

Formation Angular 11

TP N° 1 : Développement d'une première application avec Angular

SOMMAIRE

1.	Pré-requis :	3
2.	Objectifs	3
3.	Préparer l'environnement de développement	3
a.	Installation de Node.JS	3
b.	Installation de Angular CLI	5
c.	Installation de Microsoft Visual Studio Code	7
4.	Initialisation du projet avec CLI	8
5.	L'arborescence de l'application hello	11
a.	La page index.html	11
b.	Le fichier main.ts	12
c.	Le « root component » :	12
d.	Le fichier package.json	14
e.	Modifier le port par défaut	16
f.	Compiler le projet (le Build)	17
g.	Déployer votre application Angular sur Tomcat	18

1. Pré-requis :

- Node.JS
- Angular CLI
- Microsoft Visual Studio Code (VSD).

2. Objectifs

- ✓ Préparer les prérequis pour développer une application avec Angular.
- ✓ Développer une première application (hello) avec Angular.
- ✓ Comprendre l'arborescence d'un projet Angular.
- ✓ Mettre à jour les dépendances d'une application Angular (fichier **package.json**).
- ✓ Comment lancer le serveur.
- ✓ Comment modifier le port (le port par défaut fourni par le serveur web de Node.JS est 4200).
- ✓ Comment lancer les tests unitaires avec **Karma**.
- ✓ Comment lancer l'analyseur de code avec **TSLint**.
- ✓ Comment lancer les tests end-2-end avec **Protractor**.
- ✓ Comment compiler le projet et générer le produit final.
- ✓ Comment déployer le produit final sur le serveur Apache Tomcat.

3. Préparer l'environnement de développement

a. Installation de Node.JS

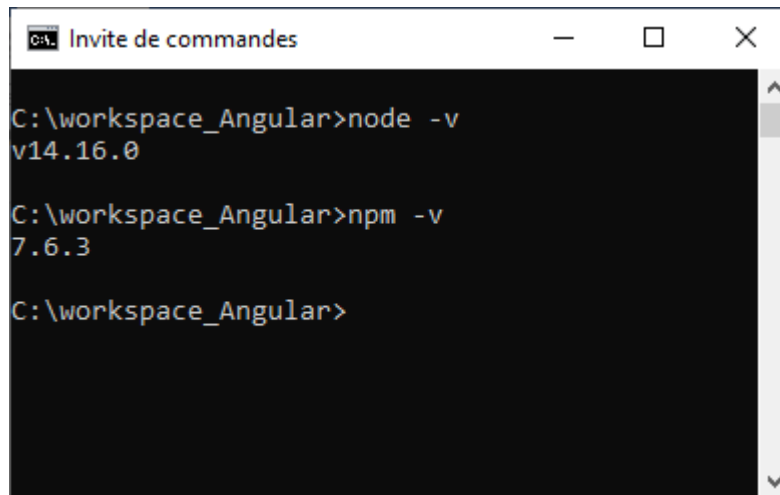
- Télécharger NodeJS via son site principal : <https://nodejs.org/fr/download>.

The screenshot shows the Node.js download page for French users. The URL in the browser is nodejs.org/fr/download/. The page features a navigation bar with links: ACCUEIL, À PROPOS, TÉLÉCHARGEMENTS, DOCS, S'IMPLIQUER, SECURITÉ, ACTUALITÉS, and CERTIFICATION. The main heading is "Téléchargements". Below it, it states "Dernière version LTS: 14.16.0 (includes npm 6.14.11)". A message encourages downloading the source code to start development today. The download options are categorized into "LTS" (Recommended for most users) and "Dernière" (Latest features). Under "LTS", there are three options: "Installateur Windows" (node-v14.16.0-x64.msi), "Installateur macOS" (node-v14.16.0.pkg), and "Code Source" (node-v14.16.0.tar.gz). The "Installateur Windows" option is highlighted with an orange box. Below these options, a table provides details for the Windows installer:

Installateur Windows (.msi)	32-bit	64-bit
Binaire Windows (.zip)	32-bit	64-bit
Installateur macOS (.pkg)	64-bit	
Binaire macOS (tar.gz)	64-bit	

Télécharger la version qui correspond à votre OS. Ici, c'est la version 14.16.0 incluant npm 6.14.11 sur Windows 64-bit.

- Vérifier si Node.JS et NPM ont été bien installées. Pour ceci, ouvrir une commande DOS et exécuter les commandes suivantes :

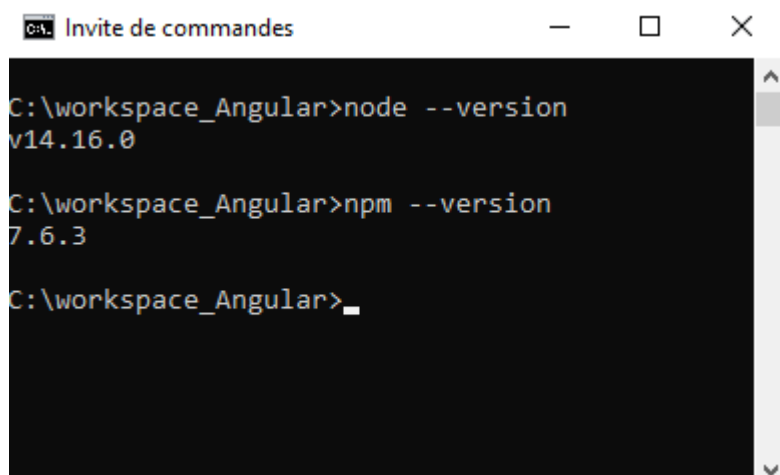


```
C:\workspace_Angular>node -v
v14.16.0

C:\workspace_Angular>npm -v
7.6.3

C:\workspace_Angular>
```

Vous pouvez également lancer les commandes suivantes :

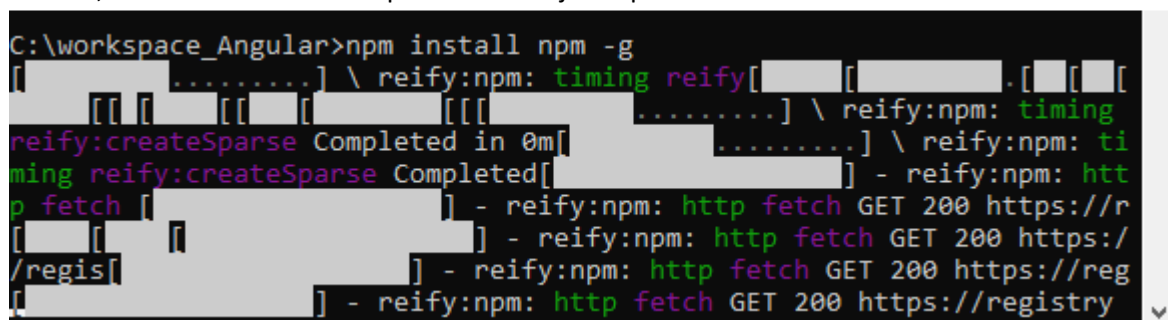


```
C:\workspace_Angular>node --version
v14.16.0

C:\workspace_Angular>npm --version
7.6.3

C:\workspace_Angular>_
```

- Ensuite, la commande suivante pour mettre à jour npm :

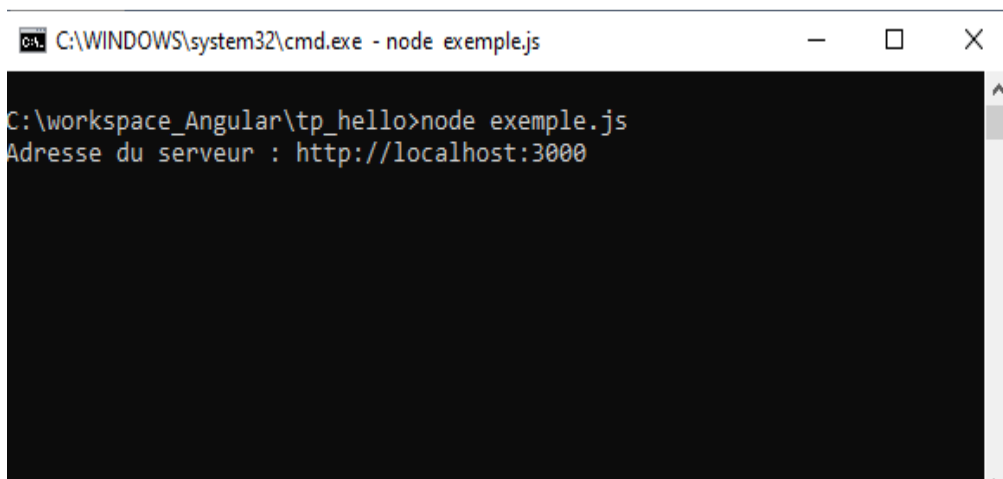


```
C:\workspace_Angular>npm install npm -g
[.....] \ reify:npm: timing reify[.....]
[[[.....] \ reify:npm: timing
reify:createSparse Completed in 0m[.....] \ reify:npm: ti
ming reify:createSparse Completed[.....] - reify:npm: htt
p fetch [.....] - reify:npm: http fetch GET 200 https://r
[.....] - reify:npm: http fetch GET 200 https:/
/regis[.....] - reify:npm: http fetch GET 200 https://reg
[.....] - reify:npm: http fetch GET 200 https://registry
```

- Pour vérifier si Node.JS fonctionne, suivre les étapes suivantes :
 - Créer un fichier vide exemple.js
 - Ouvrir le fichier avec un éditeur de code (exemple bloc-notes)
 - Copier le code suivant :

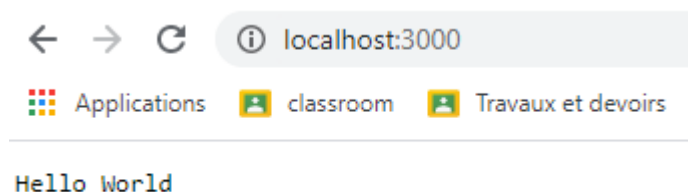
```
const { createServer } = require('http');  
//Creation du serveur  
const server = createServer((request, response) => {  
  response.writeHead(200, {'Content-Type': 'text/plain'});  
  response.end('Hello World\n');  
});  
  
server.listen(3000, () => console.log(`Adresse du serveur : http://localhost:3000`));
```

- Lancer la commande suivante :



```
C:\WINDOWS\system32\cmd.exe - node exemple.js  
  
C:\workspace_Angular\tp_hello>node exemple.js  
Adresse du serveur : http://localhost:3000
```

- Au niveau du navigateur, taper le lien <http://localhost:3000/> et vérifier que le message Hello World est bien affiché :



b. Installation de Angular CLI

- Si une version précédente était installée sur votre poste, vous pouvez la désinstaller avec la commande suivante :

```
C:\Users\toshiba satellite>npm uninstall -g @angular/cli
removed 241 packages, and audited 1 package in 20s
found 0 vulnerabilities
C:\Users\toshiba satellite>
```

- Installer CLI version 11.2.5 en exécutant la commande suivante :

```
C:\WINDOWS\system32\cmd.exe
C:\Users\toshiba satellite>npm install -g @angular/cli@11.2.5
```

- Vous pouvez également taper la commande suivante pour installer la dernière version de CLI :

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

- Vérifier si CLI est bien installée en exécutant la commande suivante :

```
C:\WINDOWS\system32\cmd.exe

C:\Users\toshiba satellite>ng --version

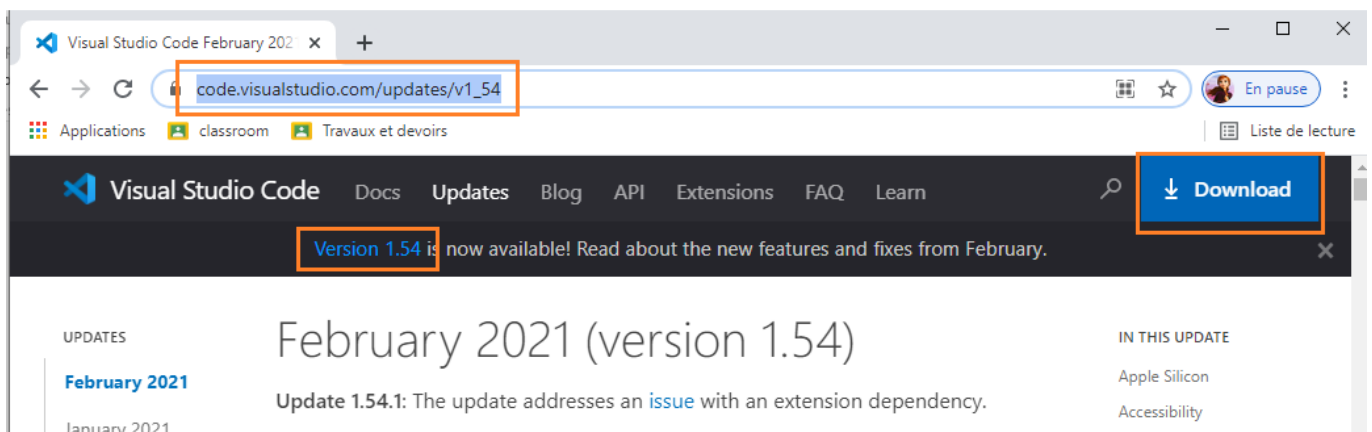
Angular CLI
Angular CLI: 11.2.5
Node: 14.16.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                                  Version
-----                                  -
@angular-devkit/architect               0.1102.5 (cli-only)
@angular-devkit/core                    11.2.5 (cli-only)
@angular-devkit/schematics              11.2.5 (cli-only)
@schematics/angular                    11.2.5 (cli-only)
@schematics/update                      0.1102.5 (cli-only)
```

c. Installation de Microsoft Visual Studio Code

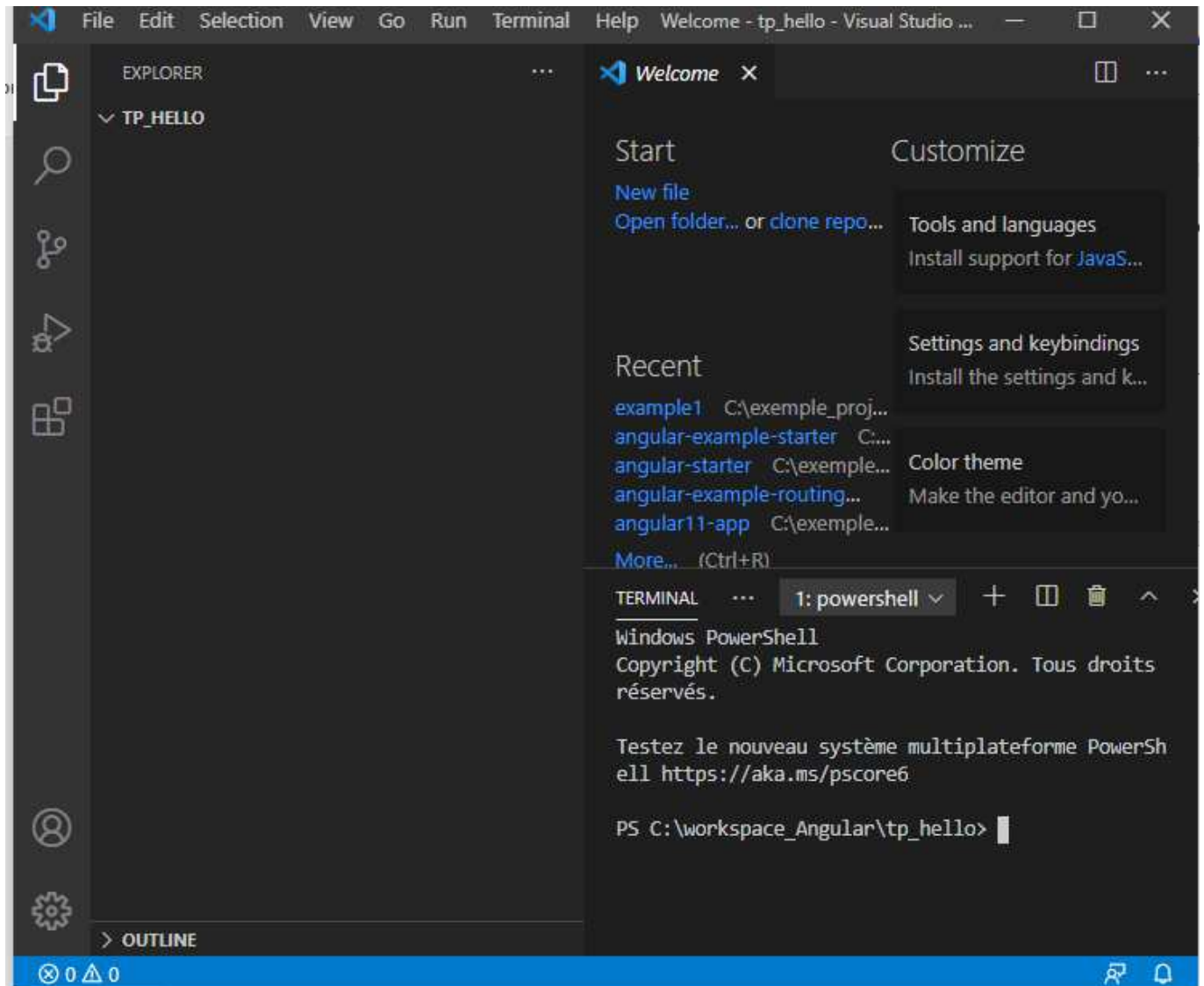
- Télécharger VSC de son site principal https://code.visualstudio.com/updates/v1_54



- L'installation est simple comme celle de Node.js. Vous choisissez votre système d'exploitation, vous téléchargez le programme et vous l'exécutez.
- Créer un dossier (exemple : tp_hello) et lancer la commande :

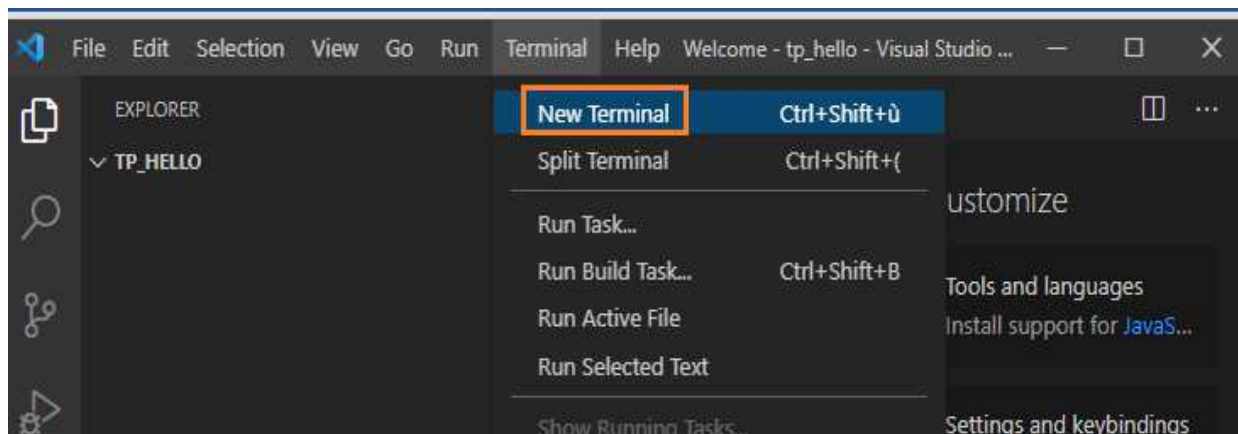
```
C:\>cd workspace_Angular  
C:\workspace_Angular>code tp_hello  
C:\workspace_Angular>
```

VSC sera démarré comme suit :

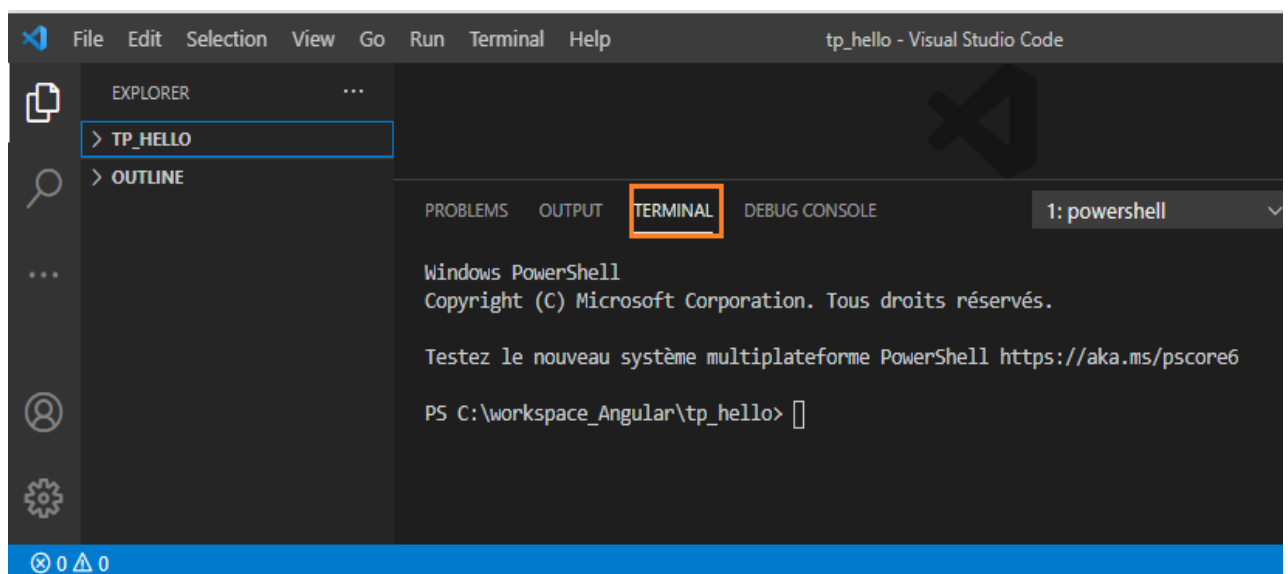


4. Initialisation du projet avec CLI

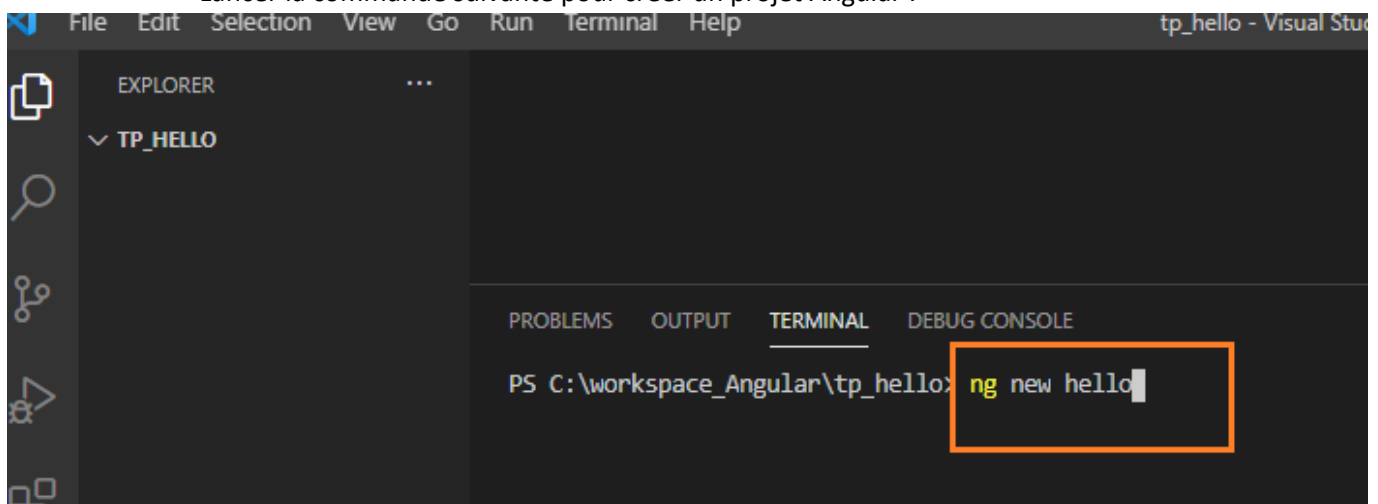
- Lancer VSC (voir ci-dessus).
- Ouvrir un terminal :



La fenêtre suivante est affichée :



- Lancer la commande suivante pour créer un projet Angular :

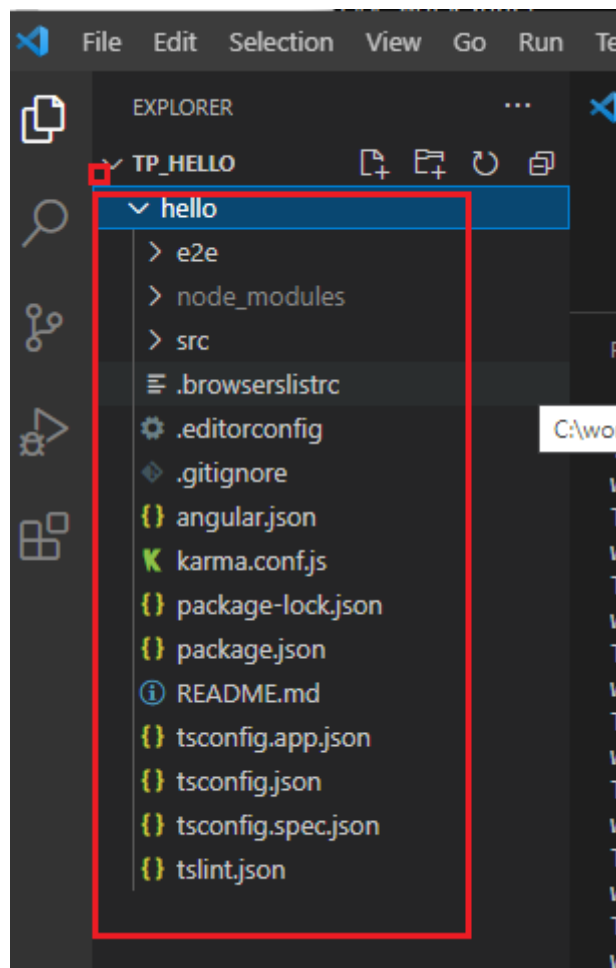


Et suivre les étapes suivantes :

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\workspace_Angular\tp_hello> ng new hello
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use?
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass  [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less  [ http://lesscss.org ]
Stylus [ https://stylus-lang.com ]
```

CLI générera automatiquement le squelette de votre application hello.

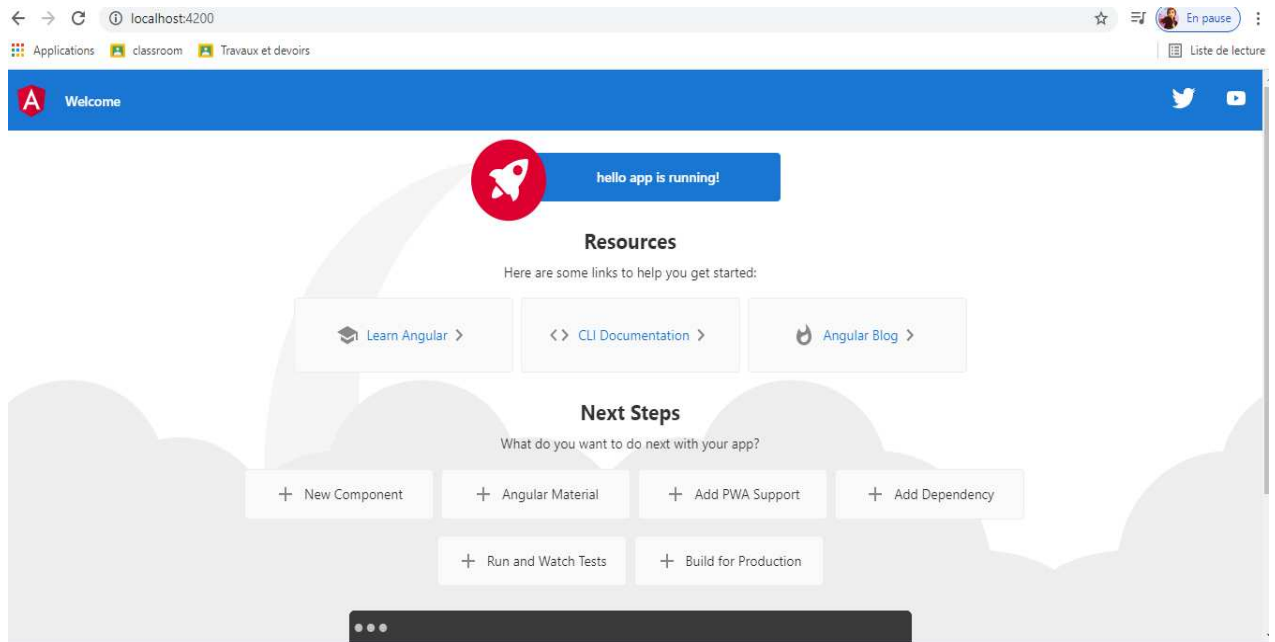


- Lancer les commandes suivantes au niveau du terminal :

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\workspace_Angular\tp_hello> cd .\hello\
PS C:\workspace_Angular\tp_hello\hello>
PS C:\workspace_Angular\tp_hello\hello>
PS C:\workspace_Angular\tp_hello\hello> ng serve
```

- Lancer le navigateur et lancer le lien <http://localhost:4200> :

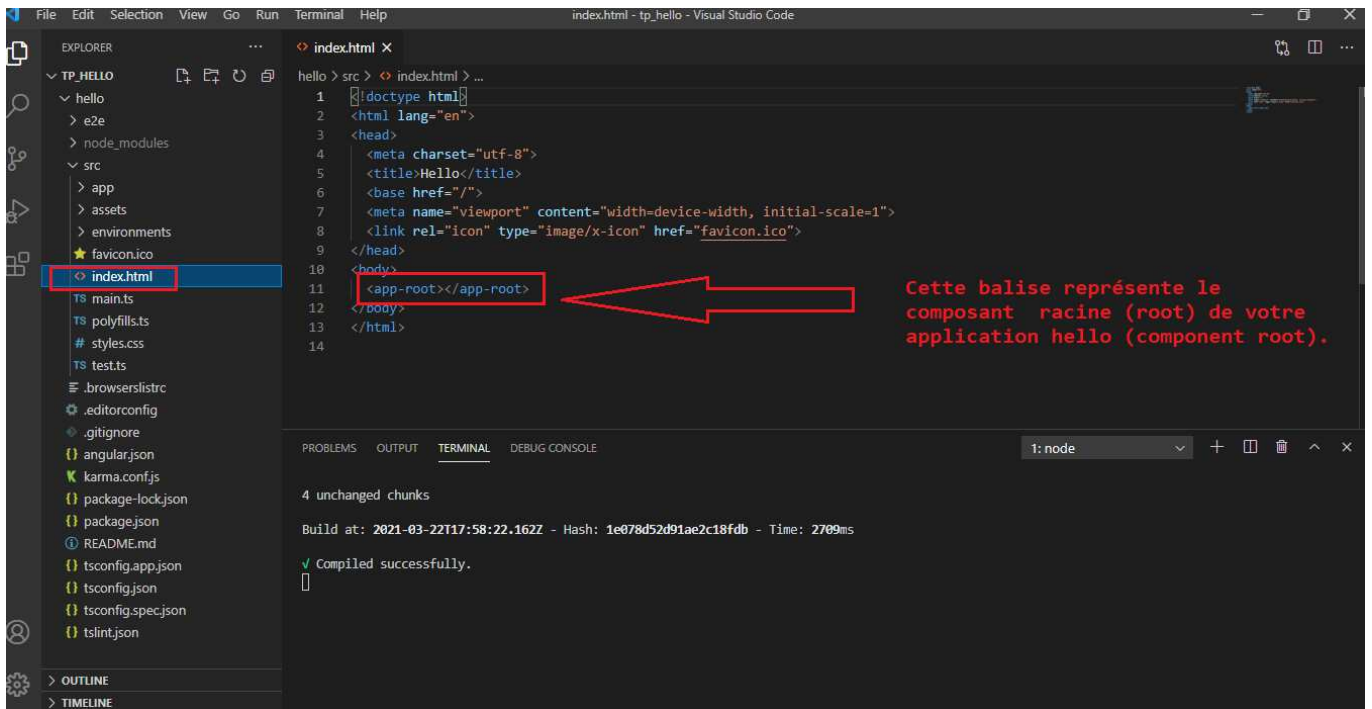


Vous venez de créer votre première application Angular !

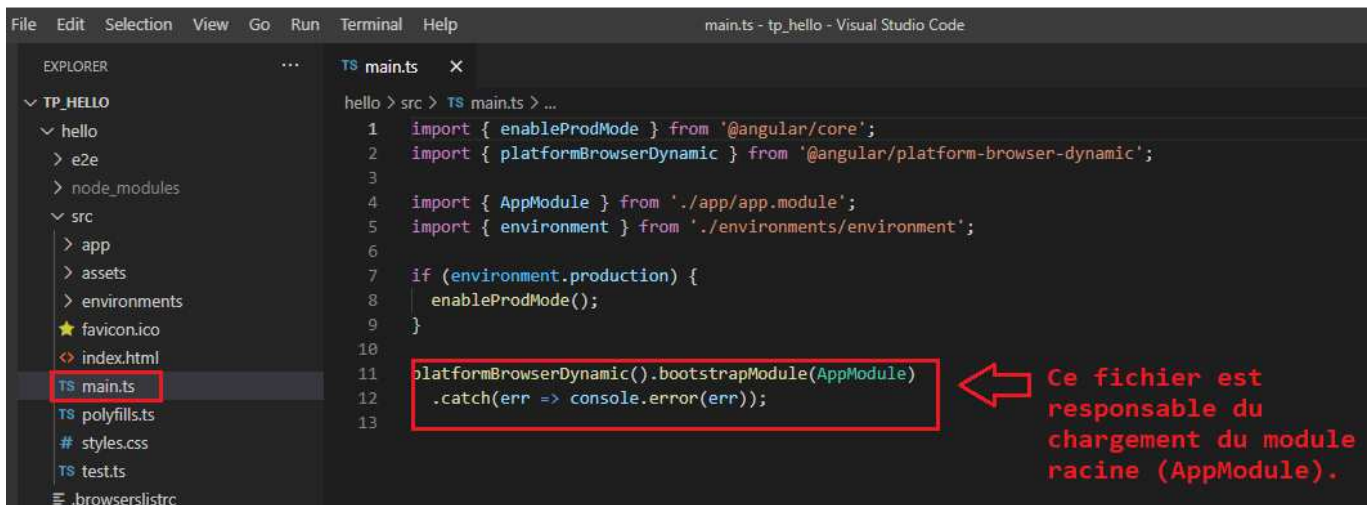
5. [L'arborescence de l'application hello](#)

a. La page index.html

Angular permet de créer les applications de type SPA (Single Page Application). Comme vous pouvez constater, il y a une seule page (index.html) qui est créée :



b. Le fichier main.ts



c. Le « root component » : app.module.ts :

TP_HELLO

hello

e2e

node_modules

src

app

TS app-routing.module.ts

app.component.css

<> app.component.html

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts

> assets

> environments

★ favicon.ico

<> index.html

TS main.ts

TS polyfills.ts

styles.css

TS test.ts

.browserslistrc

hello > src > app > TS app.module.ts > ...

```

1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
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

```

Un seul composant est déclaré par ce module qui AppComponent.

C'est la classe principale représentant le module racine (root). La classe est généralement appelée AppModule et est annotée par le Décorateur (Decorator) @NgModule

app.component.ts :

TP_HELLO

hello

e2e

node_modules

src

app

TS app-routing.module.ts

app.component.css

<> app.component.html

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts

> assets

> environments

★ favicon.ico

<> index.html

TS main.ts

TS polyfills.ts

styles.css

TS test.ts

hello > src > app > TS app.component.ts > ...

```

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 = 'hello';
10 }
11

```

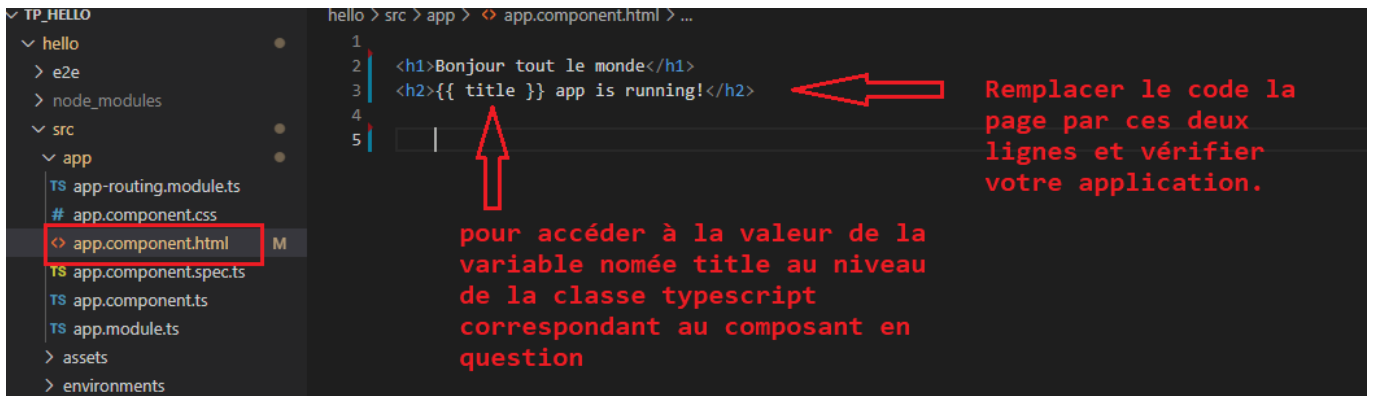
C'est un décorateur (Decorator) précisant :
1°) selector : le nom de la balise qui sera utilisée dans les pages (ici : <app-root> </app-root>)
2°) la page html qui sera exécutée par le composant.
3°) La feuille de style qui sera appliquée à la page.

title c'est une variable de la classe qui sera accessible au niveau de la page html (à travers le data binding).

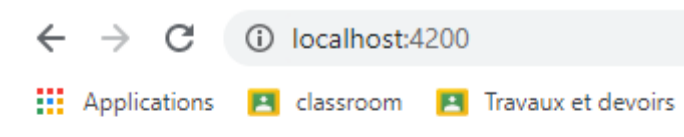
app.component.html :

TP n°1 : Première application avec Angular

13



Le résultat est le suivant :



Bonjour tout le monde

hello app is running!

d. Le fichier package.json

```
{
  "name": "hello",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~11.2.6",
    "@angular/common": "~11.2.6",
    "@angular/compiler": "~11.2.6",
    "@angular/core": "~11.2.6",
```

```

    "@angular/forms": "~11.2.6",
    "@angular/platform-browser": "~11.2.6",
    "@angular/platform-browser-dynamic": "~11.2.6",
    "@angular/router": "~11.2.6",
    "rxjs": "~6.6.0",
    "tslib": "^2.0.0",
    "zone.js": "~0.11.3"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.1102.5",
    "@angular/cli": "~11.2.5",
    "@angular/compiler-cli": "~11.2.6",
    "@types/jasmine": "~3.6.0",
    "@types/node": "^12.11.1",
    "codelyzer": "^6.0.0",
    "jasmine-core": "~3.6.0",
    "jasmine-spec-reporter": "~5.0.0",
    "karma": "~6.1.0",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage": "~2.0.3",
    "karma-jasmine": "~4.0.0",
    "karma-jasmine-html-reporter": "^1.5.0",
    "protractor": "~7.0.0",
    "ts-node": "~8.3.0",
    "tslint": "~6.1.0",
    "typescript": "~4.1.5"
  }
}

```

Explications :

- **ng serve** permet de démarrer le serveur web. Le port par défaut est 4200. Node.JS fourni un serveur web embarqué grâce à la librairie http.
- **ng build** permet de compiler le projet et générer le livrable final (en JavaScript et CSS), c'est le produit final à déployer sur le serveur de production.
- **ng test** permet de lancer les tests unitaires en utilisant **Karma**.
- **ng lint** permet de lancer l'analyse de code en utilisant **TSLint**.
- **ng e2e** permet de lancer les tests end-to-end en utilisant **Protractor**.
- L'attribut **dependencies** précise les dépendances utilisées pour développer l'application hello :
 - **version** : Doit correspondre à la version exactement
 - **~version** : Approximativement équivalente à la version
 - **^version** : Compatible avec la version
- Pour vérifier si votre application est à jour par rapport aux versions de vos dépendances, lancer la commande suivante :

```
PS C:\workspace_Angular\tp_hello\hello> npm outdated
```

Package	Current	Wanted	Latest	Location	Depended by
@angular-devkit/build-angular	0.1102.5	0.1102.6	0.1102.6	node_modules/@angular-devkit/build-angular	hello
@angular/animations	11.2.6	11.2.7	11.2.7	node_modules/@angular/animations	hello
@angular/cli	11.2.5	11.2.6	11.2.6	node_modules/@angular/cli	hello
@angular/common	11.2.6	11.2.7	11.2.7	node_modules/@angular/common	hello
@angular/compiler	11.2.6	11.2.7	11.2.7	node_modules/@angular/compiler	hello
@angular/compiler-cli	11.2.6	11.2.7	11.2.7	node_modules/@angular/compiler-cli	hello
@angular/core	11.2.6	11.2.7	11.2.7	node_modules/@angular/core	hello
@angular/forms	11.2.6	11.2.7	11.2.7	node_modules/@angular/forms	hello
@angular/platform-browser	11.2.6	11.2.7	11.2.7	node_modules/@angular/platform-browser	hello
@angular/platform-browser-dynamic	11.2.6	11.2.7	11.2.7	node_modules/@angular/platform-browser-dynamic	hello
@angular/router	11.2.6	11.2.7	11.2.7	node_modules/@angular/router	hello
@types/jasmine	3.6.7	3.6.9	3.6.9	node_modules/@types/jasmine	hello
@types/node	12.20.6	12.20.7	14.14.37	node_modules/@types/node	hello
jasmine-core	3.6.0	3.6.0	3.7.1	node_modules/jasmine-core	hello
jasmine-spec-reporter	5.0.2	5.0.2	6.0.0	node_modules/jasmine-spec-reporter	hello
karma	6.1.2	6.1.2	6.3.1	node_modules/karma	hello
ts-node	8.3.0	8.3.0	9.1.1	node_modules/ts-node	hello
typescript	4.1.5	4.1.5	4.2.3	node_modules/typescript	hello

Remarque :

Si vous voulez mettre à jour les versions de vos librairies, vous pouvez remplacer votre version par la version «**Wanted**» mentionnée au niveau du tableau ci-dessus en supprimant les symboles ~ et ^ et lancer ensuite la commande **npm install** (ou bien **npm install --force** pour forcer l'installation des mises à jour).

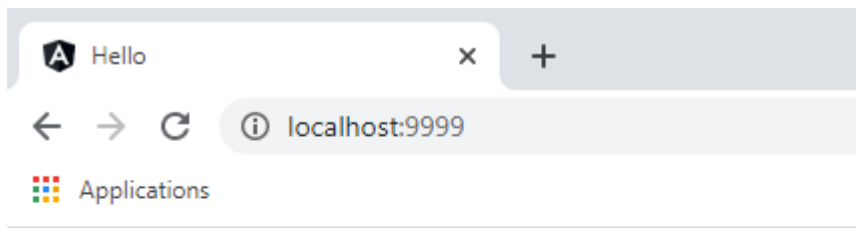
e. Modifier le port par défaut

Pour lancer le serveur dans un autre port, lancer la commande suivante :

```
PS C:\workspace_Angular\tp_hello\hello> ng serve --port 9999
✓ Browser application bundle generation complete.
```

Initial Chunk Files	Names	Size
vendor.js	vendor	2.69 MB

Lancer ensuite l'URL :



Bonjour tout le monde

hello app is running!

f. Compiler le projet (le Build)

Lancer la commande suivante :

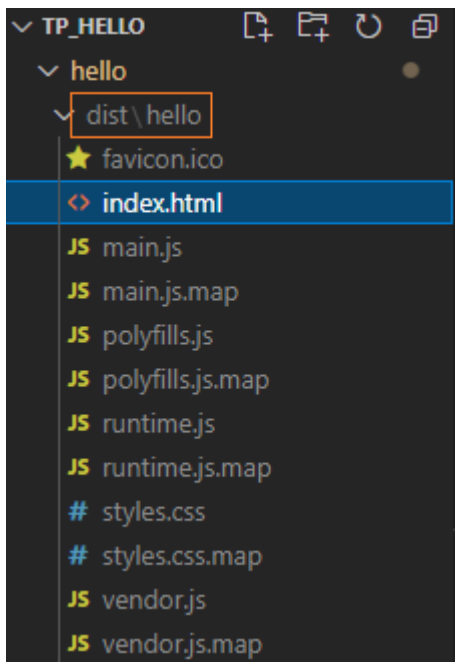
```
PS C:\workspace_Angular\tp_hello\hello> ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names      | Size
vendor.js           | vendor     | 2.35 MB
polyfills.js        | polyfills  | 128.81 kB
main.js             | main       | 8.61 kB
runtime.js          | runtime    | 6.15 kB
styles.css          | styles     | 119 bytes

| Initial Total | 2.49 MB

Build at: 2021-03-28T11:48:19.234Z - Hash: 0949886d413eee76e4d0 - Time: 8877ms
PS C:\workspace_Angular\tp_hello\hello>
```

Vérifier que votre livrable final (le dossier hello) est généré au niveau du dossier hello/dist :



g. Déployer votre application Angular sur Tomcat

- Commencer par lancer le build en mode de production en précisant le **href** avec la commande suivante :

```
PS C:\workspace_Angular\tp_hello\hello> ng build --base-href /hello/ --prod
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names | Size
main.3b251a59acae4bac8041.js | main | 182.01 kB
polyfills.94daefd414b8355106ab.js | polyfills | 35.98 kB
runtime.7b63b9fd40098a2e8207.js | runtime | 1.45 kB
styles.3ff695c00d717f2d2a11.css | styles | 0 bytes

| Initial Total | 219.44 kB

Build at: 2021-03-28T12:43:56.985Z - Hash: ae4d27388c4a72ec9aed - Time: 20456ms
```

Vérifier que votre livrable est bien généré au niveau du dossier dist/hello :

PC > TI31320200A (C:) > workspace_Angular > tp_hello > hello > dist > hello

Nom	Modifié le	Type	Taille
3rdpartylicenses.txt	28/03/2021 13:43	Document texte	13 Ko
favicon.ico	22/03/2021 18:40	Icône	1 Ko
index.html	28/03/2021 13:43	Microsoft Edge H...	1 Ko
main.3b251a59acae4bac8041.js	28/03/2021 13:43	Fichier de JavaScript	183 Ko
polyfills.94daefd414b8355106ab.js	28/03/2021 13:43	Fichier de JavaScript	36 Ko
runtime.7b63b9fd40098a2e8207.js	28/03/2021 13:43	Fichier de JavaScript	2 Ko
styles.3ff695c00d717f2d2a11.css	28/03/2021 13:43	Document de feui...	0 Ko

- Copier le dossier hello dans Tomat au niveau du dossier :

Ce PC > TI31320200A (C:) > serveurs > apache-tomcat-8.5.5 > webapps > hello

Nom	Modifié le	Type	Taille
3rdpartylicenses.txt	28/03/2021 13:38	Document texte	13 Ko
favicon.ico	22/03/2021 18:40	Icône	1 Ko
index.html	28/03/2021 13:38	Microsoft Edge H...	1 Ko
main.3b251a59acae4bac8041.js	28/03/2021 13:38	Fichier de JavaScript	183 Ko
polyfills.94daefd414b8355106ab.js	28/03/2021 13:38	Fichier de JavaScript	36 Ko
runtime.7b63b9fd40098a2e8207.js	28/03/2021 13:38	Fichier de JavaScript	2 Ko
styles.3ff695c00d717f2d2a11.css	28/03/2021 13:38	Document de feui...	0 Ko

- Ajouter la balise <Valve> suivante au niveau du fichier server.xml :

```
<Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />
```

Voir extrait du fichier conf/server.xml :

```
<Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
  <Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />
  <!-- SingleSignOn valve, share authentication between web applications
       Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->
```

- Créer ensuite le fichier **rewrite.config** dans le dossier conf/Catalina/localhost :

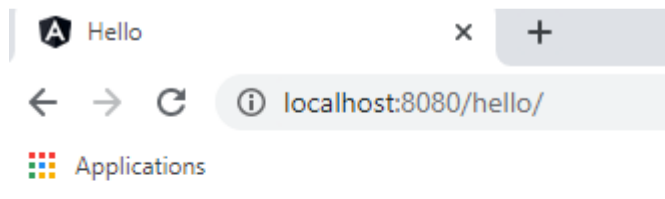
le PC > TI31320200A (C:) > serveurs > apache-tomcat-8.5.5 > **conf > Catalina > localhost**

Nom	Modifié le	Type	Taille
rewrite.config	28/03/2021 13:18	Fichier CONFIG	

- Copier les lignes suivantes au niveau du fichier rewrite.config :

```
RewriteCond %{REQUEST_PATH} !-f
RewriteRule ^/hello/(.*) /hello/index.html
```

- Démarrer Tomcat et lancer le lien suivant :



Bonjour tout le monde

hello app is running!

Vous venez de déployer votre première application Angular sur Apache Tomcat 8.5 !.

Maintenant, vous pouvez passer au TP n°2.

Fin du TP 1.