## Liste des Rendez-Vous

Id	Description	Lieu	Date	
2	Cours Flutter	UGB	11 October 2010 à 12:00:00	<button>Editer</button> <button>Supprimer</button>
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00	<button>Editer</button> <button>Supprimer</button>
4	Examen Développement WEB	UASZ	21 October 2020 à 04:49:25	<button>Editer</button> <button>Supprimer</button>

Prof. Ousmane SALL, Univ

Tous Droits Réservés (C) 2020 avec @Input

421

A PremiereAppAngular x localhost:3000/rv/1 x | +

localhost:4200/accueil/osall@... 🔍 ⭐ ↕ 📁 🗂️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

2

Cours Flutter Sénégal

UGB-Saint-Louis

10/11/2010

Mettre à jour un nouveau RV

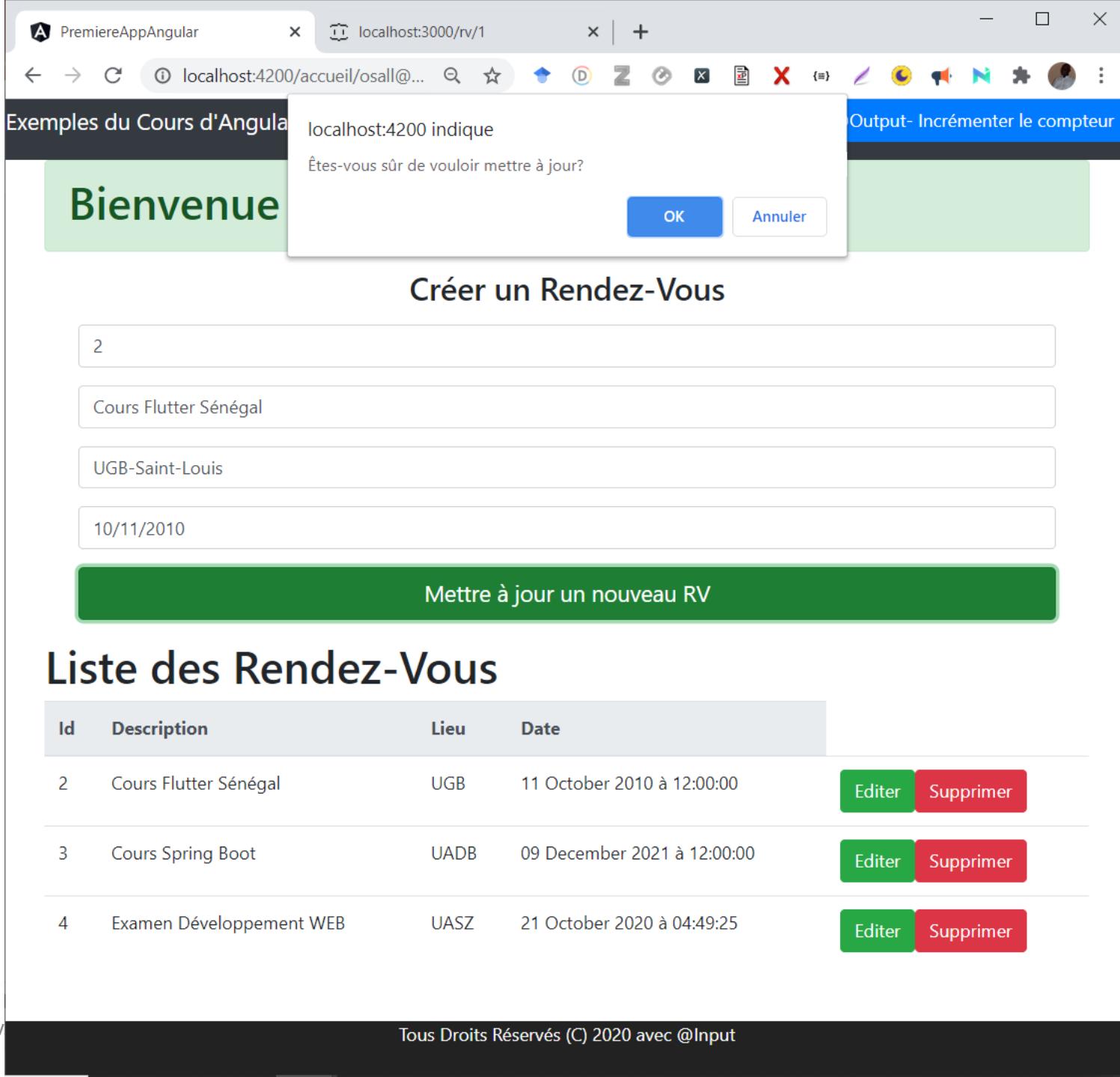
## Liste des Rendez-Vous

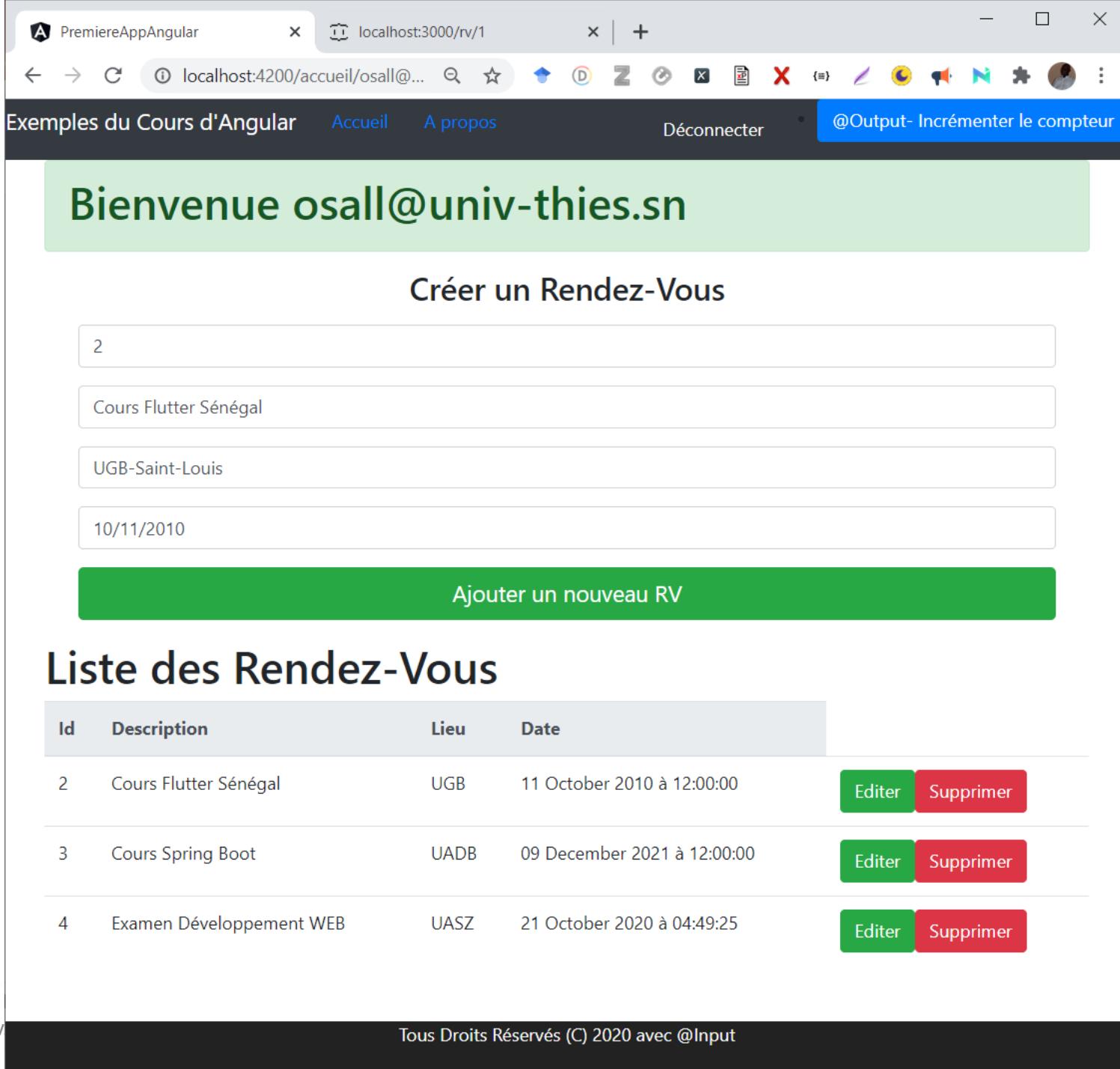
Id	Description	Lieu	Date	
2	Cours Flutter Sénégal	UGB	11 October 2010 à 12:00:00	<a href="#">Editer</a> <a href="#">Supprimer</a>
3	Cours Spring Boot	UADB	09 December 2021 à 12:00:00	<a href="#">Editer</a> <a href="#">Supprimer</a>
4	Examen Développement WEB	UASZ	21 October 2020 à 04:49:25	<a href="#">Editer</a> <a href="#">Supprimer</a>

Prof. Ousmane SALL, Univ

Tous Droits Réservés (C) 2020 avec @Input

422





EXPLORER

> OPEN EDITORS

✓ PREMIEREAPPANGULAR

- > dist
- > e2e
- > node\_modules
- ✓ serveur
  - {...} bd.json U
  - ✓ src
    - ✓ app
      - ✓ accueil
        - accueil.component.css U
        - accueil.component.html U
        - accueil.component.spec.ts U
        - accueil.component.ts U
      - entete-page
      - login
      - modeles
      - pied-page
      - pipes
    - ✓ services
      - authentification service specs U

> OUTLINE

> TIMELINE

> DEPENDENCIES

serveur > {...} bd.json > [ ] rv > {} 0 > abc description

```
1 {  
2   "rv": [  
3     {  
4       "id": 2,  
5       "description": "Cours Flutter Sénégal",  
6       "lieu": "UGB-Saint-Louis ",  
7       "date": "10/11/2010"  
8     },  
9     {  
10       "id": 3,  
11       "description": "Cours Spring Boot",  
12       "lieu": "UADB-Bambey",  
13       "date": "12/09/2021"  
14     },  
15     {  
16       "id": 4,  
17       "description": "Examen Développement WEB",  
18       "lieu": "UASZ-Zig",  
19       "date": "2020-10-21T16:49:25.409Z"  
20     }  
21   ]  
22 }
```

# Travaux Pratiques n°7

<https://developers.google.com/codelabs/angular-slider-element?hl=en>



# Angular Material

<https://www.youtube.com/watch?v=dcACk0WTqNc>

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- **Angular Material**
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Travaux Pratiques n°8

<https://codelab.fun/angular/material/intro>



# Les Formulaires

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- **Formulaires**
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Les formulaires

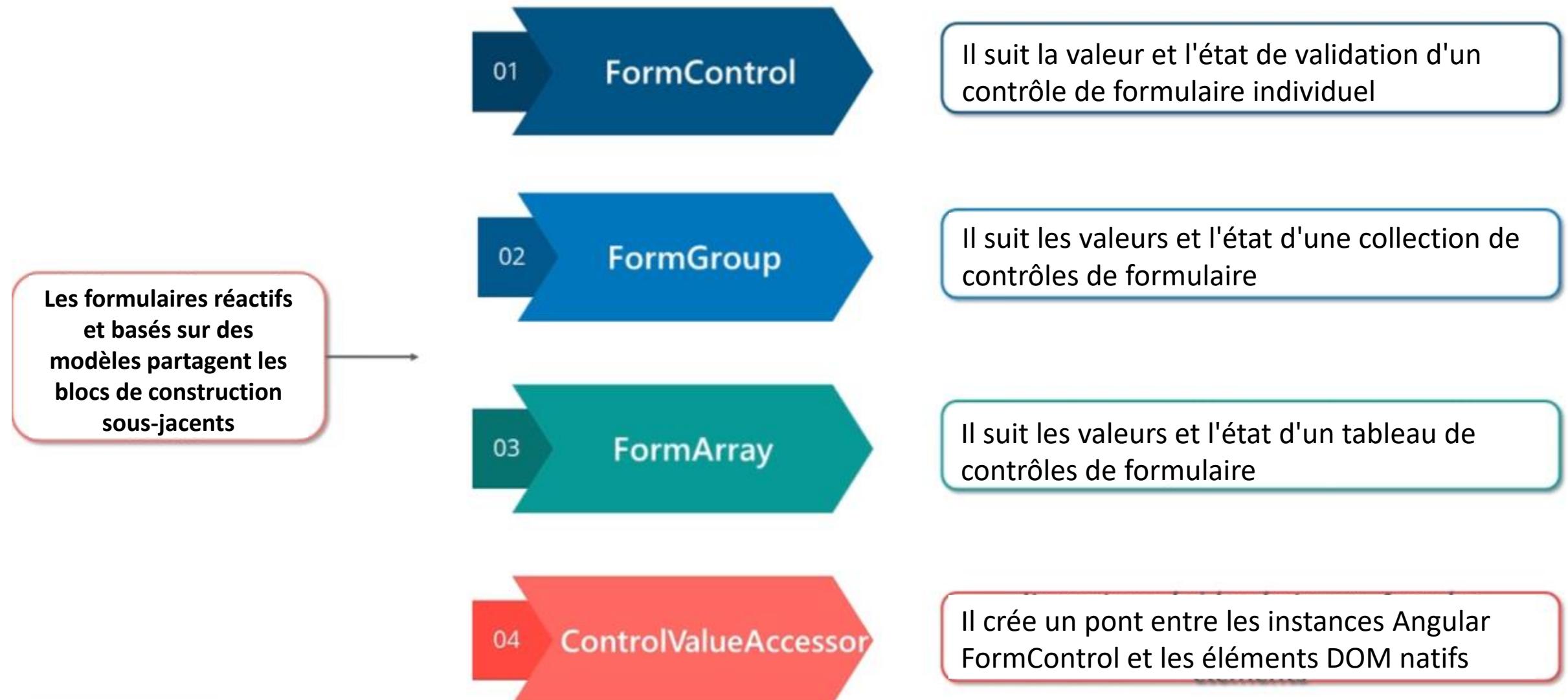
- Un outil graphique que nous créons avec le langage de description HTML
- Permet à l'utilisateur de saisir des données et de les envoyer vers une autre page, vers une base de données...
- Angular nous facilite la récupération des données saisies, la validation et le contrôle des valeurs saisies, la gestion d'erreurs...



# Formulaire Angular

- Un moyen d'interaction avec l'utilisateur pour manipuler les données de l'application: **Ajout, modification de données métiers**
- Dans Angular, il y a deux types de formulaires :
  - **Template Forms** : vous créez votre formulaire dans le template, et Angular l'analyse pour comprendre les différents inputs et pour mettre à disposition le contenu
  - **Reactive Forms**: vous créez votre formulaire en TypeScript et dans le template, puis vous en faites la liaison manuellement — cette approche est plus complexe, mais elle permet d'avoir beaucoup plus de contrôle et une approche dynamique.

# Formulaire Angular



# Template Driven Form Approach

- Asynchrone
- Liez des directives pour laisser Angular faire le gros du travail. Liaison de données via **ngModel**
- Utilise le **FormsModule**  
**import {FormsModule} depuis '@angular/forms';**
- Les liaisons et la validation existent dans le modèle
- Le modèle peut rapidement devenir difficile à lire avec des conditions et des validations complexes

EXPLORER

> OPEN EDITORS

PREMIEREAPPANG... 71

- ... bd.json
- src
  - app
    - accueil
    - entete-page
    - error
    - login
    - modeles
    - pied-page
    - pipes
  - services
    - app-routing.module.ts
    - app.component.css
    - app.component.html
    - app.component.spec.ts
    - app.component.ts
  - assets
- app.module.ts

> OUTLINE

> TIMELINE

> DEPENDENCIES

app.module.ts

src > app > app.module.ts > AppModule

```
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { AccueilComponent } from './accueil/accueil.component';
7 import { LoginComponent } from './login/login.component';
8 import { FormsModule } from '@angular/forms';
9 import { AuthenticationService } from './services/authentication.service';
10 import { GetStringAvantTiretPipe } from './pipes/get-string-avant-tiret.pipe';
11 import { EntetePageComponent } from './entete-page/entete-page.component';
12 import { PiedPageComponent } from './pied-page/pied-page.component';
13 import { HttpClientModule } from '@angular/common/http';
14 import { ErrorComponent } from './error/error.component';

15
16 @NgModule({
17   declarations: [AppComponent, AccueilComponent, LoginComponent, GetStringAvantTiretPipe, EntetePageComponent, PiedPageComponent],
18   imports: [BrowserModule, AppRoutingModule, FormsModule, HttpClientModule],
19   providers: [AuthenticationService],
20   bootstrap: [AppComponent],
21 })
22 export class AppModule {}
```

EXPLORER ...

> OPEN EDITORS

PREMIE... bd.json U

src app login

- ... bd.json U
- src 71
  - app
    - accueil
    - entete-page
    - error
  - login
    - login.comp... U
    - login.comp... U
    - login.comp... U
    - login.comp... U
  - modeles
  - pied-page
  - pipes
  - services
  - app-routing.... M
  - app.compon... M
  - app.compon... M

> OUTLINE

> TIMELINE

> DEPENDENCIES

login.component.html X

src > app > login > login.component.html

```
1  <div class="container">
2    <span [ngSwitch]="!parametresInvalides">
3      <div *ngSwitchCase="true" class="alert alert-success" role="alert">
4        Bienvenue {{nomUtilisateur}}
5      </div>
6      <div *ngSwitchCase="false" class="alert alert-danger" role="alert">
7        Vérifier les paramètres saisis, elles ne sont pas les bonnes.
8      </div>
9      <div *ngSwitchDefault>
10     </div>
11   </span>
12
13
14  <div class="col col-md-4">
15    <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
16    <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse"> <br>
17    <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
18  </div>
19 </div>
```



```
5 login.component.html X
src > app > login > login.component.html > div.container
8
9     </div>
10    <div *ngSwitchDefault>
11        </div>
12    </span>
13
14    {{loginForm.value | json}}
15    <div class="col col-md-4">
16        <form #loginForm="ngForm">
17            <input class="form-control" type="text" name="username"/>
18            <input class="form-control" type="password" name="password"/>
19            <button class="btn btn-primary" type="submit" value="Connexion">
20        </form>
21    </div>
22</div> : Angular associe à chaque fo
```

Vérifier les paramètres saisis, elles ne sont pas les bonnes.

```
{"login": "osall751@univ-thies.sn", "password": "passer123"}
```

osall751@univ-thies.sn

.....

Se Connecter

- Angular associe à chaque formulaire une directive **ngForm** pour suivre son état:
  - Les valeurs des contrôles du formulaire ?
  - Les valeurs sont-elles valides ou pas ?
- `#loginForm="ngForm"` exporte la directive dans une variable de modèle locale en utilisant **ngForm** comme clé
- `{{loginForm.value | json}}` affiche les valeurs saisies au format json

File Edit Selection View Go Run Terminal Help login

login.component.html X

src > app > login > login.component.html > div.container

```
8     </div>
9     <div *ngSwitchDefault>
10    </div>
11    </span>
12
13
14 {{loginForm.value | json}}
15 <div class="col col-md-4">
16   <form #loginForm="ngForm">
17     <input class="form-control" type="text" name="login" [(ngModel)]="nomUtilisateur"> <br>
18     <input class="form-control" type="password" name="password" [(ngModel)]="motDePasse"><br>
19     <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
20   </form>
21 </div>
22 </div>
```

PremiereAppAngular localhost:4200 Exemples du Cours d'Angular Se Connecter @Output- Incrémenter le compteur

Vérifier les paramètres saisis, elles ne sont pas les bonnes.

{ "login": "osall751@univ-thies.sn", "password": "passer123" }

osall751@univ-thies.sn

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

La directive **ngModel** permet d'identifier les contrôles à suivres et il faut obligatoirement un attribut **name** à pour chaque contrôle

master\*+ ⌛ ⚡ 0 △ 0 ⚡ Compile Hero: Off Ln 14, Col 1 Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier ⌛ ⚡



login.component.html

```
src > app > login > login.component.html > div.container
  8      </div>
  9      <div *ngSwitchDefault>
10        </div>
11      </span>
12
13
14    {{loginForm.value | json}}
15    <div class="col col-md-4">
16      <form #loginForm="ngForm">
17        <input class="form-control" type="text" name="login" ngModel> <br>
18        <input class="form-control" type="password" name="password" ngModel><br>
19        <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
20      </form>
21    </div>
22  </div>
```

Vérifier les paramètres saisis, elles ne sont pas les bonnes.

```
{ "login": "", "password": "" }
```

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

Alternative sans binding du composant vers le template:  
La directive **ngModel** permet d'identifier les contrôles à suivres et il  
faut obligatoirement un attribut **name** à pour chaque contrôle



File Edit Selection View Go Run Terminal Help login.c

login.component.html X

```
src > app > login > login.component.html > div.container >
  9
 10
 11      </div>
 12    </span>
 13
 14    {{loginForm.value | json}}
 15    <div class="col col-md-4">
 16      <form #loginForm="ngForm">
 17        <input class="form-control" type="text" name="login" ngModel> <br>
 18        <input class="form-control" type="password" name="password" ngModel><br>
 19        <div class="form-group" ngModelGroup="test" >
 20          <input class="form-control" type="text" name="test1" ngModel>
 21          <input class="form-control" type="text" name="test2" ngModel>
 22        </div>
 23        <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
 24      </form>
 25    </div>
 26  </div>
```

PremiereAppAngular localhost:4200/login Exemples du Cours d'Angular Se Connecter @Output- Incrémenter le compte Vérifier les paramètres saisies, elles ne sont pas les bonnes. { "login": "osall@univ-thies.sn", "password": "passer123", "test": [ "test1": "Valtest1", "test2": "Valtest2" ] } osall@univ-thies.sn ..... Valtest1 Valtest2 Se Connecter Tous Droits Réservés (C) 2020 avec @Input

La directive **ngModelGroup** permet de regrouper un ensemble de contrôles dans un seul objet

master\*+ ↕ ⊗ 0 △ 0 Ln 19, Col 49 (12 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier

EXPLORER ...

> OPEN EDITORS

✓ PREMIEREAPPANGULAR

- > error
- > login
- > modeles
  - rendez-vous.spec.ts
  - rendez-vous.ts
  - utilisateur.spec.ts
  - utilisateur.ts
- > pied-page
- > pipes
- > services
- app-routing.module.ts M
- app.component.css M
- app.component.html M
- app.component.spec.... A
- app.component.ts M
- app.module.ts M
- > assets
- > environments

> OUTLINE

> TIMELINE

> DEPENDENCIES

## utilisateur.ts X

```
src > app > modeles > utilisateur.ts > Utilisateur
1  export class Utilisateur {
2    constructor(private _login:string, private _motDePasse:string){}
3
4    public get login(): string {
5      return this._login;
6    }
7    public get motDePasse(): string {
8      return this._motDePasse;
9    }
10   public set login(v : string) {
11     this._login = v;
12   }
13   public set motDePasse(v : string) {
14     this._motDePasse = v;
15   }
16 }
17
```

Nous allons maintenant générer dans le dossier modeles une classe des utilisateurs afin de faire le binding de ses attributs avec les contrôles du formulaire

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> cd .\src\app\modeles\
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\modeles> ng g cl utilisateur
CREATE src/app/modeles/utilisateur.spec.ts (174 bytes)
CREATE src/app/modeles/utilisateur.ts (29 bytes)
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\modeles>
```



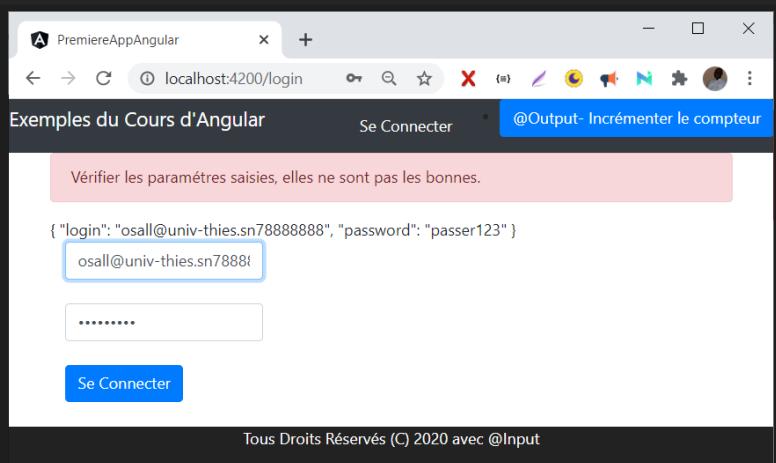
## A login.component.ts X

```
src > app > login > A login.component.ts > LoginComponent > ⚒ utilisateur
10    })
11    export class LoginComponent implements OnInit {
12      // nomUtilisateur = 'osall@univ-thies.sn';
13      // motDePasse = 'passer123';
14      utilisateur=new Utilisateur('osall@univ-thies.sn', 'passer123');
15      parametresInvalides = true;
16
17      constructor(private router:Router, private authentificationService:AuthentificationService) {}
18
19      ngOnInit(): void {}
20
21      valider() {
22        if(this.authentificationService.seLoguer(this.utilisateur.login, this.utilisateur.motDePasse)){
23          this.router.navigate(["accueil", this.utilisateur.login]);
24          this.parametresInvalides = false;
25        }else{
26          this.parametresInvalides = true;
27        }
28      }
29    }
```

login.component.ts

login.component.html X

```
src > app > login > login.component.html
1   <div class="container">
2       <span [ngSwitch]="!parametresInvalides">
3           <div *ngSwitchCase="true" class="alert alert-success" role="alert">
4               Bienvenue {{utilisateur.login}}
5           </div>
6           <div *ngSwitchCase="false" class="alert alert-danger" role="alert">
7               Vérifier les paramètres saisies, elles ne sont pas les bonnes.
8           </div>
9           <div *ngSwitchDefault>
10
11      </div>
12  </span>
13
14  {{loginForm.value | json}}
15  <div class="col col-md-4">
16      <form #loginForm="ngForm">
17          <input class="form-control" type="text" name="login" [(ngModel)]="utilisateur.login"> <br>
18          <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
19          <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>
20
21  </div>
22 </div>
```



# Suivi de l'état et de la validité du formulaire

- En plus du binding dans les deux sens, **ngModel** permet de savoir
  - Si l'utilisateur a touché un contrôle
  - Si la valeur a changé
  - Si la valeur saisie est valide
- Elle met à jour alors le contrôle avec des classe CSS prédéfinies qui reflètent son état
  - **form-control** : classe Bootstrap qu'on a ajoutée
  - **ng-untouched** : classe Angular appliquée quand le champ n'est pas touché (son inverse est **ng-touched**)
  - **ng-pristine** : classe Angular appliquée quand le champ est vide (son inverse est **ng-dirty**)
  - **ng-invalid** : classe Angular appliquée quand le champ n'est pas valid (son inverse est **ng-valid**)

File Edit Selection View Go Run Terminal Help login.component.html - premiereAppAngular - Visual Studio Code

login.component.ts login.component.html

PremiereAppAngular localhost:4200/login

Exemples du Cours d'Angular Se Connecter @Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

```
{ "login": "osall@univ-thies.sn", "password": "passer123" }  
form-control ng-untouched ng-pristine ng-valid  
osall@univ-thies.sn  
.....  
Se Connecter
```

Tous Droits Réservés (C) 2020 avec @Input

```
15  
16 {{loginForm.value | json}}  
17 <br>  
18 {{inputLogin.className}}  
19 <div class="col col-md-4">  
20 <form #loginForm="ngForm">  
21 <input class="form-control" #inputLogin type="text" name="login" [(ngModel)]="utilisateur.login" > <br>  
22 <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse" ><br>  
23 <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>  
24 </form>  
25 </div>  
26 </div>
```

#inputLogin est une variable locale contenant les attributs de cet élément HTML. Les variables locales peuvent être utilisées avec tout élément HTML, <p>, <h>...

Compile Hero: Off Ln 18, Col 17 (10 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier

File Edit Selection View Go Run Terminal Help

login.component.html - premiereAppAngular - Visual Studio Code

login.component.ts login.component.html

src > app > login > login.component.html > div.container > div.col

```
9     <div *ngSwitchDefault>
10
11         </div>
12     </span>
13
14
15
16     {{loginForm.value | json}}
17     <br>
18     Validité de la saisie: {{inputLogin.valid}}
19     <div class="col col-md-4">
20         <form #loginForm="ngForm">
21             <div class="form-group">
22                 <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"
23                     pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>
24             <div *ngIf="inputLogin.touched">
25                 <small class="text-danger" *ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>
26                 <small class="text-danger" *ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>
27             </div>
28         </div>
29         <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
30         <button class="btn btn-primary" type="button" (click)="valider()>Se Connecter</button>
31
32     </form>
33 </div>
```

PremiereAppAngular localhost:4200

Exemples du Cours d'Angular Se Connecter @Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "osall@univ-thies.sn", "password": "passer123" }

Validité de la saisie: true

osall@univ-thies.sn

.....

Se Connecter

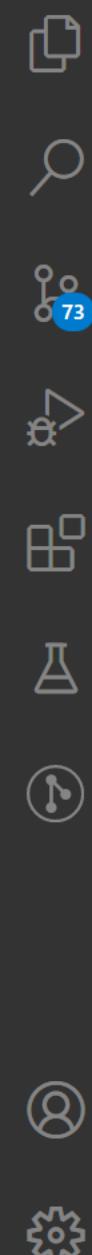
Tous Droits Réservés (C) 2020 avec @Input

Pour obtenir la référence du ngModel

Le message sera affiché si l'utilisateur touche la zone de saisie et si elle est vide ou le texte saisie ne colle pas au pattern de validation de l'email

master\*+ 0 △ 0

Compile Hero: Off Ln 22, Col 82 (10 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier



login.component.ts

login.component.html X

```
src > app > login > login.component.html > div.container > div.col.col-md-6 > div > div *ngSwitchDefault>
  9      <div *ngSwitchDefault>
10        ...
11      </div>
12    </span>
13
14
15
16    {{loginForm.value | json}}
17    <br>
18    Validité de la saisie: {{inputLogin.valid}}
19    <div class="col col-md-4">
20      <form #loginForm="ngForm">
21        <div class="form-group">
22          <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"
23          pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>
24          <div *ngIf="inputLogin.touched">
25            <small class="text-danger" *ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>
26            <small class="text-danger" *ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>
27          </div>
28        </div>
29        <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
30        <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>
31      </form>
32    </div>
33  </div>
```

PremiereAppAngular



&lt; &gt; C ⓘ localhost:4200



Exemples du Cours d'Angular

Se Connecter

@Output- Incrémenteur le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "osall@univ-thies", "password": "passer123" }

Validité de la saisie: false

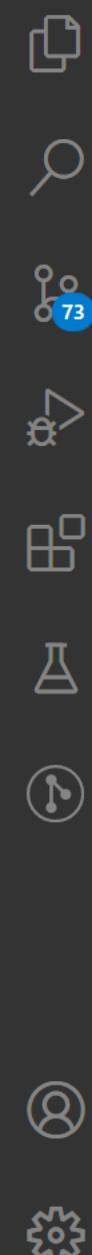
osall@univ-thies

L'email saisie n'est pas valide !

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input



login.component.ts

login.component.html X

```
src > app > login > login.component.html > div.container > div.col.col-md-6 > div > div *ngSwitchDefault>
  9      <div *ngSwitchDefault>
10        </div>
11      </span>
12
13
14
15
16      {{loginForm.value | json}}
17      <br>
18      Validité de la saisie: {{inputLogin.valid}}
19      <div class="col col-md-4">
20        <form #loginForm="ngForm">
21          <div class="form-group">
22            <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"
23              pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>
24            <div *ngIf="inputLogin.touched">
25              <small class="text-danger" *ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>
26              <small class="text-danger" *ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>
27            </div>
28          </div>
29          <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
30          <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>
31        </form>
32      </div>
33    </div>
```

PremiereAppAngular



&lt; → C ⓘ localhost:4200



Exemples du Cours d'Angular

Se Connecter

@Output- Incrémenteur le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "osall", "password": "passer123" }

Validité de la saisie: false

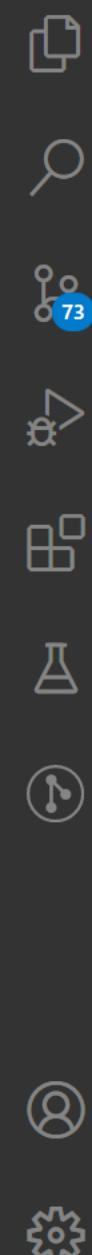
osall

L'email saisie n'est pas valide !

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input



login.component.ts

login.component.html X

```
src > app > login > login.component.html > div.container > div.col.col-md-6 > div > div *ngSwitchDefault>
  9      <div *ngSwitchDefault>
10        </div>
11      </span>
12
13
14
15
16    {{loginForm.value | json}}
17    <br>
18    Validité de la saisie: {{inputLogin.valid}}
19    <div class="col col-md-4">
20      <form #loginForm="ngForm">
21        <div class="form-group">
22          <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"
23            pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>
24          <div *ngIf="inputLogin.touched">
25            <small class="text-danger" *ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>
26            <small class="text-danger" *ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>
27          </div>
28        </div>
29        <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
30        <button class="btn btn-primary" type="button" (click)="valider()">Se Connecter</button>
31      </form>
32    </div>
33  </div>
```

PremiereAppAngular

localhost:4200

Exemples du Cours d'Angular

Se Connecter

@Output- Incrémenteur le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "", "password": "passer123" }

Validité de la saisie: false

Le login ne peut pas être vide !

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

File Edit Selection View Go Run Terminal Help

login.component.ts login.component.html

src > app > login > login.component.html > div.container > div.col

9           <div \*ngSwitchDefault>

10            

## Validation du formulaire avec activation ou désactivation du bouton de validation du formulaire selon l'état du formulaire

15           {{loginForm.value | json}}

16           <br>

17           Validité de la saisie: {{inputLogin.valid}}

18           <div class="col col-md-4">

19            <form #loginForm="ngForm">

20            <div class="form-group">

21            <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"

22            | pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}\$" required>

23            <div \*ngIf="inputLogin.touched">

24            | <small class="text-danger" \*ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>

25            | <small class="text-danger" \*ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>

26            |>

27            </div>

28            </div>

29            <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>

30            <button class="btn btn-primary" type="button" (click)="valider()" [disabled]="loginForm.form.invalid">

31            | Se Connecter

32            |>

33            </button>

34            </form>

35          </div>

PremiereAppAngular

localhost:4200

Exemples du Cours d'Angular

Se Connecter

@Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "osall@univ-thies.sn", "password": "passer123" }

Validité de la saisie: true

osall@univ-thies.sn

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

Compile Hero: Off Ln 20, Col 25 (9 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier

File Edit Selection View Go Run Terminal Help

login.component.ts login.component.html

src > app > login > login.component.html > div.container > div.col

9           <div \*ngSwitchDefault>

10              

## Validation du formulaire avec activation ou désactivation du bouton de validation du formulaire selon l'état du formulaire

15              {{loginForm.value | json}}

16              <br>

17              Validité de la saisie: {{inputLogin.valid}}

18              <div class="col col-md-4">

19                  <form #loginForm="ngForm">

20                      <div class="form-group">

21                          <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"

22                          pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}\$" required>

23                          <div \*ngIf="inputLogin.touched">

24                              <small class="text-danger" \*ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>

25                              <small class="text-danger" \*ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>

26                          </div>

27                      </div>

28                  <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>

29                  <button class="btn btn-primary" type="button" (click)="valider()" [disabled]="loginForm.form.invalid">

30                      Se Connecter

31                  </button>

32              </div>

33              </form>

34      </div>

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

{ "login": "osall@univ-thies", "password": "passer123" }

Validité de la saisie: false

osall@univ-thies

L'email saisie n'est pas valide !

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

Compile Hero: Off Ln 20, Col 25 (9 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier



login.component.ts

login.component.html X



src &gt; app &gt; login &gt; login.component.html &gt; div.container &gt; div.col.col-md-4 &gt; form

```
 9      <div *ngSwitchDefault>
10        </div>          Soumettre le formulaire si valide: le bouton de validation
11      </span>          doit être de type submit et utiliser la directive ngSubmit
12      </div>          dans le tag form
13
14
15
16    {{loginForm.value | json}}
17    <br>
18    Validité de la saisie: {{inputLogin.valid}}
19    <div class="col col-md-4">
20      <form #loginForm="ngForm" [(ngSubmit)]="valider()">
21        <div class="form-group">
22          <input class="form-control" type="email" name="login" #inputLogin=ngModel [(ngModel)]="utilisateur.login"
23          | pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>
24          <div *ngIf="inputLogin.touched">
25            <small class="text-danger" *ngIf="inputLogin.errors?.required">Le login ne peut pas être vide !</small>
26            <small class="text-danger" *ngIf="inputLogin.errors?.pattern">L'email saisie n'est pas valide !</small>
27          </div>
28        </div>
29        <input class="form-control" type="password" name="password" [(ngModel)]="utilisateur.motDePasse"><br>
30        <button class="btn btn-primary" [type="submit"] [disabled]="loginForm.form.invalid">
31          Se Connecter
32        </button>
33      </form>
34    </div>
```

# Reactive Form

- La grande partie du code réside dans la classe du composant.
- Adapté aux scénarios complexes.
- Des contrôles de formulaires dynamiqueValidation dynamique
- Test unitaire

# Reactive Form

- Synchrone (flux d'événements plus prévisible)
- Utilise le **ReactiveFormsModule**  
**Import {ReactiveFormsModule} depuis '@angular/forms';**
- Trois principaux outils de création de formulaires
  - **FormControl** - Contrôle un seul élément de formulaire
  - **FormGroup** - Contrôle un groupe défini de **FormControls**
  - **FormArray** - Contrôle un groupe suivi par index de **FormControls**, **FormGroup**s ou **FormArrays**
- Le **FormControl** instancié sur le contrôleur agit comme la source de vérité pour chaque élément de formulaire respectif. Toute manipulation ou validation de données est attachée au **FormControl**

# Utiliser **FormControl**

- **FormControl** étend la classe **AbstractControl** et a accès à une multitude de méthodes puissantes (**get, set, reset, patch, markAs, validate**)

```
const ctrl = new FormControl({ value: 'n/a', disabled: true }

const myReactiveFormGroup = new FormGroup({
  first: new FormControl('Nancy', Validators.minLength(2)),
  last: new FormControl('Drew'),
}) ;
```

- Connecter au template:

```
<form [formGroup]="myReactiveFormGroup">
  <input placeholder="Name" formControlName="first" />
</form>
```

# Méthodes **FormControl** courantes

```
setValue(value: any, options?: Object):
```

```
patchValue(value: any, options?: Object):
```

```
updateValueAndValidity(opts: {  
    onlySelf?: boolean;  
    emitEvent?: boolean;  
} = {}):
```

```
get valueChanges: Observable
```

```
get statusChanges: Observable (tracks validity state)
```

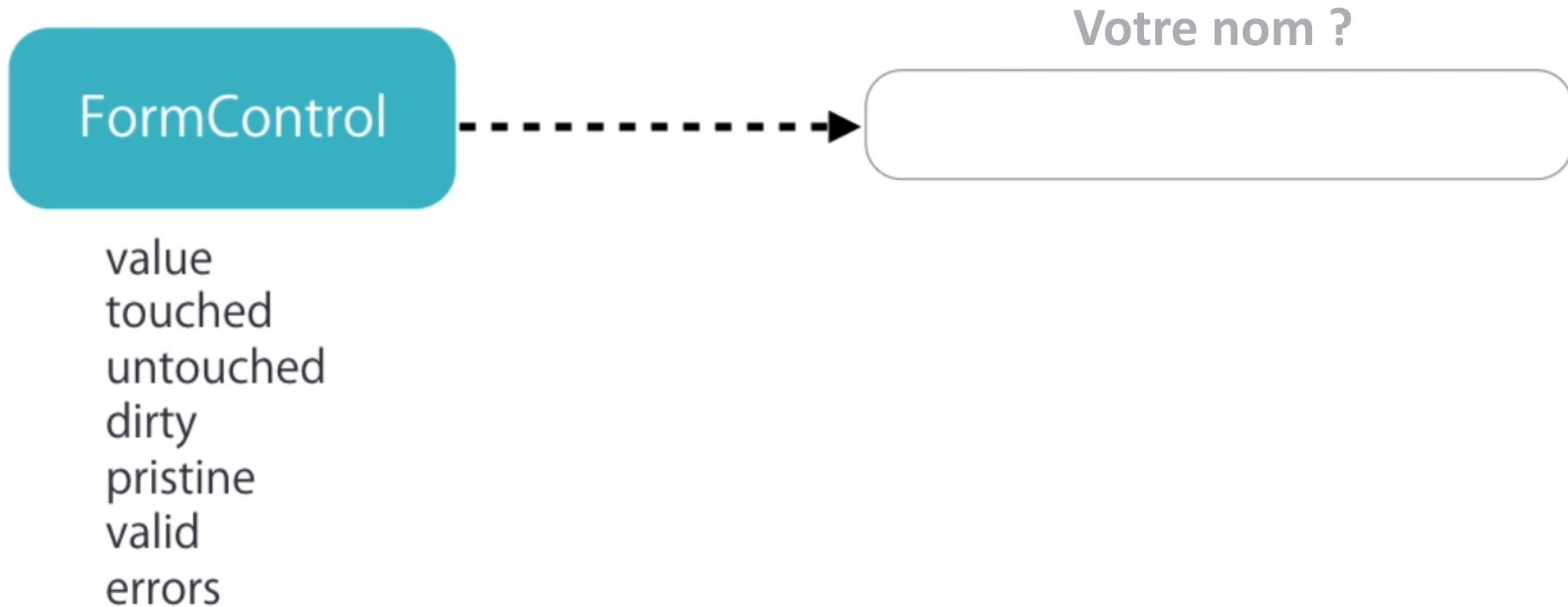
```
setValidators(newValidator: ValidatorFn |
```

```
ValidatorFn[] | null)
```

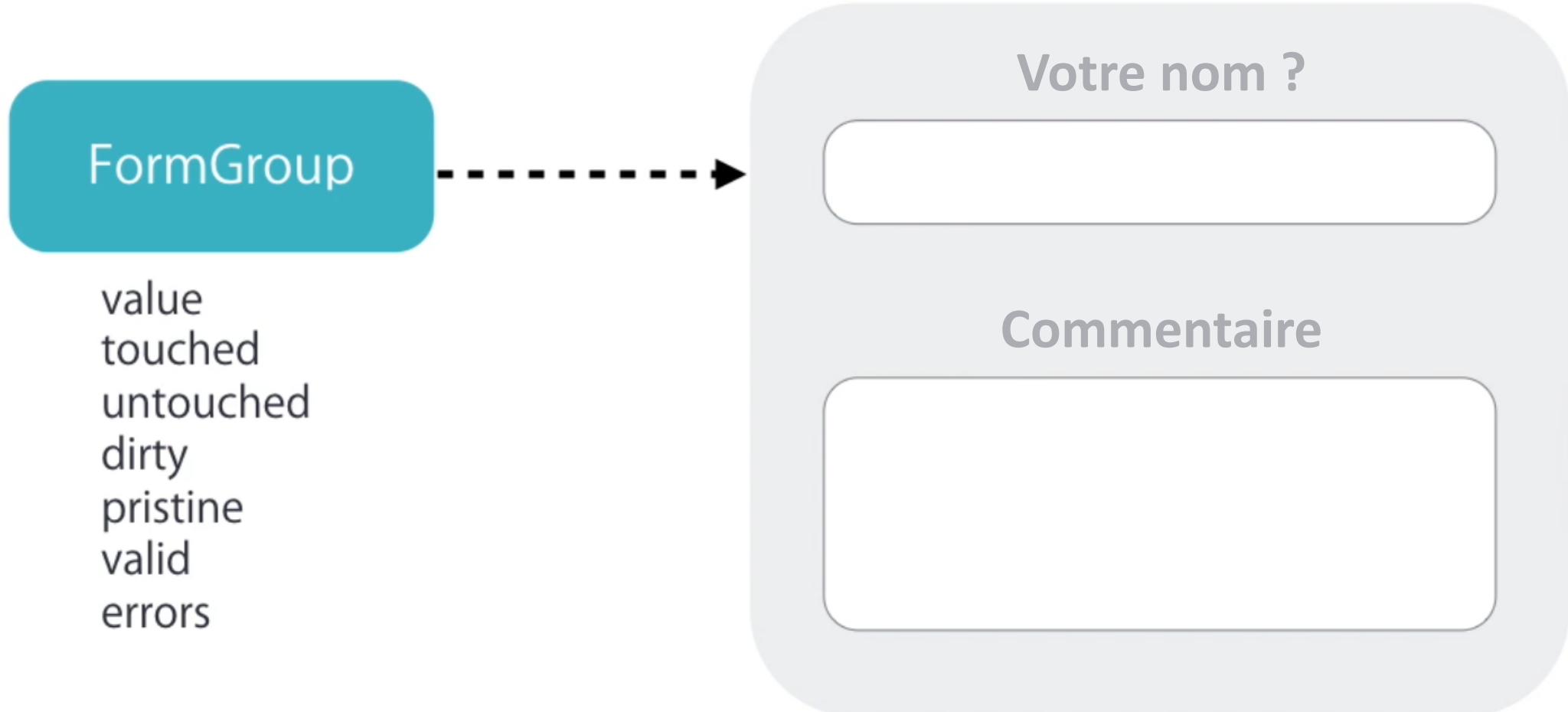
```
disable(opts: {...})
```

```
enable(opts: { ...})
```

# Attributs de **FormControl**



# FormGroup



# FormControl et FormGroup



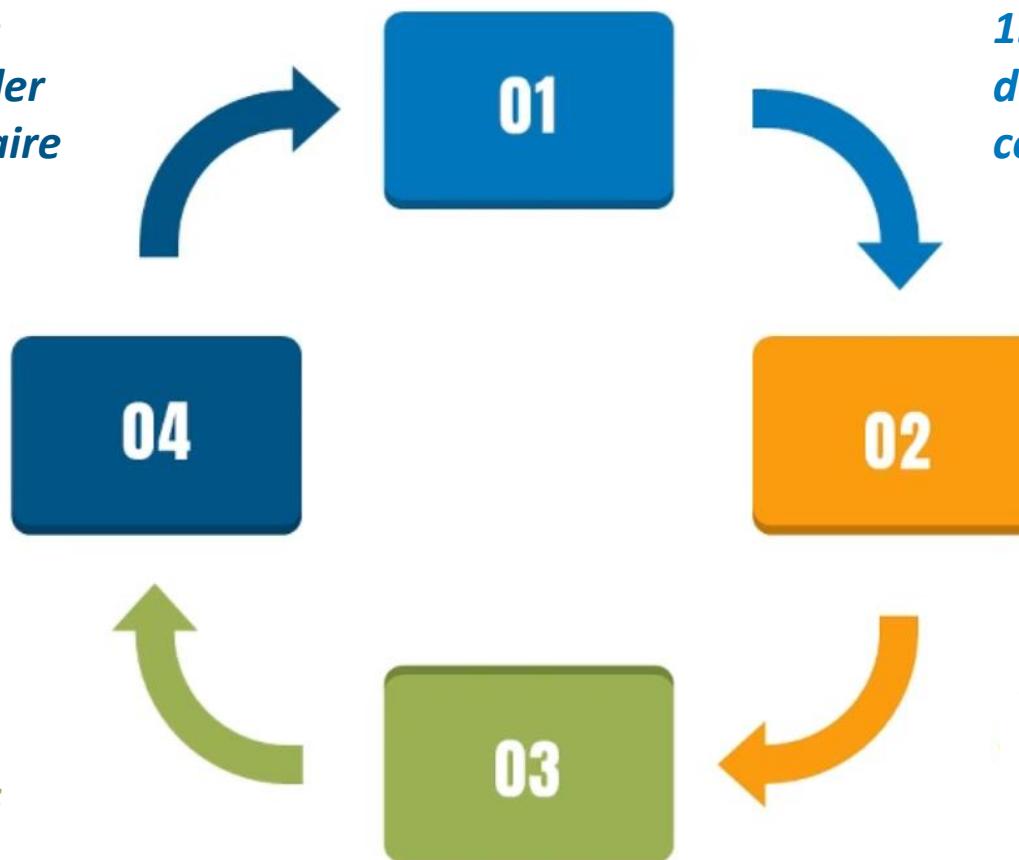
# Créer un formulaire reactif

**4. Utilisez l'instance de FormGroup pour accéder aux valeurs du formulaire**

**3. Créez un modèle de formulaire HTML en ajoutant des directives de recetteur**

**1. Importez ReactiveFormsModule dans le @NgModule où les composants sont déclarés**

**2. Dans le code TypeScript, créez une instance FormGroup pour stocker les valeurs du formulaire**



EXPLORER

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR

- login
- login.component.css
- login.component.html
- login.component.spec.ts
- login.component.ts

- modeles
- pied-page
- pipes
- services
- app-routing.module.ts
- app.component.css
- app.component.html
- app.component.spec.ts
- app.component.ts
- app.module.ts

- assets
- environments
- favicon.ico

> OUTLINE

> TIMELINE

> DEPENDENCIES

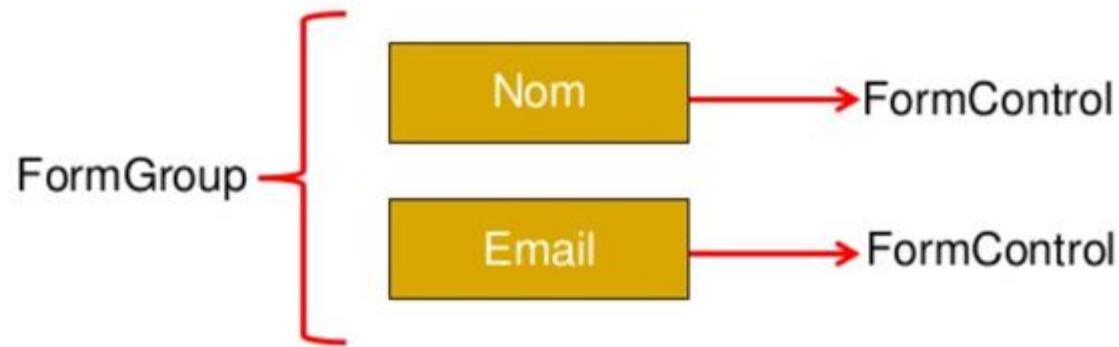
login.component.ts app.module.ts login.component.html

src > app > app.module.ts > ...

```
13 import { HttpClientModule } from '@angular/common/http';
14 import { ErrorComponent } from './error/error.component';
15 import { ReactiveFormsModule } from '@angular/forms';
16
17 @NgModule({
18   declarations: [
19     AppComponent,
20     AccueilComponent,
21     LoginComponent,
22     GetStringAvantTiretPipe,
23     EntetePageComponent,
24     PiedPageComponent,
25     ErrorComponent,
26   ],
27   imports: [
28     BrowserModule,
29     AppRoutingModule,
30     FormsModule,
31     HttpClientModule,
32     ReactiveFormsModule,
33   ],
34   providers: [AuthentificationService],
35   bootstrap: [AppComponent],
36 })
37 export class AppModule {}
```

# FormGroup, FormControl et FormBuilder

Un formulaire réactif est de type **FormGroup** et il regroupe plusieurs **FormControl**

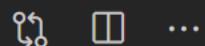


**FormBuilder** est une classe qui vous met à disposition des méthodes facilitant la création d'objet **FormGroup**



login.component.ts X

login.component.html



```
src > app > login > A login.component.ts > LoginComponent > iniatliserFormulaire
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css'],
}

export class LoginComponent implements OnInit {
  utilisateur = new Utilisateur('osall@univ-thies.sn', 'passer123');
  parametresInvalides = true;

  userForm: FormGroup;

  constructor(
    private router: Router,
    private authentificationService: AuthentificationService,
    private formBuilder: FormBuilder
  ) {}

  iniatliserFormulaire() {
    return (this.userForm = this.formBuilder.group({
      login: [this.utilisateur.login],
      motDePasse: [this.utilisateur.motDePasse],
    }));
  }

  ngOnInit(): void {
    this.iniatliserFormulaire();
  }
}
```

File Edit

PremiereAppAngular

localhost:4200/login

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

Vérifier les paramètres saisies, elles ne sont pas les bonnes.

Status: VALID  
osall@univ-thies.sn77777 et passer123  
osall@univ-thies.sn77777:  
.....  
Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

```
1
2
3
4
5
6
7
8
9
10
11
12
13 Status: {{userForm.status}}
14 <br>
15 {{userForm.value.login}} et {{userForm.value.motDePasse}}
16 <div class="col col-md-4">
17   <form [formGroup]="userForm" (ngSubmit)="valider()">
18     <div class="form-group">
19       <input class="form-control" type="email" name="login" formControlName="login">
20     </div>
21     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
22     <button class="btn btn-primary" type="submit">
23       Se Connecter
24     </button>
25   </form>
26 </div>
```

Compile Hero: Off Ln 15, Col 43 (8 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓



login.component.ts X

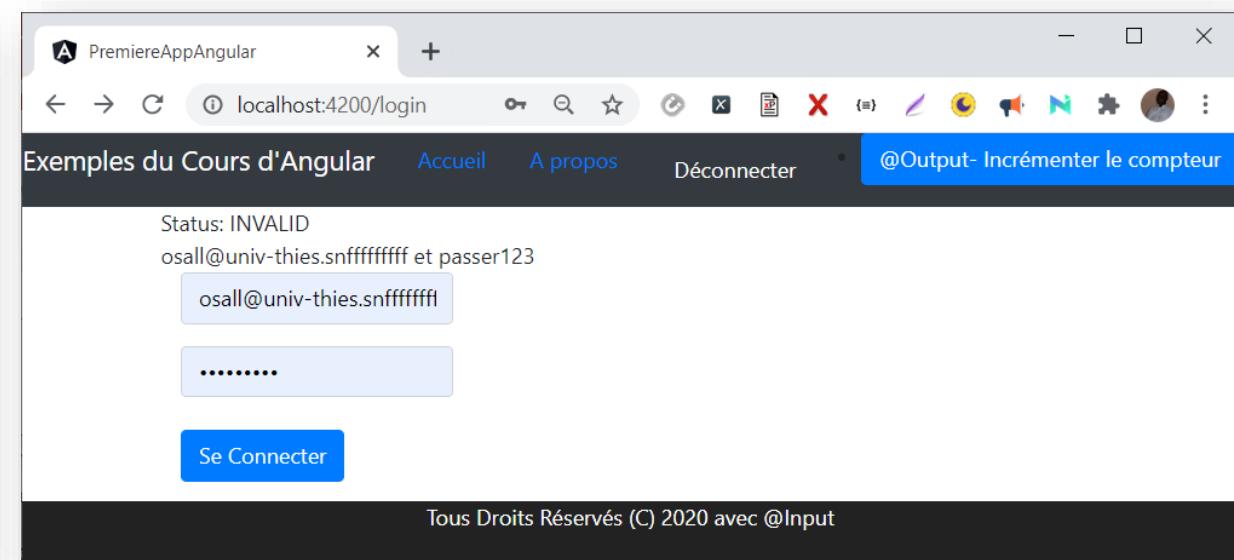
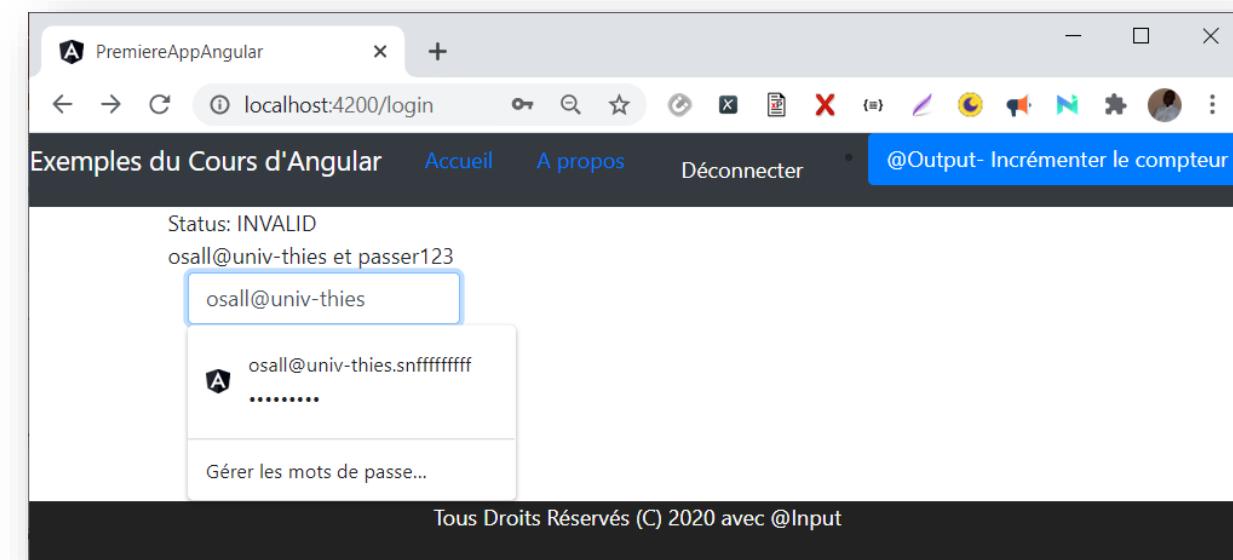
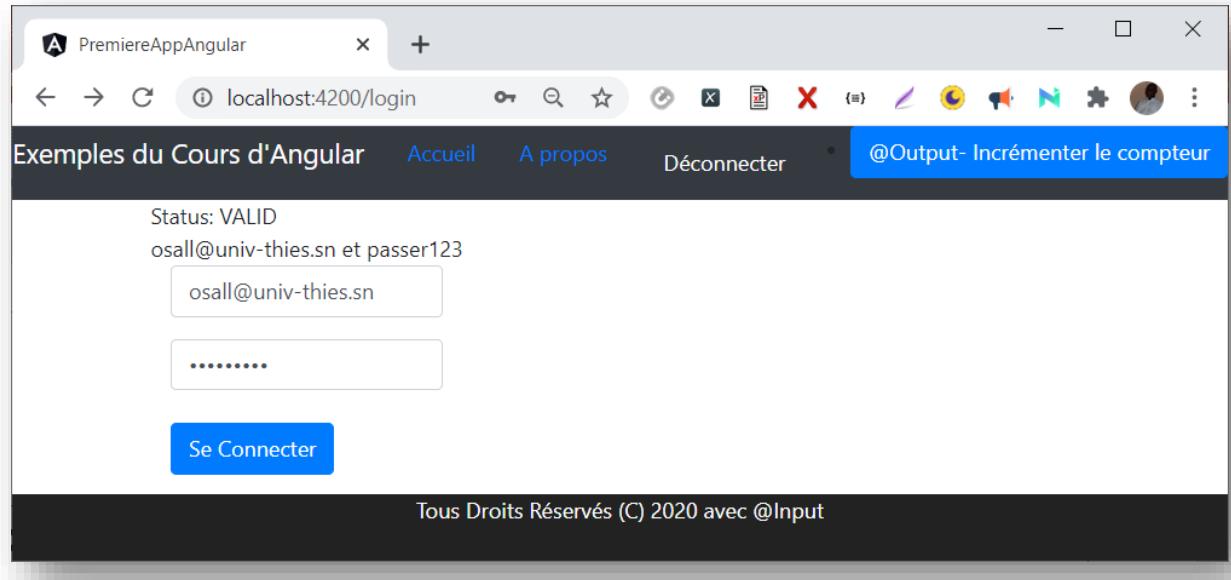
login.component.html



src &gt; app &gt; login &gt; A login.component.ts &gt; LoginComponent &gt; initiliserFormulaire &gt; login

```
19     private router: Router,
20     private authenticationService: AuthenticationService,
21     private formBuilder: FormBuilder
22   ) {}
23
24   initiliserFormulaire() {
25     return (this.userForm = this.formBuilder.group({
26       login: [
27         this.utilisateur.login,
28         [
29           Validators.required,
30           Validators.pattern('^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$'),
31         ],
32       ],
33       motDePasse: [this.utilisateur.motDePasse],
34     });
35   }
36   ngOnInit(): void {
37     this.initiliserFormulaire();
38   }

```





A login.component.ts ●

login.component.html

src &gt; app &gt; login &gt; A login.component.ts &gt; LoginComponent &gt; valider

```
26
27     this.utilisateur.login,
28     [
29         Validators.required,
30         Validators.pattern('^[a-z0-9._%+-]+@[a-z0-9.-]+\.\.[a-z]{2,4}$'),
31     ],
32     [
33         motDePasse: [this.utilisateur.motDePasse],
34     ]);
35 }
36 ngOnInit(): void {
37     this.iniatliserFormulaire();
38 }
39
40 get login(){
41     return this.userForm.get("login");
42 }
43 get motDePasse(){
44     return this.userForm.get("motDePasse");
45 }
```

Afficher les messages d'erreur de validation

Pour rendre notre code court et facile à lire, créons une méthode getter qui renvoie un contrôle de formulaire, dans le fichier **login.component.ts**. De cette façon, nous pouvons facilement accéder au champ de messagerie principal en utilisant **login** au lieu de **userForms.get ("login")** et au champ de messagerie secondaire en utilisant **motDePasse** au lieu de **userForms.get ("motDePasse")**

A PremiereAppAngular

localhost:4200/login

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

Status: VALID  
osall@univ-thies.sn et passer123  
osall@univ-thies.sn  
.....  
Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

The screenshot shows a web browser window displaying a login page for 'Exemples du Cours d'Angular'. The browser title is '.html - premiereAppAngular - Visual Studio Code'. The page URL is 'localhost:4200/login'. The login form has fields for email ('osall@univ-thies.sn') and password ('.....'). Below the form is a blue 'Se Connecter' button. At the top right of the browser, there is a tooltip for the '@Output- Incrémenter le compteur' button, which points to the element path: > form > div.form-group > div > small.text-danger. The status bar at the bottom of the browser indicates 'Status: VALID'.

```
24 Status: {{userForm.status}}<br>
25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
26 <div class="col col-md-4">
27   <form [formGroup]="userForm" (ngSubmit)="valider()">
28     <div class="form-group">
29       <input class="form-control" type="email" name="login" formControlName="login">
30       <div *ngIf="(login.invalid && login.touched) || login.dirty">
31         <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être vide !</small>
32         <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas valide !</small>
33       </div>
34     </div>
35     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36     <button class="btn btn-primary" type="submit">
37       Se Connecter
38     </button>
39   </form>
40 </div>
```

A PremiereAppAngular

localhost:4200/login

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

Status: INVALID  
osall@univ-thies et passer123  
osall@univ-thies  
L'email saisie n'est pas valide !  
.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

form > div.form-group > div > small.text-danger

```
24 Status: {{userForm.status}}<br>
25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
26 <div class="col col-md-4">
27   <form [formGroup]="userForm" (ngSubmit)="valider()">
28     <div class="form-group">
29       <input class="form-control" type="email" name="login" formControlName="login">
30       <div *ngIf="(login.invalid && login.touched) || login.dirty">
31         <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être vide !</small>
32         <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas valide !</small>
33       </div>
34     </div>
35     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36     <button class="btn btn-primary" type="submit">
37       Se Connecter
38     </button>
39   </form>
40 </div>
```

A PremiereAppAngular

localhost:4200/login

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

Status: INVALID et passer123

Le login ne peut pas être vide !

.....

Se Connecter

Tous Droits Réservés (C) 2020 avec @Input

Le code source correspondant à l'écran est :

```
24 Status: {{userForm.status}}<br>
25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
26 <div class="col col-md-4">
27   <form [formGroup]="userForm" (ngSubmit)="valider()">
28     <div class="form-group">
29       <input class="form-control" type="email" name="login" formControlName="login">
30       <div *ngIf="(login.invalid && login.touched) || login.dirty">
31         <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être vide !</small>
32         <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas valide !</small>
33       </div>
34     </div>
35     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36     <button class="btn btn-primary" type="submit">
37       Se Connecter
38     </button>
39   </form>
40 </div>
```

24 Status: {{userForm.status}}<br>
25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
26 <div class="col col-md-4">
27 <form [formGroup]="userForm" (ngSubmit)="valider()">
28 <div class="form-group">
29 <input class="form-control" type="email" name="login" formControlName="login">
30 <div \*ngIf="(login.invalid && login.touched) || login.dirty">
31 <small class="text-danger" \*ngIf="login.errors?.required">Le login ne peut pas être vide !</small>
32 <small class="text-danger" \*ngIf="login.errors?.pattern">L'email saisie n'est pas valide !</small>
33 </div>
34 </div>
35 <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36 <button class="btn btn-primary" type="submit">
37 Se Connecter
38 </button>
39 </form>
40 </div>

master\*+ 0 △ 0

Compile Hero: Off Ln 31, Col 60 (5 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓



A login.component.ts

login.component.html X



Validation du formulaire avec activation ou désactivation du bouton de validation du formulaire selon l'état du formulaire

```
src > login.component.html
  21 Validation du formulaire avec activation ou désactivation du bouton de validation du formulaire selon l'état du formulaire
  22
  23
  24 Status: {{userForm.status}}<br>
  25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
  26 <div class="col col-md-4">
  27   <form [formGroup]="userForm" (ngSubmit)="valider()">
  28     <div class="form-group">
  29       <input class="form-control" type="email" name="login" formControlName="login">
  30       <div *ngIf="(login.invalid && login.touched) || login.dirty">
  31         <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être
  32         <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas val
  33       </div>
  34     </div>
  35     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
  36     <button class="btn btn-primary" type="submit" [disabled]="userForm.invalid">
  37       Se Connecter
  38     </button>
  39   </form>
  40 </div>
```



A login.component.ts

login.component.html X



Soumettre le formulaire si valide: le bouton de validation doit être de type **submit** et utiliser la directive **ngSubmit** dans le tag form

```
src > a[1] Soumettre le formulaire si valide: le bouton de validation doit être de type submit et utiliser la directive ngSubmit dans le tag form
21
22
23
24 Status: {{userForm.status}}<br>
25 {{userForm.value.login}} et {{userForm.value.motDePasse}}
26 <div class="col col-md-4">
27   <form [formGroup]="userForm" (ngSubmit)="valider()">
28     <div class="form-group">
29       <input class="form-control" type="email" name="login" formControlName="login">
30       <div *ngIf="(login.invalid && login.touched) || login.dirty">
31         <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être
32         <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas va
33       </div>
34     </div>
35     <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36     <button class="btn btn-primary" type="submit" [disabled]="userForm.invalid">
37       Se Connecter
38     </button>
39   </form>
40 </div>
```



login.component.ts X

login.component.html



src &gt; app &gt; login &gt; login.component.ts &gt; LoginComponent &gt; valider

```
44     return this.userForm.get("motDePasse");  
45 }  
46  
47 valider() {  
48     this.utilisateur = this.userForm.value;  
49     if (  
50         this.authenticationService.seLoguer(  
51             this.utilisateur.login,  
52             this.utilisateur.motDePasse  
53         )  
54     ) {  
55         this.router.navigate(['accueil', this.utilisateur.login]);  
56         this.parametresInvalides = false;  
57     } else {  
58         this.parametresInvalides = true;  
59     }  
60 }  
61 }  
62 }
```



...



# FormControl dynamique: ajouter dynamiquement des entrées au formulaire

A screenshot of a web browser window titled "PremiereAppAngular" at "localhost:4200/login". The page has a header with "Exemples du Cours d'Angular" and "Se Connecter". A blue navigation bar at the top right contains the text "@Output- Incrémenter le compteur". The main content area shows a status message "Status: VALID" and a user entry "osall@univ-thies.sn et passer123". Below this are two input fields: one with "osall@univ-thies.sn" and another with ".....". At the bottom, there is a section titled "Vous souhaitez" with a green button "Ajouter un souhait" and a blue button "Se Connecter".

A screenshot of a web browser window titled "PremiereAppAngular" at "localhost:4200/login", identical to the first one but showing a validation error. The status message now reads "Status: INVALID". The other elements (user entry, input fields, and buttons) remain the same.

EXPLORER ... A login.component.ts X

> OPEN EDITORS

PREMIEREAPPANGU... .firebase dist e2e node\_modules server src

Définir un tableau de souhaits initialement vide

app accueil entete-page error login login.component.css login.component.html login.component.spec.ts login.component.ts modeles pied-page pipes

> OUTLINE

> TIMELINE

> DEPENDENCIES Phil. Ousmane SALL, Univ. Thiès, SN

src > app > login > A login.component.ts > LoginComponent > initiliserFormulaire > souhait

24      initiliserFormulaire() { > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

25      return (this.userForm = this. )

26            login: [ > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

27              this.utilisateur.login,

28              [ > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

29               Validators.required,

30               Validators.pattern('^[a-z0-9.\_%+-]+@[a-z0-9.-]+\.\.[a-z]{2,4}\$'),

31              ], > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

32              motDePasse: [this.utilisateur.motDePasse], > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

33              souhaits: this.formBuilder.array([]) > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

34          })); > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

35      }

36 }

37

38      getSouhaits(){ > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

39          return this.userForm.get("souhaits") as FormArray; > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

40      }

41

42      onAjouterSouhait(){ > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

43          const newSouhaitControl=this.formBuilder.control("", Validators.required); > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

44          this.getSouhaits().push(newSouhaitControl); > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

45      }

46

47      ngOnInit(): void { > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

48          this.initiliserFormulaire(); > vosSouhaits Aa Ab! \* No results ↑ ↓ ≡ x

49      } Programmation Applications Mobiles

Définir une fonction **onAjouterSouhait** qui ajoute dynamiquement une entrée au formulaire pour saisir un souhait

478

## login.component.html X

src &gt; app &gt; login &gt; login.component.html

```
26  <div class="col col-md-4">
27    <form [formGroup]="userForm" (ngSubmit)="valider()">
28      <div class="form-group">
29        <input class="form-control" type="email" name="login" formControlName="login">
30        <div *ngIf="(login.invalid && login.touched) || login.dirty">
31          <small class="text-danger" *ngIf="login.errors?.required">Le login ne peut pas être vide !</small>
32          <small class="text-danger" *ngIf="login.errors?.pattern">L'email saisie n'est pas valide !</small>
33        </div>
34      </div>
35      <input class="form-control" type="password" name="password" formControlName="motDePasse"><br>
36      <div formArrayName="souhaits">
37        <h2>Vous souhaitez</h2>
38        <div class="form-group" *ngFor="let souhait of getSouhaits().controls; let i=index">
39          <label for="my-input">Souhait n° {{i+1}}</label>
40          <input class="form-control" type="text" name="{{i+1}}" [formControlName]="i">
41        </div>
42        <button class="btn btn-success" type="button" (click)="onAjouterSouhait()">
43          Ajouter un souhait
44        </button>
45      </div>
46      <button class="btn btn-primary" type="submit" [disabled]="userForm.invalid">
47        Se Connecter
48      </button>
49    </form>
50  </div>
```

# Travaux Pratiques n°9

<https://codelab.fun/angularform/intro>



Firebase Realtime  
Database

# Persistance de données et Interaction avec le Cloud

## **FireBase et FireStore**

### Base de données NoSQL Cloud, Temps réel

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- **Persistance de données et Interaction Cloud avec FireBase et FireStore**
- Déploiement Application web Angular sur Firebase
- Authentification et autorisation



# Contenu Cours Angular

- Affichage des données, Gestion des événements, Composants, Directives
- Formulaires basés sur les modèles
- Reactive forms
- Observables et Services HTTP
- Routage et navigation
- Authentification et autorisation
- Déploiement
- Création d'applications sans serveur en temps réel avec Firebase



# Firebase

- Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.
- Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet.



# Firebase vs. Firestore

**Firebase** Realtime database that is structured as a JSON tree but **Cloud Firestore** stores data in documents(which is a set of key-value pair) and collection(which is a collection of documents) formats. Realtime Database stored data in JSON tree but Cloud **firestore** stored data in documents which is very similar to JSON. 3 oct. 2017

[stack overflow.com › questions › whats-the-difference-between-cloud-fir...](https://stackoverflow.com/questions/44371770/whats-the-difference-between-cloud-firebase-realtime-database-and-cloud-firestore)

[What's the difference between Cloud Firestore and the Firebase ...](#)

# Exemple avec Firebase Realtime Database

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

Ajouter un nouveau RV

## Liste des Rendez-Vous

Id	Description	Lieu	Date	
-MKKWI2RyqCHSJgL6ZUz	Desc 1	LIEU 1	20/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWKhlLbyyn_YQfrih	Desc 2	LIEU 2	12/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWOqhjdh-SBX5dHXZ	Desc 3	LIEU 3	04/07/2021	<button>Editer</button> <button>Supprimer</button>

A PremiereAppAngular x +

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos

Bienvenue osall

localhost:4200 indique  
Êtes-vous sûr de vouloir mettre à jour?

Déconnecter

@Output- Incrémenter le compteur

Créer un Rendez-Vous

-MKKWI2RyqCHSJgL6ZUz

Desc 1 modifié

Lieu 1 modifié

20/10/2020

Mettre à jour un nouveau RV

Liste des Rendez-Vous

Id	Description	Lieu	Date	
-MKKWI2RyqCHSJgL6ZUz	Desc 1 modifié	LIEU 1 MODIFIÉ	20/10/2020	<a href="#">Editer</a> <a href="#">Supprimer</a>
-MKKWKhlLbyyn_YQfrih	Desc 2	LIEU 2	12/10/2020	<a href="#">Editer</a> <a href="#">Supprimer</a>
-MKKWOqhjdh-SBX5dHXZ	Desc 3	LIEU 3	04/07/2021	<a href="#">Editer</a> <a href="#">Supprimer</a>

Tous Droits Réservés (C) 2020 avec @Input

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

Desc 1 modifié

Lieu 1 modifié

20/10/2020

Ajouter un nouveau RV

## Liste des Rendez-Vous

Id	Description	Lieu	Date	
-MKKWI2RyqCHSJgL6ZUz	Desc 1 modifié	LIEU 1 MODIFIÉ	20/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWKhlLbyyn_YQfrih	Desc 2	LIEU 2	12/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWOqhjdh-SBX5dHXZ	Desc 3	LIEU 3	04/07/2021	<button>Editer</button> <button>Supprimer</button>

Firebase console

console.firebaseio.google.com/project/demoangular-4c77e/database/demoangular-4c77e/data

Firebase

demoAngular

Accéder à la documentation

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Extensions

: Spark

Gratuit 0 \$/mois

Changer de formule

# Realtime Database

Données Règles Sauvegardes Utilisation

https://demoangular-4c77e.firebaseio.com/

demoangular-4c77e

rvs

- MKKWI2RyqCHSJgL6ZUz
  - date: "20/10/2020"
  - description: "Desc 1 modifié"
  - key: "-MKKWI2RyqCHSJgL6ZUz"
  - lieu: "Lieu 1 modifié"
- MKKWKhlbyyn\_YQfrih
  - date: "12/10/2020"
  - description: "Desc 2"
  - lieu: "Lieu 2"
- MKKWOqhjdh-SBX5dHXZ
  - date: "04/07/2021"
  - description: "Desc 3"
  - lieu: "Lieu 3"

A PremiereAppAngular x +

localhost:4200/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos

Déconnecter @Output- Incrémenter le compteur

# Bienvenue osall

localhost:4200 indique  
Etes-vous sûr que vous voulez supprimer cet élément ?

OK Annuler

## Créer un Rendez-Vous

Desc 1 modifié

Lieu 1 modifié

20/10/2020

Ajouter un nouveau RV

## Liste des Rendez-Vous

Id	Description	Lieu	Date	
-MKKWI2RyqCHSJgL6ZUz	Desc 1 modifié	LIEU 1 MODIFIÉ	20/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWKhlLbyyn_YQfrih	Desc 2	LIEU 2	12/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWOqhjdh-SBX5dHXZ	Desc 3	LIEU 3	04/07/2021	<button>Editer</button> <button>Supprimer</button>

Tous Droits Réservés (C) 2020 avec @Input

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

Desc 1 modifié

Lieu 1 modifié

20/10/2020

Ajouter un nouveau RV

## Liste des Rendez-Vous

Id	Description	Lieu	Date	
-MKKWKhlLbyn_YQfrih	Desc 2	LIEU 2	12/10/2020	<button>Editer</button> <button>Supprimer</button>
-MKKWOqhjdh-SBX5dHXZ	Desc 3	LIEU 3	04/07/2021	<button>Editer</button> <button>Supprimer</button>

Firebase console

console.firebaseio.google.com/project/demoangular-4c77e/database/demoangular-4c77e/data

Firebase

demoAngular

Accéder à la documentation

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Extensions

: Spark

Gratuit 0 \$/mois

Changer de formule

# Realtime Database

Données Règles Sauvegardes Utilisation

https://demoangular-4c77e.firebaseio.com/

demoangular-4c77e

rvs

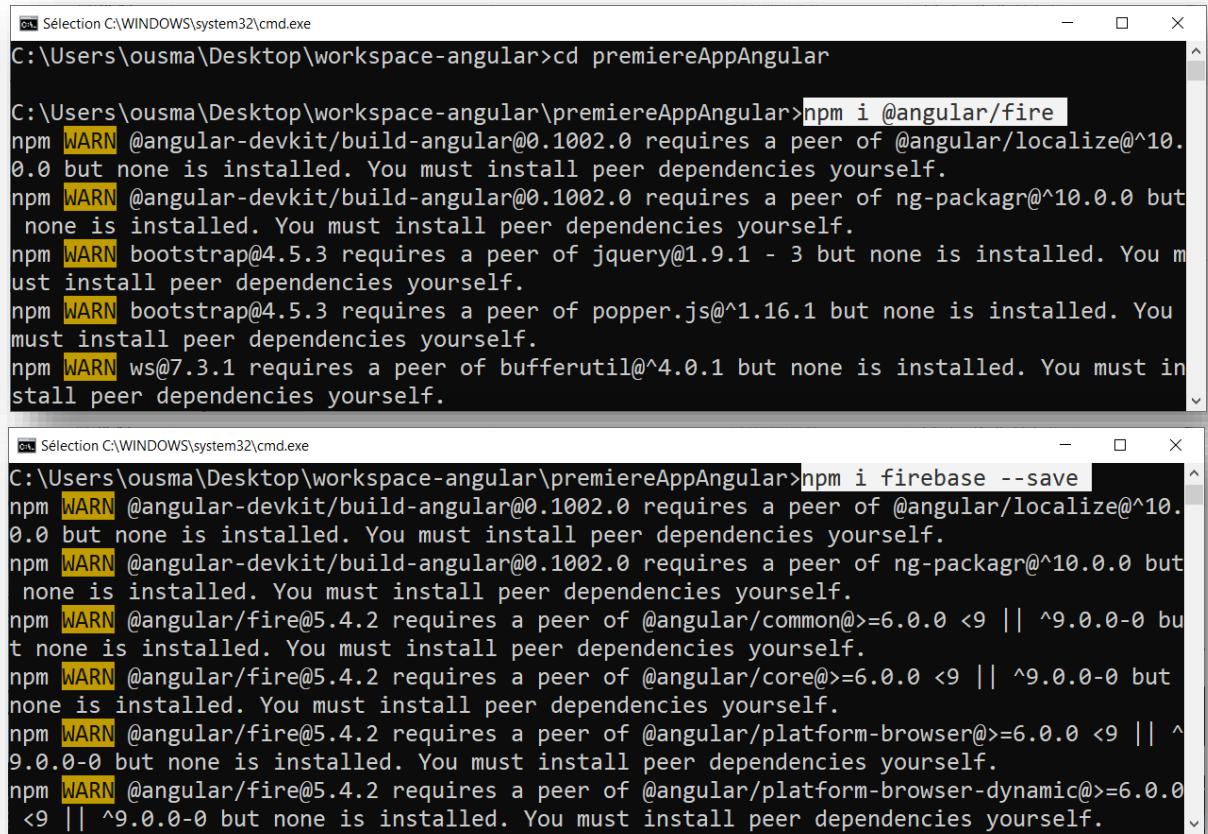
- MKKWKhilbyyn\_YQfrih
  - date: "12/10/2020"
  - description: "Desc 2"
  - lieu: "Lieu 2"
- MKKWOqhjdh-SBX5dHXZ
  - date: "04/07/2021"
  - description: "Desc 3"
  - lieu: "Lieu 3"

# Installer le plugin AngularFire: plugin qui nous permet de faire communiquer Firebase et Angular

- Installer le SDK de Firebase et le package **@angular/fire** en lançant les commandes

Le plugin AngularFire est comme son nom l'indique un plugin Angular, et qui est donc utilisable pour nous avec Angular.

Rappelez-vous, Angular n'est qu'un framework ultime pour créer des applications hybrides, il est toujours basé sur d'autres frameworks, dont le principal est Angular.



The image shows two separate windows of a Windows Command Prompt (cmd.exe) running on a Windows system. Both windows have the title 'Sélection C:\WINDOWS\system32\cmd.exe'.  
The top window shows the command 'npm i @angular/fire' being run. The output includes several 'WARN' messages from npm, indicating that peer dependencies are missing or mismatched. These warnings are for '@angular-devkit/build-angular@0.1002.0' requiring '@angular/localize@^10.0.0' and 'ng-packagr@^10.0.0' requirements, as well as 'bootstrap@4.5.3' and 'popper.js@^1.16.1' requirements.  
The bottom window shows the command 'npm i firebase --save' being run. It also displays similar 'WARN' messages for the same dependencies, specifically for 'angular/fire@5.4.2' and its peer dependencies.

console.firebaseio.google.com/?gclid=Cj0KCQjw28T8BRDbARIsAEOMBcxMqvbjgXoPbjhOVHwZCC6-A4CaNpW7TnIEuqw8cDeZSwRniD... 🔍 ⭐ 📂 🗃 ✎ 🚫 🌐 🎤 🎧 🎯 🎨 🎵 🎪 🎩

Firebase

Accéder à la documentation

Your Firebase projects



Ajouter un projet

firebaseflutter  
fir-flutter-17f20

Exemple FireStore  
exemple-firebase

testAndroid  
testandroid-20ff2

exosionic  
exosionic

NotesMobileMaster  
notesmobilemaster

Voir un projet de démo

iOS

Les projets Firebase sont des conteneurs pour vos applications

Les applications d'un projet partagent des fonctionnalités telles que Realtime Database et Analytics

En savoir plus

iOS </>

Realtime Database Analytics Cloud Functions Authentication Hosting Storage Functions Cloud Storage Cloud Functions Cloud Pub/Sub

Prof. Ousmane SALL, Univ. Thiès, SN

Programmation Applications Mobiles

496

## **×** Créer un projet (Étape 1)

**Un projet Firebase est un conteneur pour vos applications iOS, Android et Web.**

Les applications dans ce projet partageront des fonctionnalités comme Database, la gestion des utilisateurs et Remote Config.

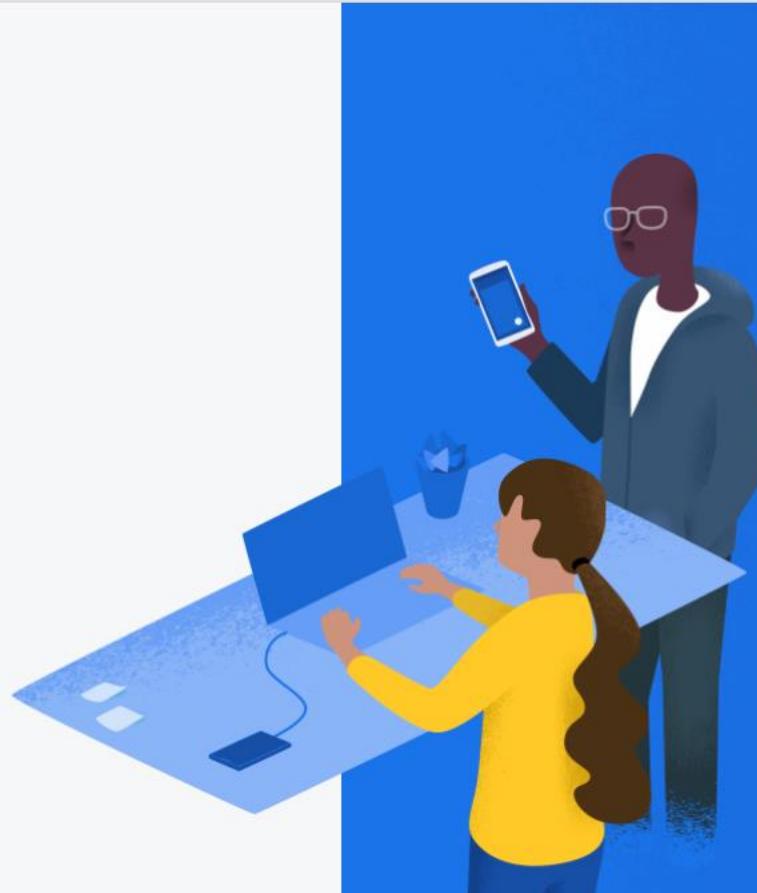
**Un projet Firebase est un projet Cloud.**

Lorsque vous créez un projet Firebase dans la console Firebase, c'est en fait un projet Google Cloud Platform (GCP) que vous créez en coulisses.

Commencez  
nom à votre **projet**<sup>®</sup>

Nom du projet

demoAngular



**Continuer**

console.firebaseio.google.com/project/demoangular-4c77e/overview

Firebase

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Analytics

Dashboard, Events, Conversions, Au...

Enrichir

Predictions, A/B Testing, Cloud Mes...

Extensions

: Spark

Gratuit 0 \$/mois

Changer de formule

demoAngular

demoAngular Formule Spark

# Lancez-vous en ajoutant Firebase à votre application

iOS Android </> | ↴ Ajoutez une application pour démarrer

Stockez et synchronisez les données de votre application en quelques millisecondes

**Authentication**  
Authentifiez et gérez les utilisateurs

**Cloud Firestore**  
Mises à jour en temps réel, requêtes puissantes et scaling automatique

Afficher toutes les fonctionnalités Développer

The screenshot shows the Firebase Project Overview page for the project "demoAngular". The left sidebar includes sections for Vue d'ensemble du projet, Développer (Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Qualité (Crashlytics, Performance, Test Lab, App Distribution), and Extensions (Spark). The main content area displays project details under "Votre projet" and "Paramètres publics". A red box highlights the "Paramètres du projet" tab in the top navigation bar.

Votre projet

Nom du projet	demoAngular
ID du projet	demoangular-4c77e
Numéro du projet	591386647537
Emplacement des ressources GCP par défaut	Non configurée
Clé API Web	AlzaSyCRCwP4WwLkl8krzCjK_grMLobcTrjQ0uM

Paramètres publics

Ces paramètres déterminent quelles instances de votre projet sont publiques

Nom public	project-591386647537
Adresse e-mail associée	Non configurée

Vos applications

console.firebaseio.google.com/project/demoangular-4c77e/settings/general

Firebase

Vue d'ensemble du proj

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Analytics

Dashboard, Events, Conversions, A...

Extensions

: Spark

Gratis 0 \$/mois

Changer de formule

Prof. Ousmane SALL, Univ. Thiès, SN

demoAngular

# Paramètres

Paramètres généraux Cloud Messaging Intégrations Comptes de service Confidentialité des données Utilisateurs et autorisations

Votre projet

Nom du projet	demoAngular
ID du projet ⓘ	demoangular-4c77e
Numéro du projet ⓘ	591386647537
Emplacement des ressources GCP par défaut ⓘ	Non configurée
Clé API Web	AlzaSyCRCwP4WwLkl8krzCjK_grMLobcTrjQ0uM

Paramètres publics

Ces paramètres déterminent quelles instances de votre projet sont publiques

Nom public ⓘ	project-591386647537
Adresse e-mail associée ⓘ	Non configurée

Vos applications

Votre projet ne comporte aucune application

Pour démarrer, sélectionnez une plateforme

iOS Android </> Applications Mobiles

Accéder à la documentation

?

500

## **× Ajouter Firebase à votre application Web**

## 1 Enregistrer l'application

Pseudo de l'application

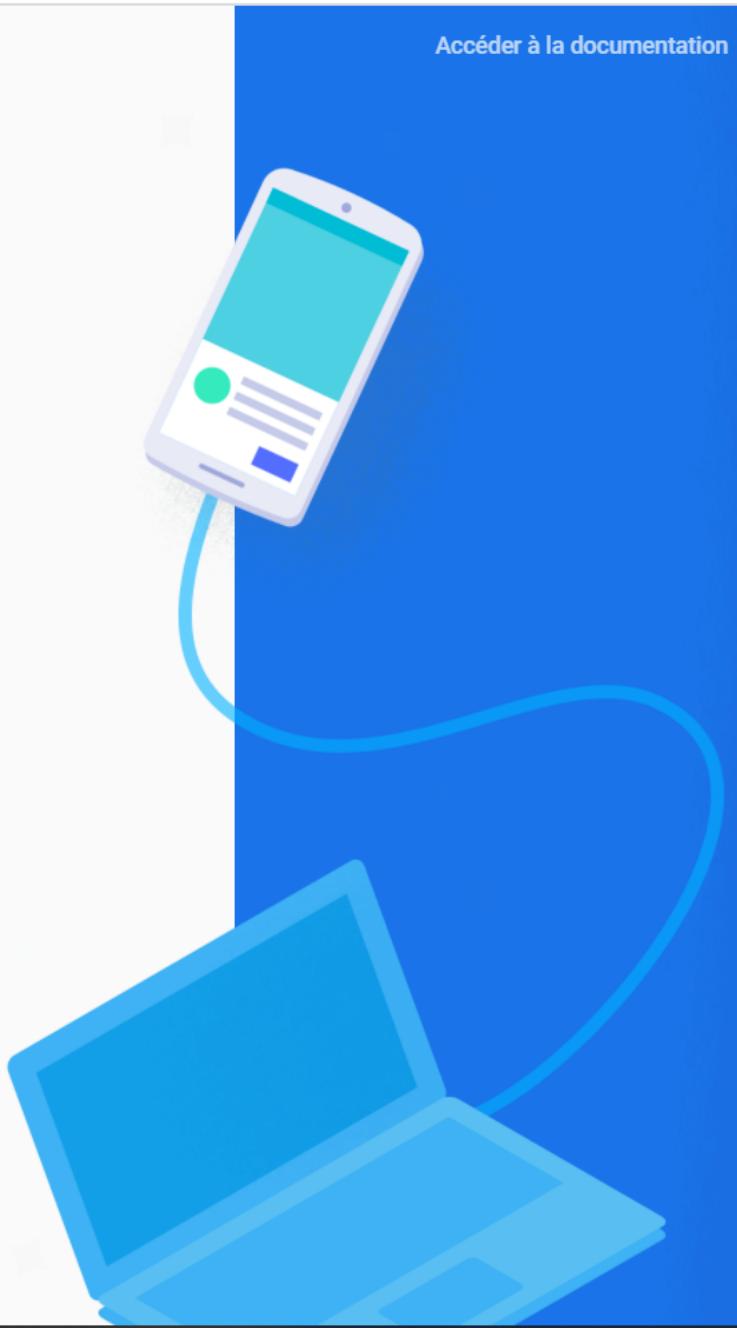
## Exemple Firebase Angular

Configurez également Firebase Hosting pour cette application. [En savoir plus](#)

Hosting peut être configuré plus tard. Vous pouvez commencer à l'utiliser gratuitement à tout moment.

## Enregistrer l'application

## 2 Ajouter le SDK Firebase



## Ajouter Firebase à votre application Web

✓ Enregister l'application

2 Ajouter le SDK Firebase

Copiez et collez ces scripts en bas de votre balise <body>, et ce, avant d'utiliser les services Firebase :

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.24.0.firebaseio-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.24.0.firebaseio-analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  var firebaseConfig = {
    apiKey: "AIzaSyCRCwP4WwLk18krzCjK_grMLobcTrjQ0uM",
    authDomain: "demoangular-4c77e.firebaseioapp.com",
    databaseURL: "https://demoangular-4c77e.firebaseioio.com",
    projectId: "demoangular-4c77e",
    storageBucket: "demoangular-4c77e.appspot.com",
    messagingSenderId: "591386647537",
    appId: "1:591386647537:web:4d878e57d0f5212b088250",
    measurementId: "G-LTHLZ2B2P5"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

Une fois votre application enregistrée, le travail sur le site est terminé, Firebase vous propose maintenant de copier un code pour authentifier votre projet Firebase.

Ces codes de configuration Firebase sont indispensables pour associer toutes applications mobiles avec une base de données, sur Angular ou iOS et Android.





EXPLORER

...

ponent.ts

rest-api.service.ts

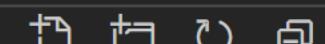
environment.ts X



...

## &gt; OPEN EDITORS

PREMIEREAPPANGU...



&gt; dist

&gt; e2e

&gt; node\_modules

&gt; serveur

&gt; src

&gt; app

&gt; assets

&gt; environments

TS environment.prod.ts A

TS environment.ts M

★ favicon.ico A

index.html A

TS main.ts A

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

src &gt; environments &gt; TS environment.ts &gt; [ ] environment &gt; [ ] firebase &gt; [ ] measurementId

```
1 // This file can be replaced during build by using the `fileReplacem
2 // `ng build --prod` replaces `environment.ts` with `environment.pro
3 // The list of file replacements can be found in `angular.json` .
4
5 export const environment = {
6   production: false,
7   firebase: [
8     apiKey: "AIzaSyCRCwP4WwLk18krzCjK_grMLobcTrjQ0uM",
9     authDomain: "demoangular-4c77e.firebaseio.com",
10    databaseURL: "https://demoangular-4c77e.firebaseio.com",
11    projectId: "demoangular-4c77e",
12    storageBucket: "demoangular-4c77e.appspot.com",
13    messagingSenderId: "591386647537",
14    appId: "1:591386647537:web:4d878e57d0f5212b088250",
15    measurementId: "G-LTHLZ2B2P5"
16  ]
17};
```

Ouvrez maintenant le fichier à l'emplacement  
“~/src/environments/environment.ts” puis ajoutez les informations  
d'identification du projet Firebase comme indiqué ci-dessous

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- app.component.ts M
- app.module.ts M
- > assets
- environments
  - environment.prod.ts M
  - environment.ts M
- favicon.ico A
- index.html A
- main.ts A
- polyfills.ts A
- styles.css M
- test.ts A
- .browserslistrc A
- .editorconfig A
- .firebaserc U
- .gitignore A
- angular.json M
- {...} bd.json U

> OUTLINE

> TIMELINE

> DEPENDENCIES

Phil. Ousmane SALL, Univ. Thiès, SN

environment.prod.ts X

src > environments > environment.prod.ts > environment > firebase

```
1 export const environment = {
2   production: false,
3   firebase: {
4     apiKey: "AIzaSyCRCwP4WwLk18krzCjK_grMLobcTrjQ0uM",
5     authDomain: "demoangular-4c77e.firebaseio.com",
6     databaseURL: "https://demoangular-4c77e.firebaseio.com",
7     projectId: "demoangular-4c77e",
8     storageBucket: "demoangular-4c77e.appspot.com",
9     messagingSenderId: "591386647537",
10    appId: "1:591386647537:web:4d878e57d0f5212b088250",
11    measurementId: "G-LTHLZ2B2P5"
12  };
13}
14
```

Ouvrez maintenant le fichier à l'emplacement  
“~/src/environments/environment.ts” puis ajoutez les  
informations d'identification du projet Firebase comme indiqué  
ci-dessous

# Associer son app Angular et son projet Firebase

- C'est dans ce fichier **app.module.ts** que se passe une grande partie des déclarations et configuration de plugin.
- C'est dans ce fichier que vous devez déclarer la plupart des plugins que vous avez installé dans votre application.
- Sans ça, vous ne pourrez pas utiliser les différentes fonctions mises à disposition par le plugin, et par exemple pour AngularFire, vous ne serez pas connecté à une base de données en particulier donc vos fonctions n'auront aucunes actions.
- Tout en haut de ce fichier, vous retrouvez tous les plugins utiles au fonctionnement de votre application. C'est ici que nous allons ajouter nos plugins.

# Associer son app Angular et son projet Firebase

- Chacun de ces modules est utile à quelques chose de particulier dans notre application et son fonctionnement:
  - **AngularFireModule**: permet d'associer notre application Angular à notre projet Firebase grâce aux code de configuration
  - **AngularFireDatabaseModule**: donne accès aux fonctions de base de données Firebase ( envoi et récupération d'informations sous forme d'objet et de liste)
  - **AngularFireAuthModule**: contient toutes les fonctions d'authentification, par email et mot de passe, Facebook ou Google
  - **AngularFireStorageModule**: permet de stocker et de récupérer des fichiers et du contenu dans la base de stockage Firebase (photos, vidéos, audio, PDF, etc)

# Associer son app Angular et son projet Firebase

- Selon les types d'applications et les fonctionnalités à développer, nous aurons besoins de deux ou plusieurs modules.
  - Le module **AngularFireModule** est indispensable, vous êtes obligé de l'importer pour utiliser Firebase.
  - Ensuite le plugin **AngularFireDatabaseModule** revient lui très souvent car les bases de données sont au cœur de toutes les application dynamique.
  - Le plugin **AngularFireAuthModule** est utile lorsque que vous développez une application à interface sociale pour identifier vos utilisateurs.
  - Le plugin **AngularFireStorageModule** revient probablement le moins souvent, il est aussi le plus technique à utiliser, mais permet par exemple le partage de photos entre amis ou la création d'une photo de profil.

EXPLORER

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGU...

- firebase-service.service.ts U
- firebase-service.service... U
- rest-api.service.spec.ts U
- rest-api.service.ts U
- app-routing.module.ts M
- app.component.css M
- app.component.html M
- app.component.spec.ts A
- app.component.ts M
- app.module.ts M

> assets

environments

- environment.prod.ts M
- environment.ts M
- favicon.ico A
- index.html A
- main.ts A
- polyfills.ts A
- styles.css M
- test.ts A
- .browserslistrc A
- .editorconfig A
- .firebaserc U
- .gitignore A

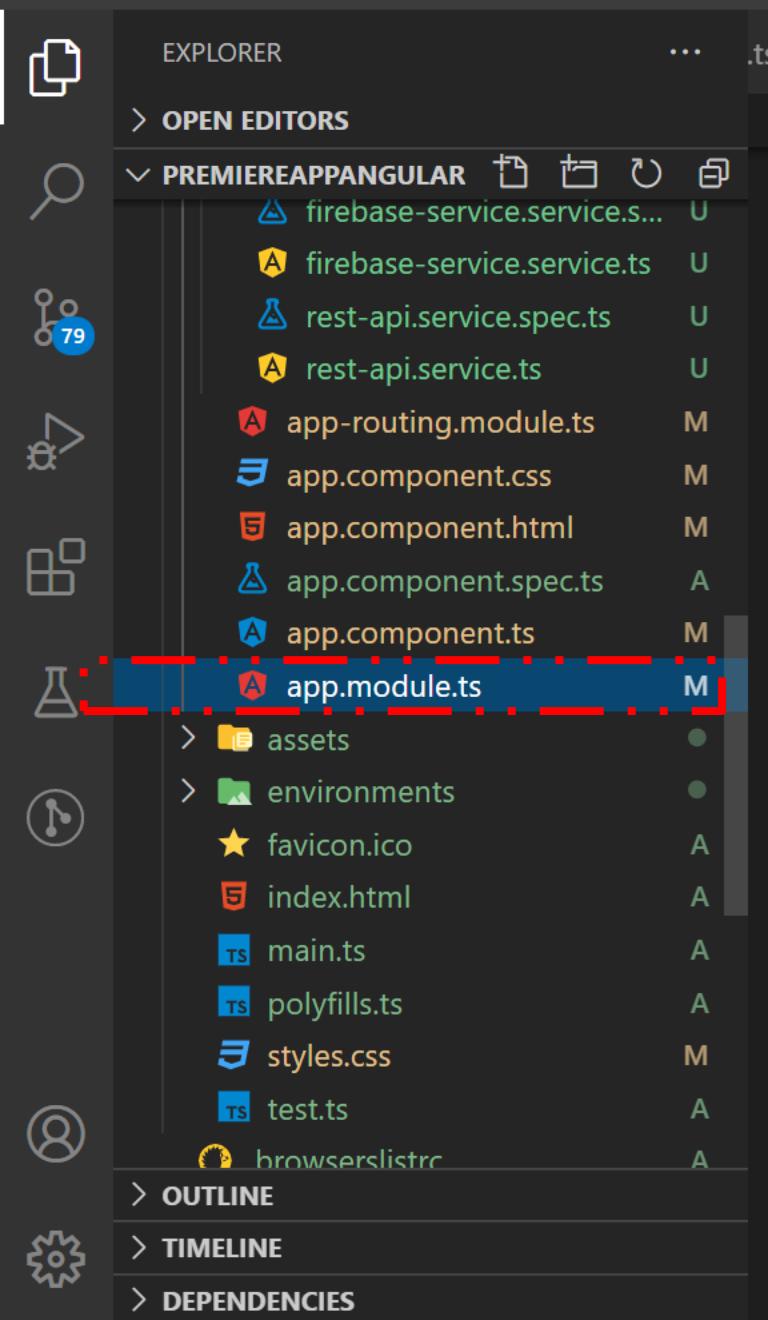
> OUTLINE

> TIMELINE

> DEPENDENCIES

src > app > app.module.ts > AppModule

```
14 import { ErrorComponent } from './error/error.component';
15 import { ReactiveFormsModule } from '@angular/forms';
16 import { AngularFireModule} from "@angular/fire";
17 import { AngularFireDatabaseModule} from "@angular/fire/database/";
18 import { AngularFirestoreModule} from "@angular/fire/firestore";
19 import { AngularFireAuthModule} from "@angular/fire/auth";
20 import { environment } from 'src/environments/environment';
21
22 @NgModule({
23   declarations: [
24     AppComponent,
25     AccueilComponent,
26     LoginComponent,
27     GetStringAvantTiretPipe,
28     EntetePageComponent,
29     PiedPageComponent,
30     ErrorComponent,
31   ],
32   imports: [
33     BrowserModule,
34     AppRoutingModule,
35     FormsModule,
36     HttpClientModule,
37     ReactiveFormsModule,
38     AngularFireModule.initializeApp(environment.firebaseio),
39     AngularFirestoreModule, AngularFireAuthModule, AngularFireDatabaseModule,
40   ],
41   providers: [AuthentificationService],
42   bootstrap: [AppComponent],
43 })
44 export class AppModule {}
```



```
src > app > A app.module.ts > ...
14 import { ErrorComponent } from './error/error.component';
15 import { ReactiveFormsModule } from '@angular/forms';
16 import { AngularFireModule} from "angularfire2";
17 import { AngularFireDatabaseModule} from "angularfire2/database";
18 import { environment } from 'src/environments/environment';
19
20
21 @NgModule({
22   declarations: [
23     AppComponent,
24     AccueilComponent,
25     LoginComponent,
26     GetStringAvantTiretPipe,
27     EntetePageComponent,
28     PiedPageComponent,
29     ErrorComponent,
30   ],
31   imports: [
32     BrowserModule,
33     AppRoutingModule,
34     FormsModule,
35     HttpClientModule,
36     ReactiveFormsModule,
37     AngularFireModule.initializeApp(environment.firebaseio),
38     AngularFireDatabaseModule,
```

## Firebase

demoAngular ▾

Accéder à la documentation

[Vue d'ensemble du projet](#)

### Développer

- [Authentication](#)
- [Cloud Firestore](#)
- [Realtime Database](#)
- [Storage](#)
- [Hosting](#)
- [Functions](#)
- [Machine Learning](#)

### Qualité

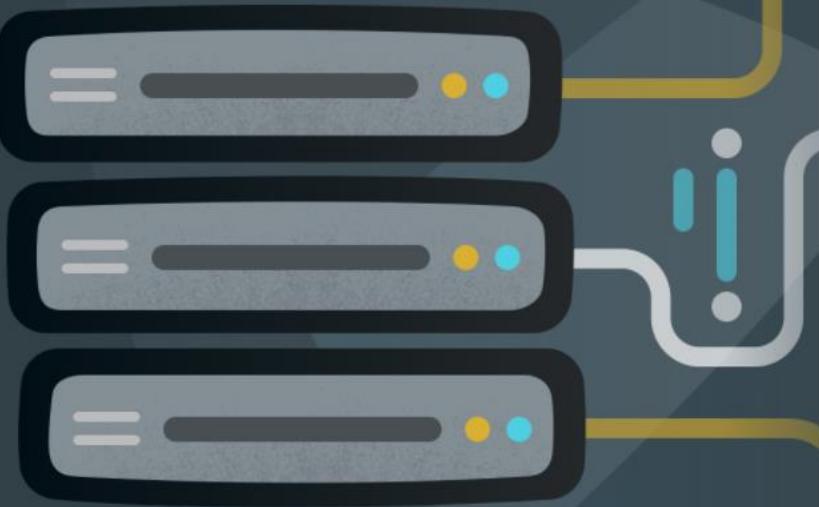
- [Crashlytics](#)
- [Performance](#)
- [Test Lab](#)
- [App Distribution](#)

### Extensions

: Spark  
Gratuit 0 \$/mois [Changer de formule](#)

# Realtime Database

Stockez et synchronisez des données en temps réel.

[Créer une base de données](#)

Realtime Database vous convient-il le mieux ? [Comparer les bases de données](#)

[En savoir plus](#)

Que dois-je faire pour commencer ?  
[Afficher les documents](#)

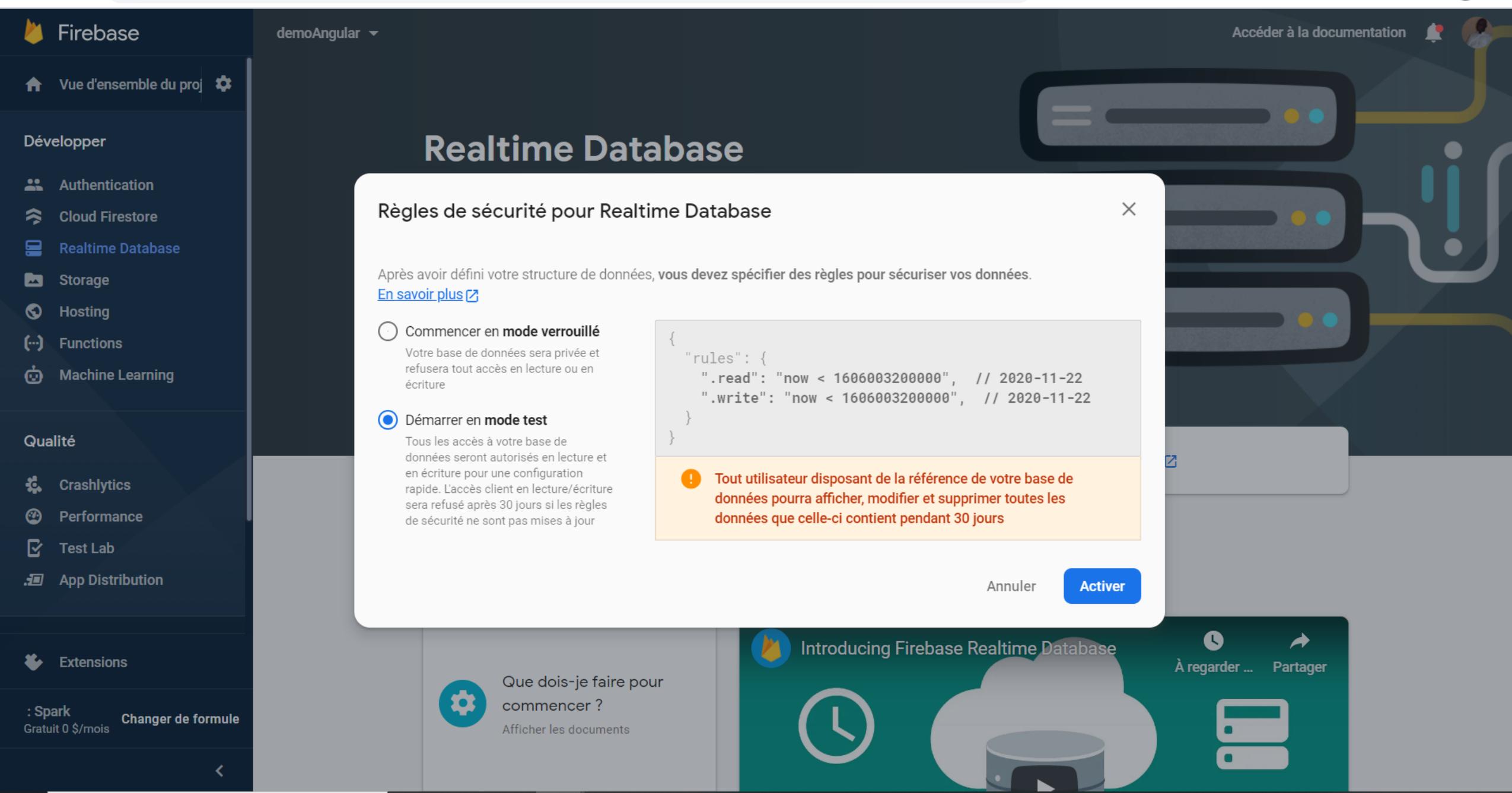


### Introducing Firebase Realtime Database



À regarder ... [Partager](#)





# Firebase

[Vue d'ensemble du projet](#)

## Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

## Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

## Extensions

: Spark  
Gratuit 0 \$/mois [Changer de formule](#)

demoAngular

[Accéder à la documentation](#)

# Realtime Database

[Données](#) [Règles](#) [Sauvegardes](#) [Utilisation](#)<https://demoangular-4c77e.firebaseio.com/>

demoAngular - Console Firebase +

console.firebaseio.google.com/project/demoangular-4c77e/database/demoangular-4c77e/rules

Firebase

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Extensions

: Spark  
Gratuit 0 \$/mois   Changer de formule

demoAngular ▾

Accéder à la documentation

Realtime Database

Données   Règles   Sauvegardes   Utilisation

Modifier les règles   Surveiller les règles

Espace de test dédié aux règles

```
1  {
2    "rules": {
3      ".read": "true",   // 2020-11-22
4      ".write": "true", // 2020-11-22
5    }
6 }
```



EXPLORER

...

ponent.html

A accueil.component.ts

TS rendez-vous.ts X



...

&gt; OPEN EDITORS

&lt; PREMIEREAPPANGULAR

accueil

accueil.component.css

accueil.component.html

accueil.component.spec.ts

accueil.component.ts

entete-page

error

login

modeles

rendez-vous.spec.ts

rendez-vous.ts

utilisateur.spec.ts

utilisateur.ts

pied-page

&gt; OUTLINE

&gt; TIMELINE

&gt; DEPENDENCIES

Prof. Ousmane SALL, Univ. Thiès, SN

src &gt; app &gt; modeles &gt; TS rendez-vous.ts &gt; RendezVous &gt; key

```
1  export class RendezVous {  
2      key: string;  
3      description: string;  
4      lieu: string;  
5      date: String;  
6  }  
7
```



EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- accueil
  - accueil.component.css
  - accueil.component.html
  - accueil.component.spec.ts
  - accueil.component.ts
- entete-page
- error
- login
- modeles
  - rendez-vous.spec.ts
  - rendez-vous.ts
  - utilisateur.spec.ts
  - utilisateur.ts
- pied-page
- pipes
- services
  - authentification.service.spec.ts
  - authentification.service.ts
  - firebase-service.service.spec.ts

> OUTLINE

> TIMELINE

> DEPENDENCIES

Phol. Ousmane SALL, Univ. Thiès, SN

ponent.html accueil.component.ts rendez-vous.ts firebase-service.service.ts

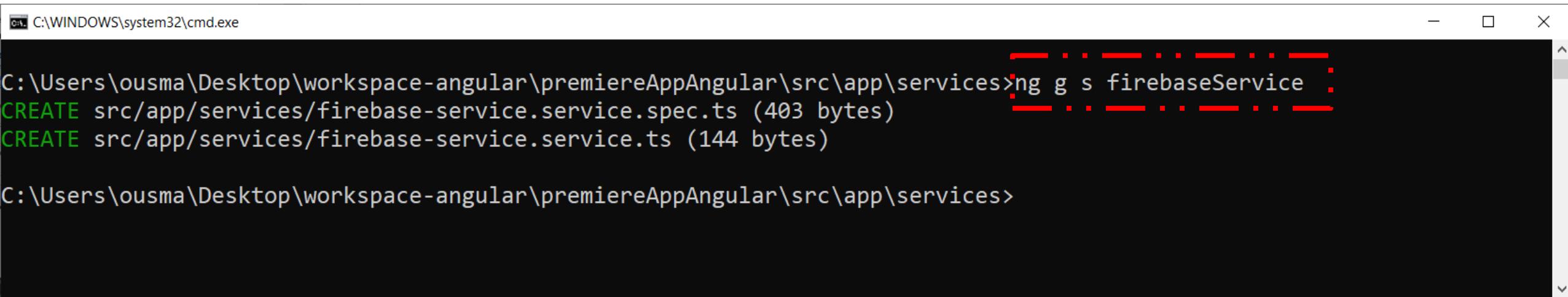
src > app > accueil > accueil.component.ts > AccueilComponent > rvSaisie

```
13 export class AccueilComponent implements OnInit {  
14     nomUtilisateur = '';  
15     rvSaisie = new RendezVous();  
16  
17     addRv() {  
18         this.firebaseioServiceService.create(this.rvSaisie);  
19         this.chargerLesRvs();  
20     }  
21  
22     rendezVous: any = [];  
23  
24     update = false;  
25     // cleRvMisAJour = null;  
26  
27     // Update Rendez-Vous  
28     updateRv(rv: RendezVous) {  
29         console.log('Clé1: ' + rv.key);  
30         this.rvSaisie = rv;  
31         this.update = true;  
32     }  
33     // Update Rendez-Vous  
34     mettreAJour() {  
35         console.log('Clé2: ' + this.rvSaisie.key);  
36         if (window.confirm('Êtes-vous sûr de vouloir mettre à jour?')) {  
37             this.firebaseioServiceService.update(this.rvSaisie.key, this.rvSaisie);  
38         }  
39     }  
40 }
```

Programmation Applications Mises à Jour

Compile Hero: Off Ln 16, Col 11 (8 selected) Spaces: 2 UTF-8 LF TS 4.0.3 Prettier: ✓

# Créer un service



A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe". The command being run is "ng g s firebaseService". The output shows two files being created: "src/app/services/firebase-service.service.spec.ts" (403 bytes) and "src/app/services/firebase-service.service.ts" (144 bytes). The command prompt then returns to the directory "C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services>".

```
C:\WINDOWS\system32\cmd.exe
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services>ng g s firebaseService
CREATE src/app/services/firebase-service.service.spec.ts (403 bytes)
CREATE src/app/services/firebase-service.service.ts (144 bytes)

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services>
```

File Edit Selection View Go Run Terminal Help • firebase-service.service.ts - premiereAppAngular - Visual Studio Code

EXPLORER ... accueil.component.html accueil.component.ts rendez-vous.ts firebase-service.service.ts

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR

- rendez-vous.ts
- utilisateur.spec.ts
- utilisateur.ts

> pied-page

> pipes

> services

- authentification.service.spec.ts
- authentification.service.ts
- firebase-service.service.spec.ts
- firebase-service.service.ts**
- rest-api.service.spec.ts
- rest-api.service.ts

app-routing.module.ts

app.component.css

app.component.html

app.component.spec.ts

app.component.ts

app.module.ts

> assets

> environments

- environment.prod.ts
- environment.ts

favicon.ico

index.html

> OUTLINE

> TIMELINE

> DEPENDENCIES

1

src > app > services > firebase-service.service.ts > FirebaseServiceService

```
6     providedIn: 'root',
7 }
8 export class FirebaseServiceService {
9     private dbPath = '/rvs';
10
11    rvs: AngularFireList<RendezVous> = null;
12
13    constructor(private db: AngularFireDatabase) {
14        this.rvs = db.list(this.dbPath);
15    }
16
17    getAll(): AngularFireList<RendezVous> {
18        return this.rvs;
19    }
20
21    create(rv: RendezVous): any {
22        return this.rvs.push(rv);
23    }
24
25    update(key: string, value: any): Promise<void> {
26        return this.rvs.update(key, value);
27    }
28
29    delete(key: string): Promise<void> {
30        return this.rvs.remove(key);
31    }
32
33    deleteAll(): Promise<void> {
34        return this.rvs.remove();
35    }
36 }
```

Ajouter le code au service des méthodes suivantes:

**create**: crée un nouvel enregistrement dans la collection spécifiée à l'aide de la méthode add

**getAll** : Appelez la méthode snapshotChanges qui obtiendra des enregistrements et également vous abonnera pour obtenir des mises à jour

**update**: Mettre à jour l'enregistrement en prenant l'ID puis en appelant la méthode de mise à jour

**delete**: appeler la méthode de suppression en prenant l'ID d'enregistrement

Remplacez le code ci-dessous dans le fichier crud.service.ts

Programmation Applications Mobiles

Compile Hero: Off Ln 36, Col 2 Spaces: 2 UTF-8 LF TS 4.0.3 Prettier: ✓

517

EXPLORER

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR

- yn
- zip-stream
- zone.js
- {...} \_ngcc\_entry\_points\_.json
- serveur
- src
  - app
    - accueil
      - accueil.component.css
      - accueil.component.html
      - accueil.component.spec.ts
      - accueil.component.ts
    - entete-page
    - error
    - login
    - modeles
      - rendez-vous.spec.ts
      - rendez-vous.ts
      - utilisateur.spec.ts
      - utilisateur.ts
    - pied-page
    - pipes
  - services
    - ...

> OUTLINE

> TIMELINE

> DEPENDENCIES

Yassine SALL, Univ. Thiès, SN

accueil.component.html accueil.component.ts X rendez-vous.ts firebase-service.service.ts

src > app > accueil > accueil.component.ts > AccueilComponent > constructor

```
44
45     constructor(
46         private authenticationService: AuthenticationService,
47         private routeActive: ActivatedRoute,
48         private router: Router,
49         // private rest ApiService: Rest ApiService,
50         private firebaseServiceService: FirebaseServiceService
51     ) {}
52
53     chargerLesRvs() {
54         this.firebaseioServiceService
55             .getAll()
56             .snapshotChanges()
57             .pipe(
58                 map((changes) =>
59                     changes.map((c) => ({ key: c.payload.key, ...c.payload.val() }))
60                 )
61             )
62             .subscribe((data) => {
63                 this.rendezVous = data;
64             });
65     }
66
67     ngOnInit(): void {
68         this.chargerLesRvs();
69         if (this.authenticationService.utilisateurConnecte())
70             this.nomUtilisateur = this.routeActive.snapshot.paramMap.get(
71                 'nomUtilisateur'
72             );
73         else this.router.navigateByUrl('/login');
74     }
75
76 // Delete Report issue
```

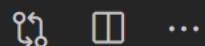
# Créer un nouveau rendez-vous

```
addRv() {
    this.firebaseioServiceService.create(this.rvSaisie);
    this.chargerLesRvs();
}
```



login.component.html

accueil.component.html



```
src > app > accueil > accueil.component.html > div.container > div.container.custom-container > div.col-md-12 > div.form-group > button.btn.btn-success
  6   <div class="container custom-container">
  7     <div class="col-md-12">
  8       <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
  9       <div class="form-group" *ngIf="update">
 10         <input type="text" [disabled]="update" [(ngModel)]="rvSaisie.key" class="form-control" placeholder="Id">
 11       </div>
 12       <div class="form-group">
 13         <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
 14       </div>
 15       <div class="form-group">
 16         <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
 17       </div>
 18       <div class="form-group">
 19         <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
 20       </div>
 21       <div class="form-group" *ngIf="!update">
 22         <button class="btn btn-success btn-lg btn-block" (click)="addRv()">Ajouter un nouveau RV </button>
 23       </div>
 24       <div class="form-group" *ngIf="update">
 25         <button class="btn btn-success btn-lg btn-block" (click)="mettreAJour()">
 26           Mettre à jour un nouveau RV
 27         </button>
 28       </div>
 29     </div>
 30   </div>
 31   <div class="table-responsive-sm">
```



# Mettre à jour un rendez-vous

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** accueil.component.ts - premiereAppAngular - Visual Studio Code.
- Explorer:** Shows the project structure under 'PREMIEREAPPANGU...'. The 'accueil' folder contains 'accueil.component.css', 'accueil.component.html', 'accueil.component.spec.ts', and 'accueil.component.ts'. The 'modeles' folder contains 'rendez-vous.spec.ts' and 'rendez-vous.ts'. There are 80 changes in total.
- Editor:** The 'accueil.component.ts' file is open, showing code related to updating a rendez-vous. The code includes methods like 'updateRv' and 'mettreAJour', and a constructor that injects services like 'authentificationService', 'routeActive', 'router', and 'rest ApiService'.
- Bottom Status Bar:** Prof. O, master\*, 0 △ 0, Compile Hero: Off, Ln 25, Col 9 (6 selected), Spaces: 2, UTF-8, LF, TS 4.0.3, Prettier: ✓, 521.

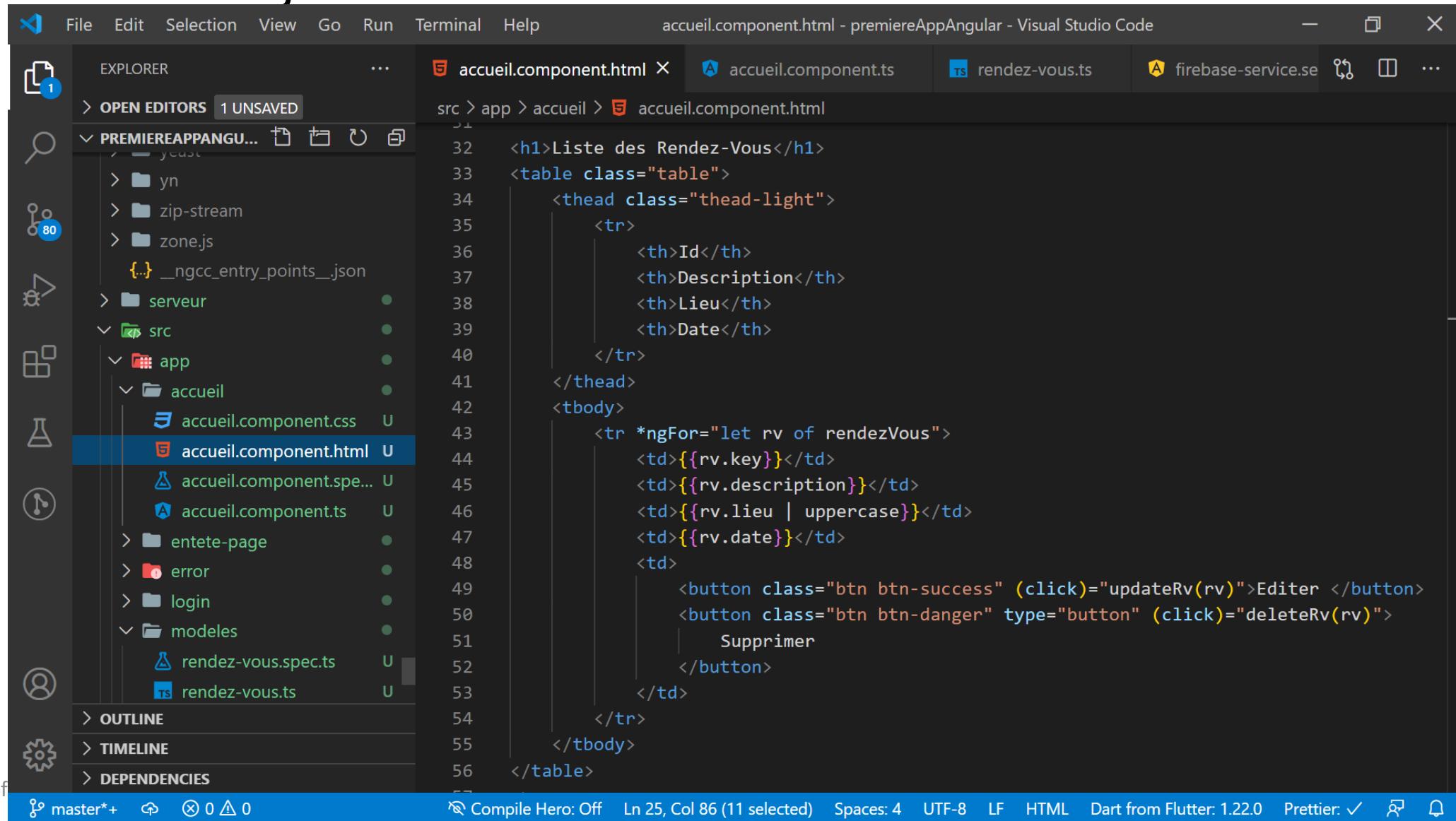
```
update = false;
// cleRvMisAJour = null;

// Update Rendez-Vous
updateRv(rv: RendezVous) {
    console.log('Clé1: ' + rv.key);
    this.rvSaisie = rv;
    this.update = true;
}

// Update Rendez-Vous
mettreAJour() {
    console.log('Clé2: ' + this.rvSaisie.key);
    if (window.confirm('Êtes-vous sûr de vouloir mettre à jour?')) {
        this.firebaseioServiceService.update(this.rvSaisie.key, this.rvSaisie);
        this.chargerLesRvs();
    }
    this.update = false;
    this.chargerLesRvs();
}

constructor(
    private authentificationService: AuthentificationService,
    private routeActive: ActivatedRoute,
    private router: Router,
    // private rest ApiService: Rest ApiService,
```

# Mettre à jour



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** accueil.component.html - premiereAppAngular - Visual Studio Code.
- Explorer:** Shows the project structure under "PREMIEREAPPANGU...".
  - src folder contains: yeast, yn, zip-stream, zone.js, \_\_ngcc\_entry\_points\_\_.json.
  - serveur folder.
  - src folder contains: app, accueil.
    - accueil.component.css
    - accueil.component.html (highlighted)
    - accueil.component.spec.ts
    - accueil.component.ts
  - entete-page folder.
  - error folder.
  - login folder.
  - modeles folder contains: rendez-vous.spec.ts, rendez-vous.ts.
- Editor:** Displays the content of accueil.component.html:

```
<h1>Liste des Rendez-Vous</h1>


| Id         | Description        | Lieu                    | Date        |                                                                                                                                                                                            |
|------------|--------------------|-------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{rv.key}} | {{rv.description}} | {{rv.lieu   uppercase}} | {{rv.date}} | <button class="btn btn-success" (click)="updateRv(rv)">Editer </button>         <button class="btn btn-danger" type="button" (click)="deleteRv(rv)">           Supprimer         </button> |


```
- Bottom Status Bar:** master\*+, 0 0 △ 0, Compile Hero: Off, Ln 25, Col 86 (11 selected), Spaces: 4, UTF-8, LF, HTML, Dart from Flutter: 1.22.0, Prettier: ✓, 522.

# Mettre à jour

```
v class="container">
  <div class="alert alert-success" role="alert">
    <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
  </div>

  <div class="container custom-container">
    <div class="col-md-12">
      <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
      <div class="form-group" *ngIf="update">
        <input type="text" [disabled]="update" [(ngModel)]="rvSaisie.key" class="form-control" placeholder="Id">
      </div>
      <div class="form-group">
        <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
      </div>
      <div class="form-group">
        <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
      </div>
      <div class="form-group">
        <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
      </div>
      <div class="form-group" *ngIf="!update">
        <button class="btn btn-success btn-lg btn-block" (click)="addRv()">Ajouter un nouveau RV </button>
      </div>
      <div class="form-group" *ngIf="update">
        <button class="btn btn-success btn-lg btn-block" (click)="mettreAJour()">Mettre à jour un nouveau RV
          </button>
      </div>
    </div>
  </div>
<h1>Liste des Rendez-Vous</h1>
```

File Edit Selection View Go Run Terminal Help accueil.component.html - premiereAppAngular - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED PREMIEREAPPANGU... yn zip-stream zone.js \_\_ngcc\_entry\_points\_\_.json serveur src app accueil accueil.component.css accueil.component.html accueil.component.spec.ts accueil.component.ts entete-page error login modeles rendez-vous.spec.ts rendez-vous.ts utilisateur.spec.ts utilisateur.ts pied-page pipes services

OUTLINE TIMELINE DEPENDENCIES

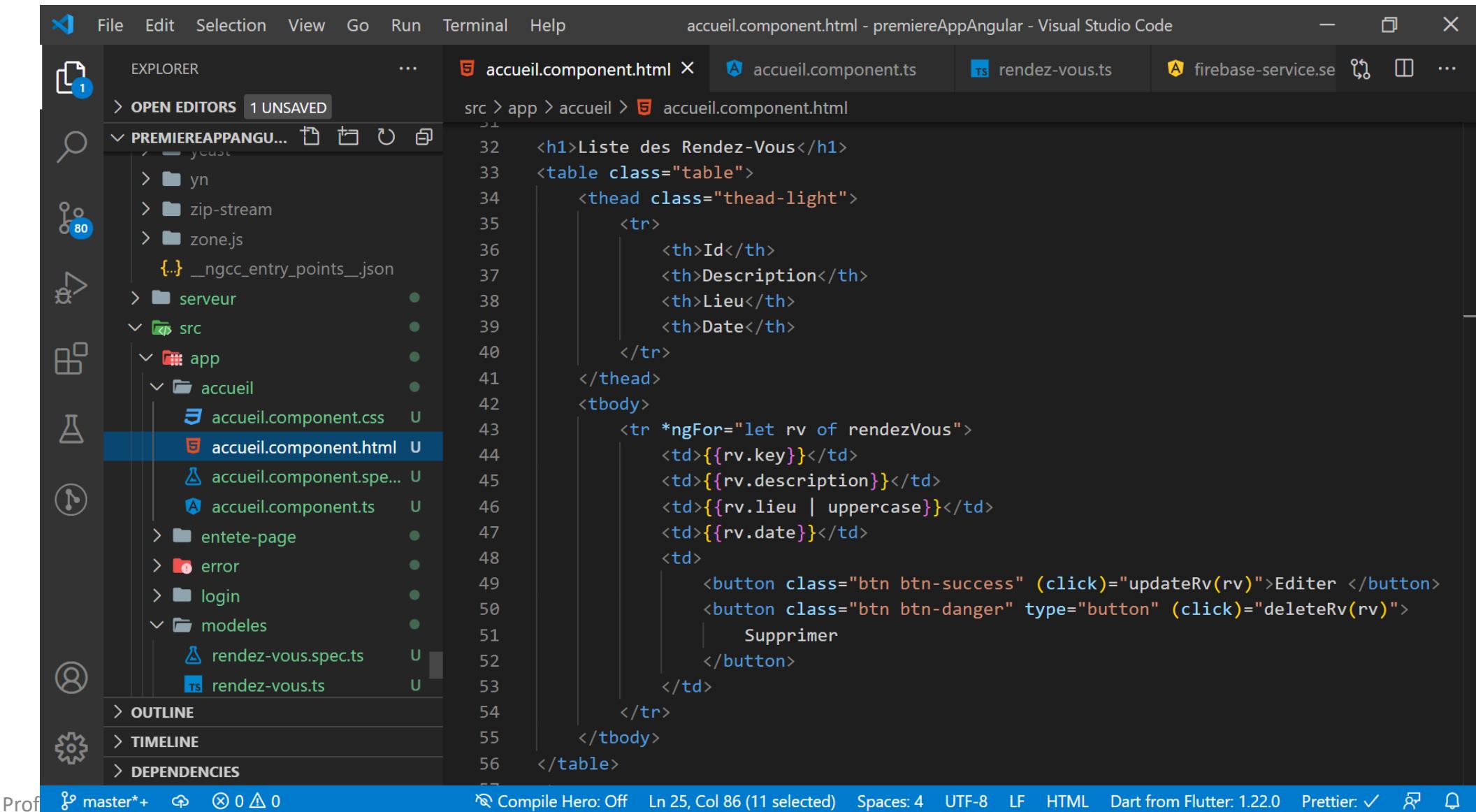
master+ Prof. Ousmane Diakhaté, Côte d'Ivoire, DA

Compile Hero: Off Ln 25, Col 86 (11 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓ 523

# Supprimer un rendez-vous

```
// Delete Rendez-vous
deleteRv(rv:RendezVous) {
    if (
        window.confirm('Etes-vous sûr que vous voulez supprimer cet élément ?')
    ) {
        this.firebaseioServiceService.delete(rv.key);
        this.chargerLesRvs();
    }
}
```

# Suppression, Editer



The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** accueil.component.html - premiereAppAngular - Visual Studio Code.
- Explorer Sidebar:** Shows the project structure under "PREMIEREAPPANGU...". The "src/app/accueil" folder is expanded, showing "accueil.component.css", "accueil.component.html", "accueil.component.spec.ts", and "accueil.component.ts". Other files like "yeast", "yn", "zip-stream", "zone.js", and "rendez-vous.ts" are also listed.
- Editor Area:** Displays the content of "accueil.component.html". The code uses Angular's \*ngFor directive to iterate over a list of rendez-vous (rv) and render a table row for each. Each row contains four columns: Id, Description, Lieu, and Date. The last column (Date) contains two buttons: "Editer" (success button) and "Supprimer" (danger button).
- Bottom Status Bar:** Shows "master\*+", "Compile Hero: Off", "Ln 25, Col 86 (11 selected)", "Spaces: 4", "UTF-8", "LF", "HTML", "Dart from Flutter: 1.22.0", "Prettier: ✓", and "525".

```
src > app > accueil > accueil.component.html
32   <h1>Liste des Rendez-Vous</h1>
33   <table class="table">
34     <thead class="thead-light">
35       <tr>
36         <th>Id</th>
37         <th>Description</th>
38         <th>Lieu</th>
39         <th>Date</th>
40     </tr>
41   </thead>
42   <tbody>
43     <tr *ngFor="let rv of rendezVous">
44       <td>{{rv.key}}</td>
45       <td>{{rv.description}}</td>
46       <td>{{rv.lieu | uppercase}}</td>
47       <td>{{rv.date}}</td>
48       <td>
49         <button class="btn btn-success" (click)="updateRv(rv)">Editer </button>
50         <button class="btn btn-danger" type="button" (click)="deleteRv(rv)">
51           Supprimer
52         </button>
53       </td>
54     </tr>
55   </tbody>
56 </table>
```

NotesMobileMaster – Firebase cc × localhost:8100/home × | +

console.firebaseio.google.com/u/0/project/notesmobilemaster/database.firebaseio/data~2FEtudiants~2FRvCPwylo3opFVBfrVdC6

Plateforme de colle... gradle - Android St... (75) Donovan - Sea... Android Full Course... 7ed85bf35f9a29e79... Configuring a Data... Discover - Kibana generatedata.com osall751/Check4De...

Firebase

Project Overview

Développeur

- Authentication
- Database
- Storage
- Hosting
- Functions
- ML Kit

Qualité

Crashlytics, Performance, Test La...

Données analytiques

Dashboard, Events, Conversions, A...

Enrichir

Predictions, A/B Testing, Cloud M...

Extensions

Spark

Gratuit 0 \$/mois Changer de formule

NotesMobileMaster

Database Cloud Firestore

Données Règles Index Utilisation

Etudiants > RvCPwylo3opF...

notesmobilemaster	Etudiants	RvCPwylo3opFVBfrVdC6
+ Commencer une collection	+ Ajouter un document	+ Commencer une collection
Etudiants >	RvCPwyIo3opFVBfrVdC6 >	+ Ajouter un champ
	cS2w5uVvEHws3zI6RBG6	Adresse : "Kebemer"
	zhsDnFYCL2ZPIxMC2RYW	Age : "97"
		Nom : "Abdoulaye WADE"

# Exemple avec Firestore Database



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

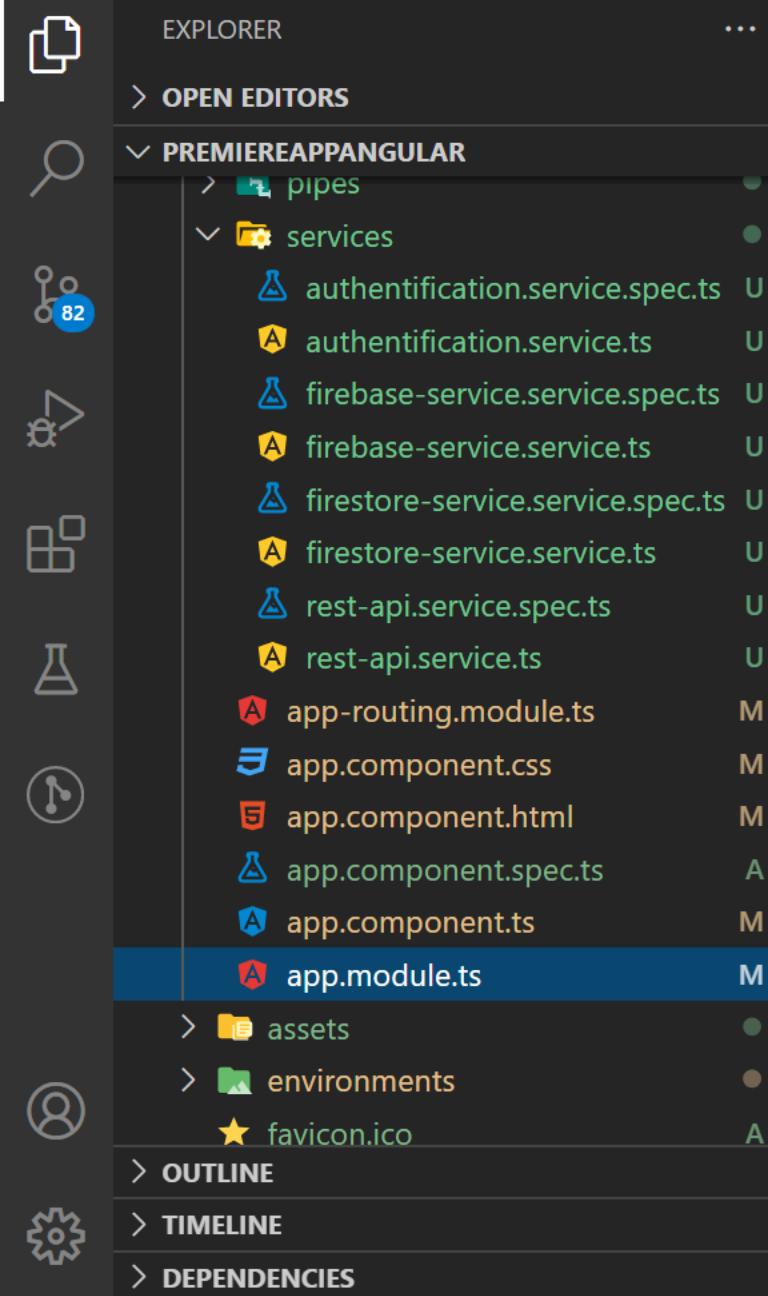
1: powershell

+ □ └ ┌ ×

Windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular> cd .\src\app\services\  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services> ng g s firestore-service  
CREATE src/app/services/firestore-service.service.spec.ts (408 bytes)  
CREATE src/app/services/firestore-service.service.ts (145 bytes)  
...  
PS C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular\src\app\services>
```



app.module.ts

```
src > app > A app.module.ts > AppModule
  import { NgModule } from '@angular/core';
  import { BrowserModule } from '@angular/platform-browser';
  import { AppComponent } from './app.component';
  import { AccueilComponent } from './accueil/accueil.component';
  import { LoginComponent } from './login/login.component';
  import { GetStringAvantTiretPipe } from './pipes/get-string-avant-tiret.pipe';
  import { EntetePageComponent } from './entete-page/entete-page.component';
  import { PiedPageComponent } from './pied-page/pied-page.component';
  import { ErrorComponent } from './error/error.component';
  import { AppRoutingModule } from './app-routing.module';
  import { HttpClientModule } from '@angular/common/http';
  import { FormsModule } from '@angular/forms';
  import { ReactiveFormsModule } from '@angular/forms';
  import { AngularFirestoreModule } from '@angular/fire/firestore';
  import { AngularFireModule } from '@angular/fire';
  import { AngularFireAuthModule } from '@angular/fire/auth';
  import { AngularFireDatabaseModule } from '@angular/fire/database';
  import { AuthenticationService } from './services/authentication.service';
  import { FirestoreServiceService } from './services/firestore-service.service';

  @NgModule({
    declarations: [
      AppComponent,
      AccueilComponent,
      LoginComponent,
      GetStringAvantTiretPipe,
      EntetePageComponent,
      PiedPageComponent,
      ErrorComponent,
    ],
    imports: [
      BrowserModule,
      AppRoutingModule,
      FormsModule,
      HttpClientModule,
      ReactiveFormsModule,
      AngularFirestoreModule.initializeApp(environment.firebaseio),
      AngularFireModule,
      AngularFireAuthModule,
      AngularFireDatabaseModule,
    ],
    providers: [AuthenticationService, FirestoreServiceService],
    bootstrap: [AppComponent],
  })
}
```

EXPLORER

> OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR

- > error
- > login
- > modeles
- > pied-page
- > pipes
- > services
  - authentication.service.spec.ts
  - authentication.service.ts
  - firebase-service.service.spec.ts
  - firebase-service.service.ts
  - firestore-service.service.spec.ts
  - firestore-service.service.ts
  - rest-api.service.spec.ts
  - rest-api.service.ts
- > app-routing.module.ts
- > app.component.css
- > app.component.html
- > app.component.spec.ts
- > app.component.ts
- > app.module.ts
- > assets
- > environments
- > favicon.ico
- > index.html

> OUTLINE

> TIMELINE

> DEPENDENCIES

A firestore-service.service.ts X A app.module.ts A accueil.component.ts ●

src > app > services > A firestore-service.service.ts > ...

```
1 import { Injectable } from '@angular/core';
2 import { AngularFirestore, AngularFirestoreCollection } from '@angular/fire/firestore';
3 import { RendezVous } from '../modeles/rendez-vous';
4
5 @Injectable({
6   providedIn: 'root',
7 })
8 export class FirestoreServiceService {
9   private dbPath = 'rvs';
10
11   rvs: AngularFirestoreCollection<RendezVous> = null;
12   constructor(private firestoreBD: AngularFirestore) {
13     this.rvs = this.firestoreBD.collection(this.dbPath);
14   }
15
16   creerUnRV(rv: RendezVous) {
17     return this.rvs.add(JSON.parse(JSON.stringify(rv)));
18   }
19
20   getAllRvs() {
21     return this.rvs;
22   }
23
24   update(key: string, rv: RendezVous): Promise<void> {
25     return this.firestoreBD.doc(this.dbPath + '/' + key).update(rv);
26   }
27
28   deleteRv(key: string): Promise<void> {
29     return this.firestoreBD.doc(this.dbPath + '/' + key).delete();
30   }
31 }
```

Exemple FireStore - Exemple Fire +

console.firebaseio.google.com/project/exemple-firebase/overview

Firebase

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

: Spark  
Gratuit 0 \$/mois   Changer de formule

Exemple FireStore

Exemple FireStore Formule Spark

com.example... | Ajouter une application

Qualité

Crashlytics | Crash-free users

Stockez et synchronisez les données de votre application en quelques millisecondes

Accéder à la documentation

Extensions

Spark

Changer de formule

Star

Upload

Document

Sync

Copy

Close

Help

Profile

# Cloud Firestore

Données Rules Indexes Usage

exemple-firestore > enregistrement > MRXdPubAr6IGnKfNGzTa

exemple-firestore	enregistrement	MRXdPubAr6IGnKfNGzTa
+ Commencer une collection	+ Ajouter un document	+ Commencer une collection
enregistrement	MRXdPubAr6IGnKfNGzTa	SFdp6y955UTCyrKYExfZ TDdhK5huDRftk6lNVffS a4Bv2MHDBzE2s29h3Xdk kt9TpGvEEVybUQ70jFnx 17YKDVejD0OysYQdVMBf

Exemple FireStore - Cloud Firestore

console.firebaseio.google.com/project/exemple-firebase/firestore/data~2Fenregistrement~2FMRXdPubAr6lGnKfNGzTa

Firebase

Exemple FireStore

Accéder à la documentation

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Extensions

: Spark

Gratuit 0 \$/mois

Changer de formule

# Cloud Firestore

Données Rules Indexes Usage

Commencer une collection

1 Spécifier un ID pour la collection      2 Ajouter le premier document

Chemin parent

/

ID de collection ②

rVS

Annuler Suivant

Cloud Firestore

enregistrement

exemple-firebase

Commencer une collection

enregistrement

GzTa

lection

2020 à 12:54:43 UTC

## Cloud Firestore

Données Rules Indexes Usages

## Commencer une collection

-  Spécifier un ID pour la collection

Chemin parent du document

## **+** Commencer ur

ID du document

ID généré automatiquement

Champ	Type	Valeur
	= string	

Annule

Enregistre

EXPLORER

1 OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGULAR

- rendez-vous.spec.ts
- rendez-vous.ts **U**
- utilisateur.spec.ts **U**
- utilisateur.ts **U**
- pied-page
- pipes
- services
- app-routing.module.ts **M**
- app.component.css **M**
- app.component.html **M**
- app.component.spec.ts **A**
- app.component.ts **M**
- app.module.ts **M**
- assets

OUTLINE

TIMELINE

DEPENDENCIES

app.module.ts accueil.component.ts rendez-vous.ts

src > app > modeles > rendez-vous.ts > RendezVous > toJson

```
1 export class RendezVous {  
2   key: string;  
3   description: string;  
4   lieu: string;  
5   date: String;  
6  
7   toJson(){  
8     return JSON.stringify(this);  
9   }  
10 }  
11 }
```

EXPLORER

> OPEN EDITORS

PREMIEREAPPANGULAR

- accueil.component.html
- accueil.component.spec.ts
- accueil.component.ts**
- > entete-page
- > error
- > login
- modeles
  - rendez-vous.spec.ts
  - rendez-vous.ts
  - utilisateur.spec.ts
  - utilisateur.ts
- > pied-page
- > pipes
- services
  - authentification.service.spec.ts
  - authentification.service.ts
  - firebase-service.service.spec.ts
  - firebase-service.service.ts
  - firestore-service.service.spec.ts
  - firestore-service.service.ts
  - rest-api.service.spec.ts
  - rest-api.service.ts
- app-routing.module.ts
- app.component.css

> OUTLINE

> TIMELINE

> DEPENDENCIES

A accueil.component.ts X

src > app > accueil > A accueil.component.ts > AccueilComponent > chargerLesRvs

```
48     constructor(  
49       private authenticationService: AuthentificationService,  
50       private routeActive: ActivatedRoute,  
51       private router: Router,  
52       private firestoreServiceService: FirestoreServiceService  
53     ) {}  
54     ngOnInit(): void {  
55       this.chargerLesRvs();  
56       if (this.authenticationService.utilisateurConnecte())  
57         this.nomUtilisateur = this.routeActive.snapshot.paramMap.get(  
58           'nomUtilisateur'  
59         );  
60       else this.router.navigateByUrl('/login');  
61     }  
  
63     chargerLesRvs() {  
64       this.firestoreServiceService  
65         .getAllRvs()  
66         .snapshotChanges()  
67         .pipe(  
68           map(changes =>  
69             changes.map(c =>  
70               ({ key: c.payload.doc.id, ...c.payload.doc.data() })  
71             )  
72           )  
73         .subscribe((data) => {  
74           this.rendezVous = data;  
75         });  
76       this.rvSaisie = new RendezVous();  
77     }  
78   }
```

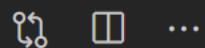
# Créer un nouveau rendez-vous

```
addRv() {  
    this.firestoreServiceService.creerUnRV(this.rvSaisie);  
    this.chargerLesRvs();  
}
```



login.component.html

accueil.component.html



```
src > app > accueil > accueil.component.html > div.container > div.container.custom-container > div.col-md-12 > div.form-group > button.btn.btn-success
  6   <div class="container custom-container">
  7     <div class="col-md-12">
  8       <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
  9       <div class="form-group" *ngIf="update">
 10         <input type="text" [disabled]="update" [(ngModel)]="rvSaisie.key" class="form-control" placeholder="Id">
 11       </div>
 12       <div class="form-group">
 13         <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
 14       </div>
 15       <div class="form-group">
 16         <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
 17       </div>
 18       <div class="form-group">
 19         <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
 20       </div>
 21       <div class="form-group" *ngIf="!update">
 22         <button class="btn btn-success btn-lg btn-block" (click)="addRv()">Ajouter un nouveau RV </button>
 23       </div>
 24       <div class="form-group" *ngIf="update">
 25         <button class="btn btn-success btn-lg btn-block" (click)="mettreAJour()">
 26           Mettre à jour un nouveau RV
 27         </button>
 28       </div>
 29     </div>
 30   </div>
 31   <div class="table-responsive-sm">
```



# Mettre à jour un rendez-vous

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** accueil.component.ts - premiereAppAngular - Visual Studio Code.
- Explorer Bar (Left):**
  - OPEN EDITORS:** accueil.component.html, accueil.component.spec.ts, accueil.component.ts (selected).
  - PREMIEREAPPANGULAR:** 82 files. Subfolders: entete-page, error, login, modeles, pied-page, pipes. Services: authentication.service.spec.ts, authentication.service.ts, firebase-service.service.spec.ts, firebase-service.service.ts, firestore-service.service.spec.ts, firestore-service.service.ts, rest-api.service.spec.ts, rest-api.service.ts.
- Code Editor (Right):** The code for `accueil.component.ts`. The code handles updating a rendezvous (RendezVous) object. It includes methods `updateRv` and `mettreAJour`, and a constructor that injects an `authentificationService`.
- Bottom Status Bar:** master\*, 0 ▲ 0, Compile Hero: Off, Ln 36, Col 2, Spaces: 2, UTF-8, LF, TS 4.0.3, Prettier: ✓, 539.

# Mettre à jour

accueil.component.html - premiereAppAngular - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED

PREMIEREAPPANGU... 80

src > app > accueil > accueil.component.html

```
32 <h1>Liste des Rendez-Vous</h1>
33 <table class="table">
34   <thead class="thead-light">
35     <tr>
36       <th>Id</th>
37       <th>Description</th>
38       <th>Lieu</th>
39       <th>Date</th>
40     </tr>
41   </thead>
42   <tbody>
43     <tr *ngFor="let rv of rendezVous">
44       <td>{{rv.key}}</td>
45       <td>{{rv.description}}</td>
46       <td>{{rv.lieu | uppercase}}</td>
47       <td>{{rv.date}}</td>
48       <td>
49         <button class="btn btn-success" (click)="updateRv(rv)">Editer </button>
50         <button class="btn btn-danger" type="button" (click)="deleteRv(rv)">
51           Supprimer
52         </button>
53       </td>
54     </tr>
55   </tbody>
56 </table>
```

File Edit Selection View Go Run Terminal Help

Profile master\*+ 0 △ 0

Compile Hero: Off Ln 25, Col 86 (11 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓

540

# Mettre à jour

```
v class="container">
  <div class="alert alert-success" role="alert">
    <h1 class="alert-heading">Bienvenue {{nomUtilisateur}}</h1>
  </div>

<div class="container custom-container">
  <div class="col-md-12">
    <h3 class="mb-3 text-center">Créer un Rendez-Vous</h3>
    <div class="form-group" *ngIf="update">
      <input type="text" [disabled]="update" [(ngModel)]="rvSaisie.key" class="form-control" placeholder="Id">
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="rvSaisie.description" class="form-control" placeholder="Description">
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="rvSaisie.lieu" class="form-control" placeholder="Lieu">
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="rvSaisie.date" class="form-control" placeholder="Date">
    </div>
    <div class="form-group" *ngIf="!update">
      <button class="btn btn-success btn-lg btn-block" (click)="addRv()">Ajouter un nouveau RV </button>
    </div>
    <div class="form-group" *ngIf="update">
      <button class="btn btn-success btn-lg btn-block" (click)="mettreAJour()">Mettre à jour un nouveau RV
      </button>
    </div>
  </div>
</div>

<h1>Liste des Rendez-Vous</h1>
```

File Edit Selection View Go Run Terminal Help accueil.component.html - premiereAppAngular - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED PREMIEREAPPANGU... 80 src app accueil accueil.component.css accueil.component.html accueil.component.spec.ts accueil.component.ts entete-page error login modeles rendez-vous.spec.ts rendez-vous.ts utilisateur.spec.ts utilisateur.ts pied-page pipes services

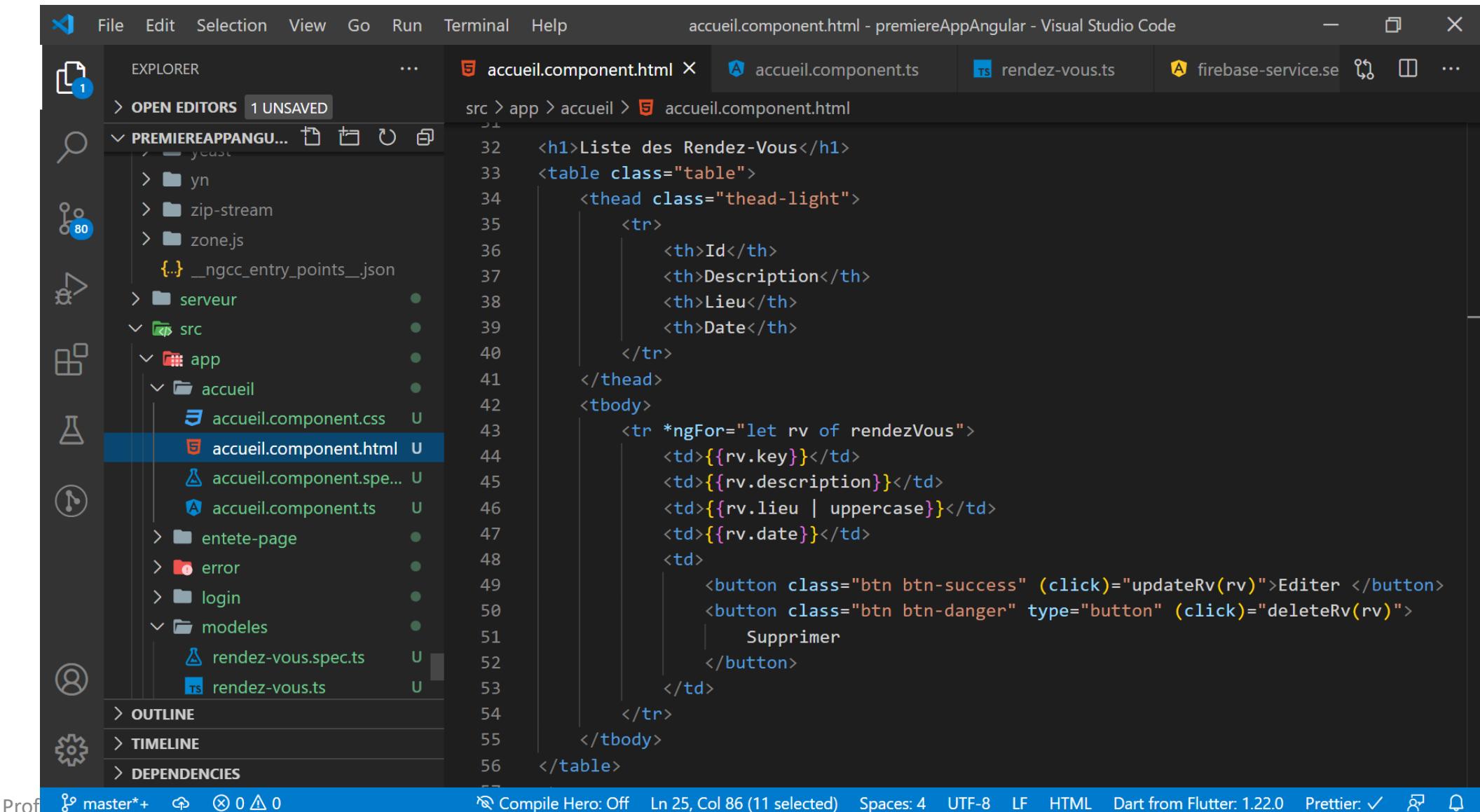
OUTLINE TIMELINE DEPENDENCIES

master+ 80 △ 0 Compile Hero: Off Ln 25, Col 86 (11 selected) Spaces: 4 UTF-8 LF HTML Dart from Flutter: 1.22.0 Prettier: ✓ 541

# Supprimer un rendez-vous

```
// Delete Rendez-vous
deleteRv(rv:RendezVous) {
    if (
        window.confirm('Etes-vous sûr que vous voulez supprimer cet élément ?')
    ) {
        // this.firebaseioServiceService.delete(rv.key);
        this.firestoreServiceService.deleteRv(rv.key);
        this.chargerLesRvs();
    }
}
```

# Suppression, Editer



The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** accueil.component.html - premiereAppAngular - Visual Studio Code.
- Explorer Sidebar:** Shows the project structure under "PREMIEREAPPANGU...". The "src/app/accueil" folder is expanded, showing "accueil.component.css", "accueil.component.html", "accueil.component.spec.ts", and "accueil.component.ts". Other files like "yeast", "yn", "zip-stream", "zone.js", "server", and "modeles" are also listed.
- Editor Area:** Displays the content of "accueil.component.html". The code uses Angular's \*ngFor directive to iterate over a list of rendez-vous (rv) and render a table row for each. Each row contains four columns: Id, Description, Lieu, and Date. The last column (Date) includes buttons for "Editer" (Edit) and "Supprimer" (Delete).
- Bottom Status Bar:** Shows "master\*+", "Compile Hero: Off", "Ln 25, Col 86 (11 selected)", "Spaces: 4", "UTF-8", "LF", "HTML", "Dart from Flutter: 1.22.0", "Prettier: ✓", and "543".

```
src > app > accueil > accueil.component.html
32   <h1>Liste des Rendez-Vous</h1>
33   <table class="table">
34     <thead class="thead-light">
35       <tr>
36         <th>Id</th>
37         <th>Description</th>
38         <th>Lieu</th>
39         <th>Date</th>
40     </tr>
41   </thead>
42   <tbody>
43     <tr *ngFor="let rv of rendezVous">
44       <td>{{rv.key}}</td>
45       <td>{{rv.description}}</td>
46       <td>{{rv.lieu | uppercase}}</td>
47       <td>{{rv.date}}</td>
48       <td>
49         <button class="btn btn-success" (click)="updateRv(rv)">Editer </button>
50         <button class="btn btn-danger" type="button" (click)="deleteRv(rv)">
51           Supprimer
52         </button>
53       </td>
54     </tr>
55   </tbody>
56 </table>
```



Firebase



# Déploiement

Déployer votre application web  
Angular sur Firebase

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- **Déploiement Application web Angular sur Firebase**
- Authentification et autorisation



# 05 étapes pour déployer l'application

1. Construire la version à déployer: **ng build --prod**
2. Installer Firebase Command Line Interface (CLI)  
**npm install -g firebase-tools**
3. Se loguer à la console firebase: **firebase login**
4. Connecter le projet local et celui de firebase: **firebase init**
5. Déployer l'application dans firebase: **firebase deploy**

# Se positionner dans le dossier du projet Angular

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [version 10.0.19041.572]
(c) 2020 Microsoft Corporation. Tous droits réservés.

C:\Users\ousma>cd Desktop

C:\Users\ousma\Desktop>cd workspace-angular

C:\Users\ousma\Desktop\workspace-angular>cd premiereAppAngular

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

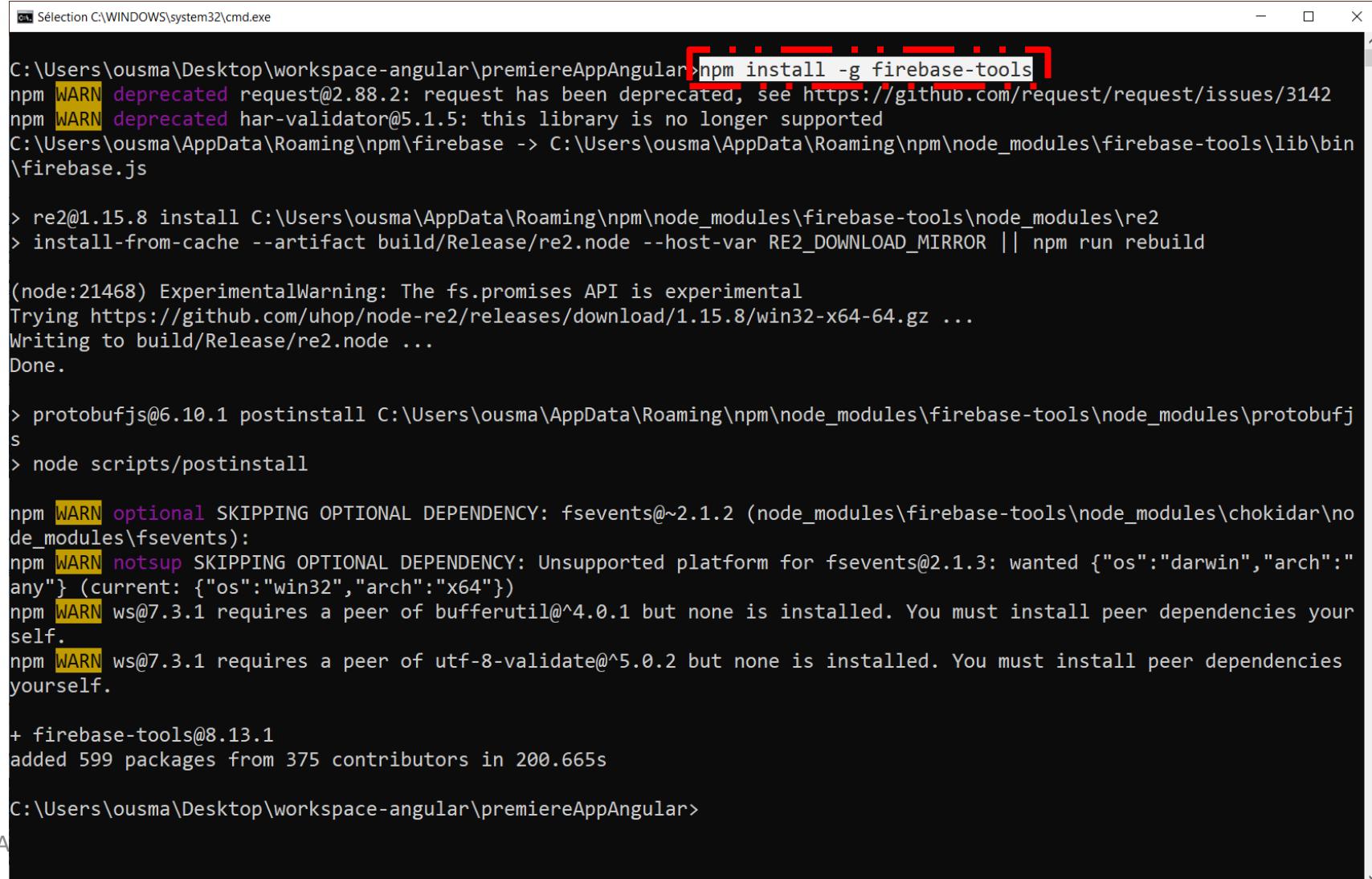
# Etape 1: construire la version déployable de votre application *ng build --prod*

```
C:\WINDOWS\system32\cmd.exe
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>ng build --prod

chunk {} runtime.acf0dec4155e77772545.js (runtime) 1.45 kB [entry] [rendered]
chunk {1} main.c6a312ecaa72253fa5d0.js (main) 281 kB [initial] [rendered]
chunk {2} polyfills.35a5ca1855eb057f016a.js (polyfills) 36 kB [initial] [rendered]
chunk {3} styles.7e1aad8413b8f069a75c.css (styles) 141 kB [initial] [rendered]
Date: 2020-10-22T23:53:16.778Z - Hash: adfdf33b54657d8765e0 - Time: 60357ms

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

# Etape 2 : Installer globalement Firebase CLI ***firebase-tools***



```
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>npm install -g firebase-tools
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\ousma\AppData\Roaming\npm\firebase -> C:\Users\ousma\AppData\Roaming\npm\node_modules\firebase-tools\lib\bin\firebase.js

> re2@1.15.8 install C:\Users\ousma\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\re2
> install-from-cache --artifact build/Release/re2.node --host-var RE2_DOWNLOAD_MIRROR || npm run rebuild

(node:21468) ExperimentalWarning: The fs.promises API is experimental
Trying https://github.com/uop/node-re2/releases/download/1.15.8/win32-x64-64.gz ...
Writing to build/Release/re2.node ...
Done.

> protobufjs@6.10.1 postinstall C:\Users\ousma\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\protobufjs
> node scripts/postinstall

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\firebase-tools\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN ws@7.3.1 requires a peer of bufferutil@^4.0.1 but none is installed. You must install peer dependencies yourself.
npm WARN ws@7.3.1 requires a peer of utf-8-validate@^5.0.2 but none is installed. You must install peer dependencies yourself.

+ firebase-tools@8.13.1
added 599 packages from 375 contributors in 200.665s

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

# Etape 3 : Se loguer à firebase *firebase login*

```
C:\WINDOWS\system32\cmd.exe
C:\Users\admin\Desktop\AppsIonicUVS\correctionTP3>firebase login
!  JSON error trying to load C:\Users\admin\Desktop\AppsIonicUVS\correctionTP3\.firebaserc
i  Firebase optionally collects CLI usage and error reporting information to help improve our
products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? Yes
i  To change your data collection preference at any time, run `firebase logout` and log in ag
ain.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client\_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response\_type=code&state=634223168&redirect\_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
+  Success! Logged in as osall@univ-thies.sn

C:\Users\admin\Desktop\AppsIonicUVS\correctionTP3>
```



```
C:\WINDOWS\system32\cmd.exe
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>firebase login
i  Firebase optionally collects CLI usage and error reporting information to help improve
our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? Yes
i  To change your data collection preference at any time, run `firebase logout` and log in
again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client\_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response\_type=code&state=645807385&redirect\_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
+  Success! Logged in as osall1751@gmail.com

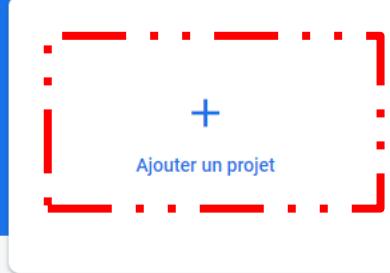
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```



console.firebaseio.google.com/?gclid=Cj0KCQjw28T8BRDbARIsAEOMBcxMqvbjgXoPbjhOVHwZCC6-A4CaNpW7TnIEuqw8cDeZSwRniD... Accéder à la documentation Profile icon

Firebase

## Your Firebase projects



Ajouter un projet

**firebaseflutter**  
fir-flutter-17f20

**Exemple FireStore**  
exemple-firebase

**testAndroid**  
testandroid-20ff2

**exosionic**  
exosionic

**NotesMobileMaster**  
notesmobilemaster

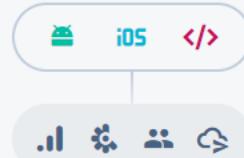
Voir un projet de démo



Les projets Firebase sont des conteneurs pour vos applications

Les applications d'un projet partagent des fonctionnalités telles que  
Realtime Database et Analytics

En savoir plus



← → ⌛ 🔍 ☆ ⚡ 📁 🗃 🗑 🗑 🗑 🗑 🗑 🗑 🗑 🗑

console.firebaseio.google.com/?gclid=Cj0KCQjw28T8BRDbARIsAEOMBcxMqvbjgXoPbjhOVHwZCC6-A4CaNpW7TnIEuqw8cDeZSwRniD...

Créer un projet(Étape 1)

Un projet Firebase est un conteneur pour vos applications iOS, Android et Web.

Les applications dans ce projet partageront des fonctionnalités comme Database, la gestion des utilisateurs et Remote Config.

Un projet Firebase est un projet Cloud.

Lorsque vous créez un projet Firebase dans la console Firebase, c'est en fait un projet Google Cloud Platform (GCP) que vous créez en coulisses.

[En savoir plus](#)

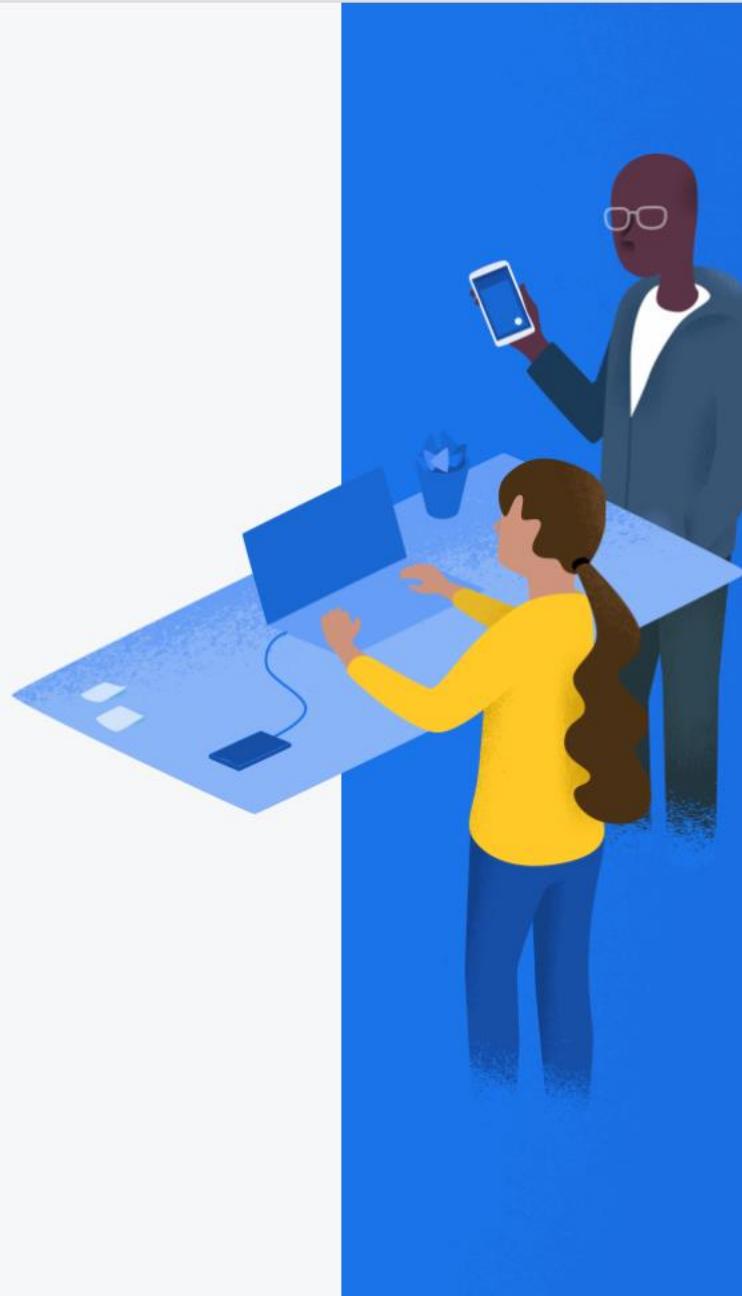
Commencez par donner un nom à votre [projet](#) <sup>?</sup>

Nom du projet

demoAngular

[demoangular-4c77e](#)

Continuer



console.firebaseio.google.com/project/demoangular-4c77e/overview

Firebase

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Analytics

Dashboard, Events, Conversions, Au...

Enrichir

Predictions, A/B Testing, Cloud Mes...

Extensions

: Spark

Gratuit 0 \$/mois

Changer de formule

demoAngular

demoAngular Formule Spark

# Lancez-vous en ajoutant Firebase à votre application

iOS Android </> | ↴ Ajoutez une application pour démarrer

Stockez et synchronisez les données de votre application en quelques millisecondes

**Authentication**  
Authentifiez et gérez les utilisateurs

**Cloud Firestore**  
Mises à jour en temps réel, requêtes puissantes et scaling automatique

Afficher toutes les fonctionnalités Développer

**Firebase**

[Vue d'ensemble du projet](#) [⚙️](#)

**Développer**

- [Authentication](#)
- [Cloud Firestore](#)
- [Realtime Database](#)
- [Storage](#)
- [Hosting](#)
- [Functions](#)
- [Machine Learning](#)

**Qualité**

- [Crashlytics](#)
- [Performance](#)
- [Test Lab](#)
- [App Distribution](#)

**Extensions**

: Spark Gratuit 0 \$/mois [Changer de formule](#)

demoAngular

[Accéder à la documentation](#)

# Hosting

Déployez des applications Web et mobiles en quelques secondes seulement en utilisant un réseau global de diffusion de contenu sécurisé

[Commencer](#)[En savoir plus](#)

Que dois-je faire pour commencer ?  
[Afficher les documents](#)



Comment fonctionne Hosting ?  
[Afficher les documents](#)

Introducing Firebase Hosting

À regarder... Partager

Accéder à la documentation

## Set up Firebase Hosting

### 1 Install Firebase CLI

To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).

Exécutez la commande [npm](#) pour installer la CLI ou la mettre à jour.

```
$ npm install -g firebase-tools
```



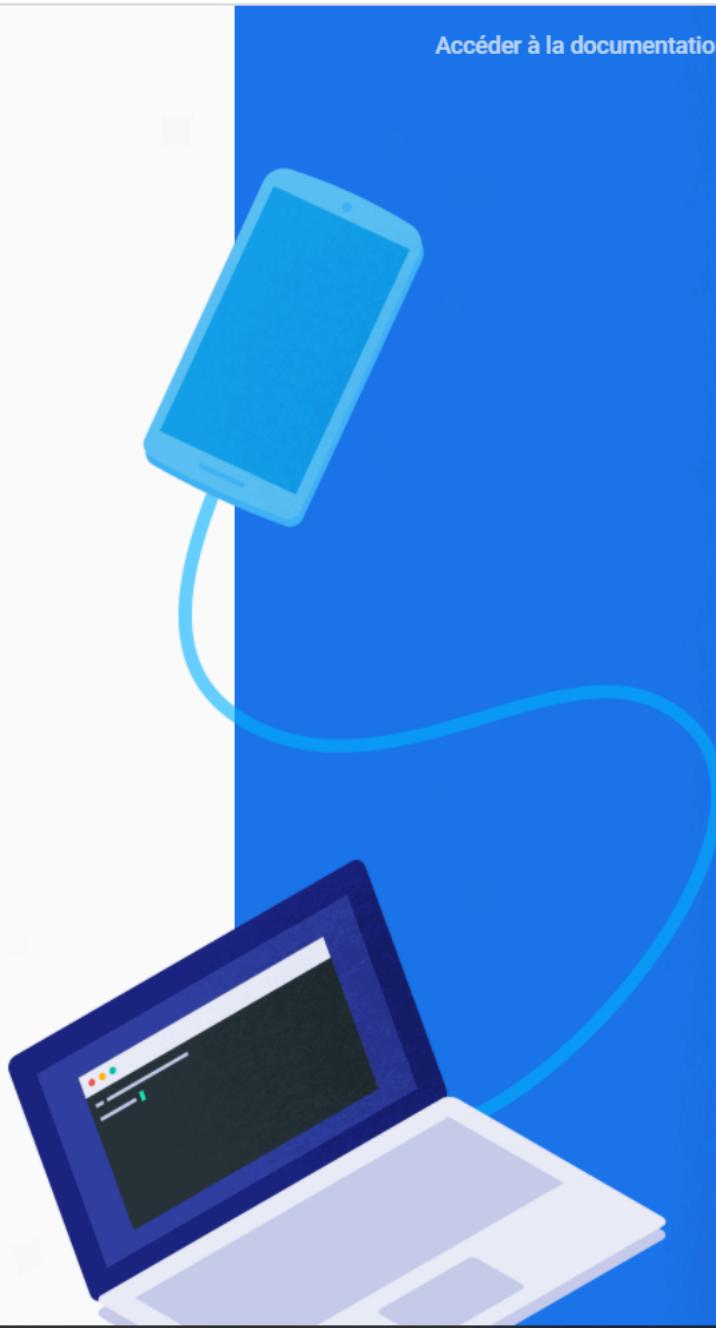
Un problème de fonctionnement ? Consultez la [documentation de référence de la CLI Firebase](#) ou modifiez vos [autorisations npm](#).

- Afficher également la procédure permettant d'ajouter le SDK Firebase JavaScript à mon application Web  
Le SDK inclut Cloud Firestore, Authentication, Performance Monitoring et bien plus encore.  
Vous pouvez l'ajouter maintenant ou ultérieurement.

**Suivant**

### 2 Initialize your project

### 3 Deploy to Firebase Hosting



solutionsexosonic - Console Firel +

console.firebaseio.google.com/u/1/project/solutionsexosonic/hosting

Fact Checking Bb Collaborate How to Host Your I... Cours : Développe... Salle de Cours et Tr... TRADUCTION DE A... Améliorez la structu... PowerShell - Active... nlp - Lemmatizatio... Formulaire Tableau...

Accéder à la documentation

## Set up Firebase Hosting

- ✓ Install Firebase CLI
- ✓ Initialize your project
- 3 Deploy to Firebase Hosting

Lorsque vous êtes prêt, déployez votre application Web  
Placez vos fichiers statiques (par exemple, HTML, CSS, JS) dans le répertoire de déploiement de votre application (le paramètre par défaut est "public"), puis exécutez cette commande depuis le répertoire racine de l'application :

```
$ firebase deploy
```

Une fois votre application déployée, vous pourrez la voir à l'adresse  
[solutionsexosonic.web.app](https://solutionsexosonic.web.app) ↗  
Besoin d'aide ? Consultez la [documentation Hosting](#) ↗

Accéder à la console



Accéder à la documentation

## Set up Firebase Hosting

- ✓ Install Firebase CLI

2 Initialize your project

Ouvrez une fenêtre de terminal et accédez à un répertoire racine pour votre application Web, ou créez-en un

Se connecter à Google

```
$ firebase login
```



Lancez votre projet

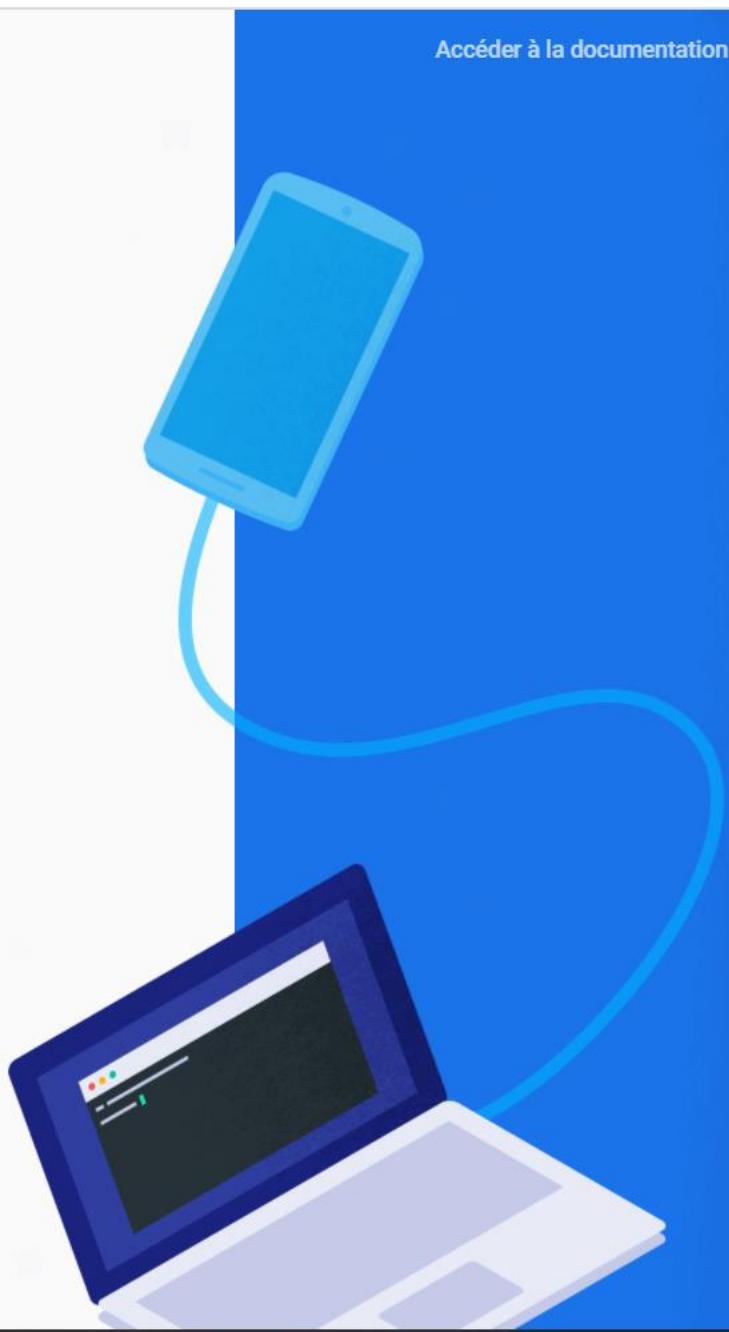
Exécutez la commande suivante à partir du répertoire racine de votre application :

```
$ firebase init
```

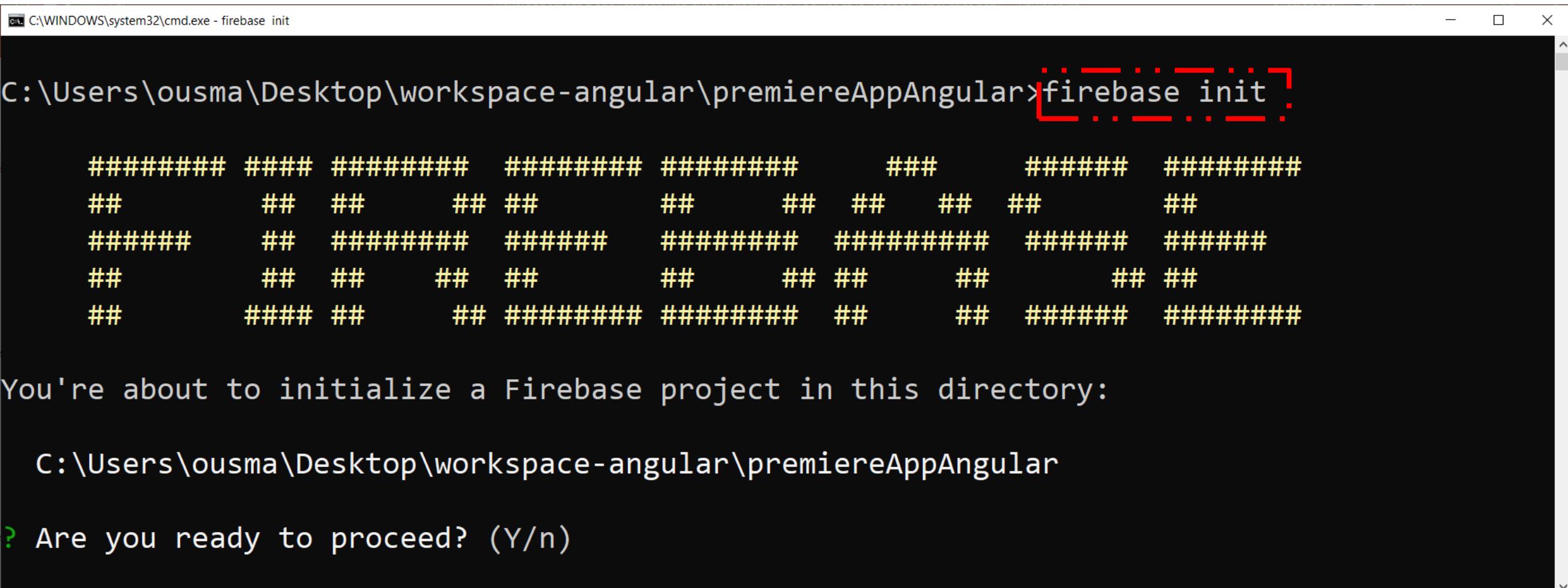


Suivant

3 Deploy to Firebase Hosting



# Etape 4: Connecter le projet local à celui de firebase avec la commande ***firebase init***



```
C:\WINDOWS\system32\cmd.exe - firebase init  
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>firebase init  
##### #### ##### #### ##### ##### ##### #####  
## ## ## ## ## ## ## ## ## ## ## ## ## ##  
##### ## ##### ##### ## ##### ##### ##### #####  
## ## ## ## ## ## ## ## ## ## ## ## ## ##  
## ## ## ## ## ## ## ## ## ## ## ## ## ##  
  
You're about to initialize a Firebase project in this directory:  
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular  
? Are you ready to proceed? (Y/n)
```

# Etape 4: Connecter le projet local à celui de firebase avec la commande ***firebase init***

```
C:\WINDOWS\system32\cmd.exe - firebase init

You're about to initialize a Firebase project in this directory:
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular

? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection)
  ( ) Database: Deploy Firebase Realtime Database Rules
  ( ) Firestore: Deploy rules and create indexes for Firestore
  ( ) Functions: Configure and deploy Cloud Functions
  ( ) Hosting: Configure and deploy Firebase Hosting sites
  ( ) Storage: Deploy Cloud Storage security rules
  ( ) Emulators: Set up local emulators for Firebase features
  ( ) Remote Config: Get, deploy, and rollback configurations for Remote Config
```

# Etape 4: Connecter le projet local à celui de firebase avec la commande ***firebase init***

```
C:\WINDOWS\system32\cmd.exe - firebase init
? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. Hosting: Configure and deploy Firebase Hosting sites

== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)
> Use an existing project
Create a new project
Add Firebase to an existing Google Cloud Platform project
Don't set up a default project
```

# Etape 4: Connecter le projet local à celui de firebase avec la commande ***firebase init***

```
C:\WINDOWS\system32\cmd.exe - firebase init
sting sites

==== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
> demoangular-4c77e (demoAngular)
exemple-firebase (Exemple FireStore)
exosionic (exosionic)
fir-flutter-17f20 (firebaseflutter)
notesmobilemaster (NotesMobileMaster)
testandroid-20ff2 (testAndroid)
```

# Etape 4: Connecter le projet local à celui de firebase avec la commande `firebase init`

```
C:\WINDOWS\system32\cmd.exe
? Please select an option: Use an existing project
? Select a default Firebase project for this directory: demoangular-4c77e (demoAngular)
i  Using project demoangular-4c77e (demoAngular)

==> Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist/premiereAppAngular
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
? File dist/premiereAppAngular/index.html already exists. Overwrite? No
i  Skipping write of dist/premiereAppAngular/index.html

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

+ Firebase initialization complete!

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

## Set up Firebase Hosting

Install Firebase CLI

Initialize your project

**3** Deploy to Firebase Hosting

Lorsque vous êtes prêt, déployez votre application Web

Placez vos fichiers statiques (par exemple, HTML, CSS, JS) dans le répertoire de déploiement de votre application (le paramètre par défaut est "public"), puis exécutez cette commande depuis le répertoire racine de l'application :

```
$ firebase deploy
```

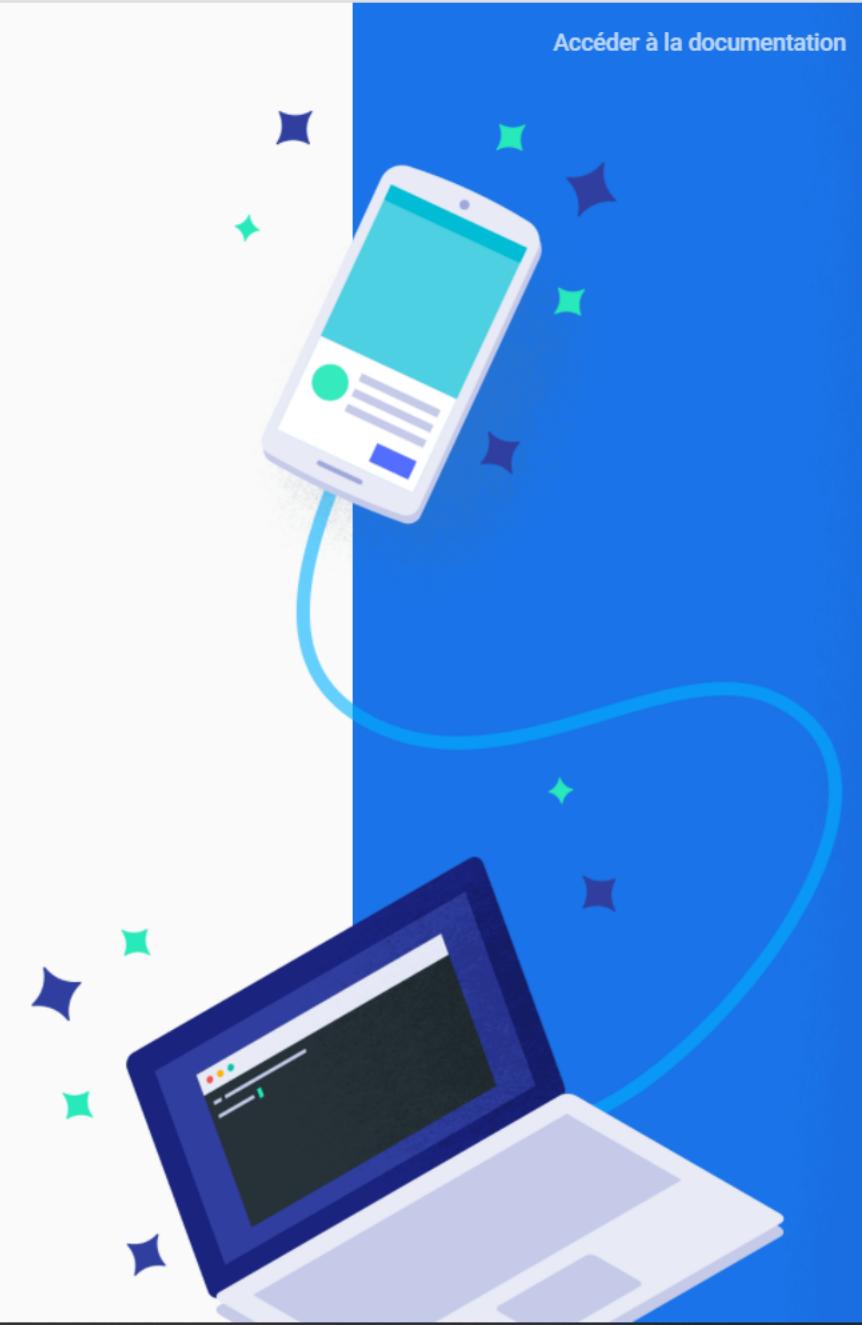


Une fois votre application déployée, vous pourrez la voir à l'adresse

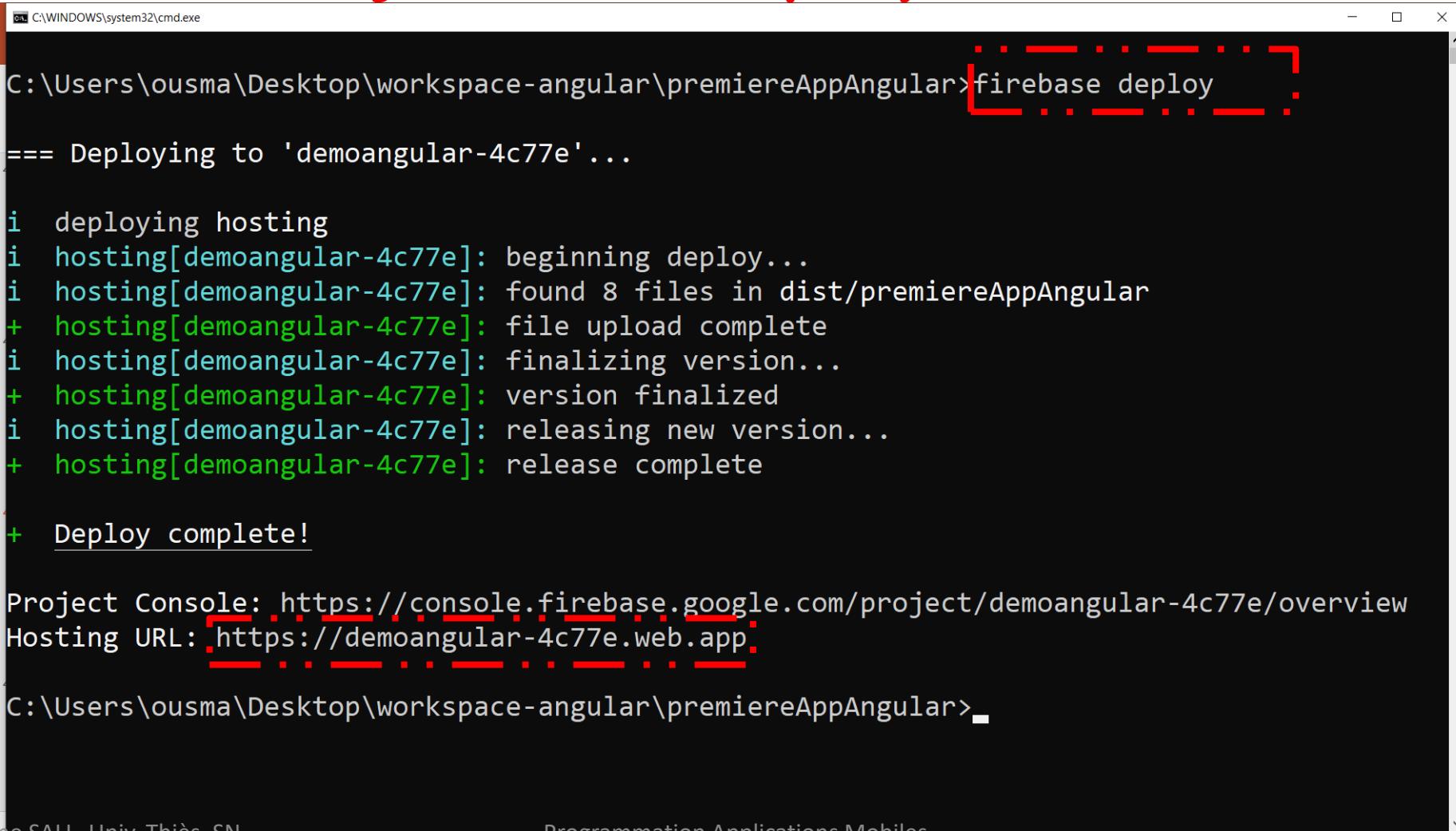
[demoangular-4c77e.web.app](https://demoangular-4c77e.web.app)

Besoin d'aide ? Consultez la [documentation Hosting](#)

[Accéder à la console](#)



# Etape 5: Déployer l'application dans le firebase avec *firebase deploy*



```
C:\WINDOWS\system32\cmd.exe
C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>firebase deploy
[...]
==== Deploying to 'demoangular-4c77e'...

i  deploying hosting
i  hosting[ demoangular-4c77e]: beginning deploy...
i  hosting[ demoangular-4c77e]: found 8 files in dist/premiereAppAngular
+ hosting[ demoangular-4c77e]: file upload complete
i  hosting[ demoangular-4c77e]: finalizing version...
+ hosting[ demoangular-4c77e]: version finalized
i  hosting[ demoangular-4c77e]: releasing new version...
+ hosting[ demoangular-4c77e]: release complete

+ Deploy complete!

Project Console: https://console.firebaseio.google.com/project/demoangular-4c77e/overview
Hosting URL: https://demoangular-4c77e.web.app

C:\Users\ousma\Desktop\workspace-angular\premiereAppAngular>
```

Accéder à la documentation

## Set up Firebase Hosting

Install Firebase CLI

Initialize your project

Deploy to Firebase Hosting

Lorsque vous êtes prêt, déployez votre application Web

Placez vos fichiers statiques (par exemple, HTML, CSS, JS) dans le répertoire de déploiement de votre application (le paramètre par défaut est "public"), puis exécutez cette commande depuis le répertoire racine de l'application :

```
$ firebase deploy
```

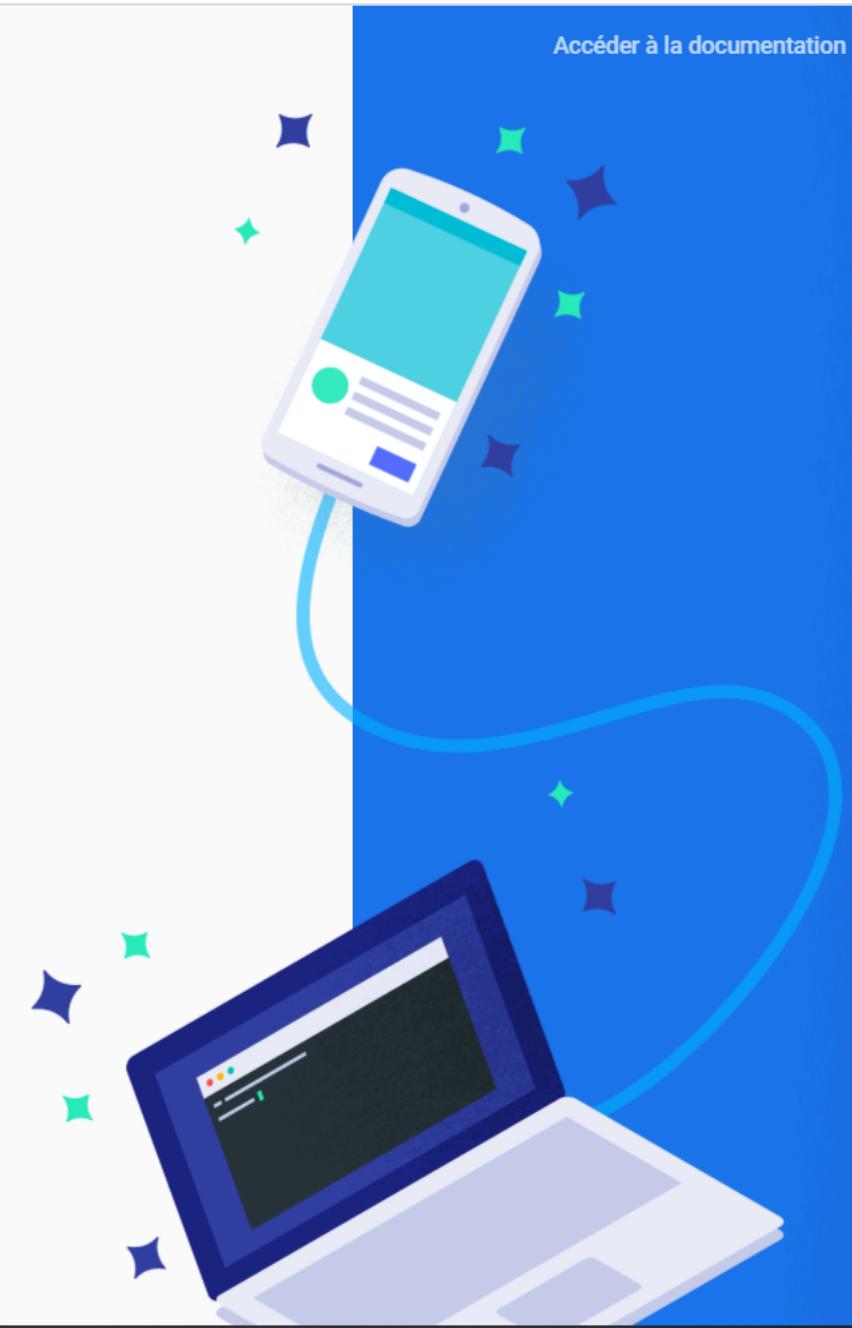


Une fois votre application déployée, vous pourrez la voir à l'adresse

[demoangular-4c77e.web.app](https://demoangular-4c77e.web.app)

Besoin d'aide ? Consultez la [documentation Hosting](#)

[Accéder à la console](#)



Firebase

Vue d'ensemble du projet

Développer

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Qualité

- Crashlytics
- Performance
- Test Lab
- App Distribution

Extensions

: Spark  
Gratuit 0 \$/mois      Changer de formule

demoAngular

## Hosting

Tableau de bord

Utilisation



Surveillez les requêtes Web envoyées à votre site avec Cloud Logging

Premiers pas

## Domaines demoangular-4c77e

Ajouter un domaine personnalisé

Domaine	État
<a href="#">demoangular-4c77e.web.app</a>	Par défaut
<a href="#">demoangular-4c77e.firebaseio.com</a>	Par défaut

## Historique des versions de demoangular-4c77e

État	Heure	Déploiement	Fichiers
Actuellement	23 oct. 2020 00:21	osall751@gmail.com 462a7d	10

Canaux de prévisualisation

Bêta

## Exemples du Cours d'Angular

[Se Connecter](#)

@Output- Incrémenteur le compteur

Status: VALID

osall@univ-thies.sn et passer123

## Vous souhaitez

[Ajouter un souhait](#)[Se Connecter](#)

Firebase CLI x | 🔥 demoAngular - Console Firebase x A PremiereAppAngular x +

demoangular-4c77e.web.app/accueil/osall@univ-thies.sn

Exemples du Cours d'Angular Accueil A propos Déconnecter @Output- Incrémenter le compteur

# Bienvenue osall@univ-thies.sn

## Créer un Rendez-Vous

0

Description

Lieu

Fri Oct 23 2020 00:24:22 GMT+0000 (heure moyenne de Greenwich)

Ajouter un nouveau RV

## Liste des Rendez-Vous

Tous Droits Réservés (C) 2020 avec @Input

En cas de mise à jour du code où d'une nouvelle version de l'application

1. Reconstruire la version à déployer: ***ng build --prod***
2. Se loguer si nécessaire à la console firebase: ***firebase login***
3. Optionnelle: Connecter le projet local et celui de firebase: ***firebase init***
4. Déployer l'application dans firebase: ***firebase deploy***

# Travaux Pratiques n°10

<https://developers.google.com/codelabs/building-a-web-app-with-angular-and-firebase#0>



Firebase



# Authentification et autorisation

# Contenu Cours Angular

- Composants
- Data Binding
- Routage et Navigation
- Directives
- Services et Injection de dépendances
- Pipes
- Gestion des Sessions avec localStorage et sessionStorage



# Contenu Cours Angular

- Interaction entre composants @input et @output
- Observables et Services HTTP
- Angular Material
- Formulaires
- Persistance de données et Interaction Cloud avec FireBase et FireStore
- Déploiement Application web Angular sur Firebase
- **Authentification et autorisation**

