

COMANDI UTILI

CMD + D -> next occurrence of selection
CMD + Maiusc + L-> select all occurrence
CMD + Alt + N -> Run code (dopo installazione runcode in vscode)
CMD + P -> nel terminale. Ricopia l'ultimo comando inserito
CMD + C -> nel terminale. Ferma i controlli watch
CMD + Q -> chiusura forzata di qualsiasi app Mac

TYPESCRIPT

Compila ES6

tsc -> compila tutti i ts file in questa directory
tsc -w -> attiva l'auto compilazione a ogni modifica

TS CONFIG

tsc - -init -> crea il TSCONFIG.JSON
tsc 'our dir' -> togliere il commento e inserire il percorso dove creare i file js

ANGULAR

VS Code Extensions

- Angular Essentials
- Angular Schematics

npm i -> scarica i node modules in una cartella dove li abbiamo eliminati

ng new nome_cartella -> routing (no) -> scss -> crea nuovo progetto Angular

ng serve -o / ng s -o -> avvia compilazione continua

crtl + c -> chiudi processo di compilazione!!! (Prima di chiudere vs code altrimenti il processo resta attivo finì allo svuotarsi della ram)

npm i bootstrap

da vs code. Vai poi in style.scss principale @import '~bootstrap/scss/bootstrap'

Direttive Built In

- ngClass: classe ad elemento
- ngStyle: stile ad elemento
- ngIf: condizione per renderizzare o meno un elemento
- ngFor: stampare in loop un certo numero di elementi

Cartelle utili da creare in app

- components
- models
- service
- pages

COMPONENTI ANGULAR

Nuovo Componente

- tasto destro sulla cartella components
- Angular: generate a component
- Inserire nome componente
- Seleziona default component
- Seleziona addMoreOption
- Spunta skipTests
- Confirm

Inserire Componente

Per inserire un componente scrivere `<app-nome_componente></app-nome_componente>` nell'html dove lo vogliamo inserire.

FORM ANGULAR

Per importare i moduli dei form aprire il file `app.modules.ts` e importare `FormsModule` e dichiararlo in `imports`

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
```

Interfacce e Services - ANGULAR

Nuova Interfaccia

- Creare la cartella modules
- Cliccare con il destro
- Angular: generate another schematics
- Seleziona interface
- Inserire nome interface
- Premi ok senza spunte
- Confirm

Per il resto si comporta esattamente come le interfacce di TypeScript.

Nuovo Service

- Creare la cartella service
- Cliccare con il destro
- Angular: generate a service
- Inserire nome service
- Seleziona addMoreOption
- Spunta skipTests
- Confirm

I service fungono da “pacchetti” di metodi che possono essere importati.

Per importare i service o le interfacce si utilizza import dentro il ts del componente scelto.

```
import { Component, OnInit } from '@angular/core';
import { NomeInterfaccia } from 'src/app/models/nomeInterfaccia';
import { NomeService } from 'src/app/service/nome.service';
```

RouterModule ANGULAR - Creazione

Per creare delle rotte si deve prima importare il modulo built-in corrispondente dentro il file `app.modules.ts`

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

Successivamente si deve creare l'array di rotte che vogliamo far eseguire

```
const routes: Route[] = [
  {
    path: '',
    component: NomeComponente
  },
  {
    path: 'nomepath',
    component: NomeComponente
  },
  {
    path: 'nomepath',
    component: NomeComponente
  }
]
```

- `routes`: nome dell'array (si usa routes per convenzione)
- `Route[]`: tipo built-in dell'array dei nostri percorsi
- `path`: nome che identifica il pacchetto di componenti che deve essere mostrato quando viene richiamato. Il path principale avrà path vuoto!
- `component`: nomi dei componenti che devono essere richiamati

Aggiungere anche il router dentro gli `imports`, mettendo `.forRoot()` con dentro il nome del nostro array di rotte

```
imports: [
  BrowserModule,
  RouterModule.forRoot(routes)
]
```

Infine inserire `<router-outlet></router-outlet>` nell'html per aggiungere il nostro gruppo di rotte.

RouterModule ANGULAR - Agganciare Elementi

Per agganciare una rotta ad un elemento [a](#) si deve scrivere al suo interno al posto di href

```
[routerLink]="['/']" ( la pagina principale avrà come rotta / )  
[routerLink]="['/nomepath']"
```

Nel momento in cui noi cliccheremo sull'elemento a cui è collegato, se fa parte di un gruppo di routes, i suoi componenti andranno a rimpiazzare il contenuto precedente, che invece verrà rimosso.

[] : direttive

{{ }} : bindign unidirezionale