# **Coding Guidelines for Central Color Library front-end**

## **Naming conventions for Folders and Filenames:-**

* + Folder and file names should be named for the feature area they represent.
  + Feature type should be used as a suffix to the feature name.

e.g. **/src/components/login/login.component.ts**

* + Dashes should be used to separate words.

e.g. **/src/components/navbar/nav-bar.component.ts**

* + Dot-separated “spec” suffix to indicate test files.

e.g. **/src/components/login/login.component.spec.ts**

## **Naming and Coding guidelines for TypeScript code:-**

* + Pascal case should be used for classes (Components, Services, Directives and Pipe definitions).

**e.g. class Login { }**

* + All names should start with a letter.

**e.g. class Login { }**

* + Lower camelcase should be used for variables, function names, class instances, directive selectors, property bindings, event bindings, template variables and template element attributes .

e.g. **let userList = [ ... ];**

**function calculateTax(...) {}**

**<my-cmp \*ngIf="cond">**

**<my-cmp [myProp]="exp">**

**<my-cmp (myEvent)="action()">**

**<my-cmp [(myProp)]="prop">**

**<my-cmp #myCmp>**

**<template ngFor #myItem [ngForOf]=items #myIndex="index">**

* + Upper camelcase should be used for interface, prefixed with **“I”**.

e.g. **interface IUser {}**

* + Lower kebab-case should be used for Angular 2 component selectors.

e.g. **@Component({**

**selector: 'md-button',  
 })  
 class MdButton {  
 }**

* + The module name should be suffixed with the word **“Module”.**

e.g. **@NgModule({ ... })**

**export class AppModule { }**

for **“app.module.ts”**

* + Custom prefix should be used for the selector of directives.

e.g, **the prefix toh from Tour of Heroes**

* + Upper camelcase should be used for enums.

e.g. **enum UserRole { }**

* + Upper snake case should be used for constants.

**e.g. CONSTANT\_ONE = “ONE”;**

* + All the names should be self-explanatory.
  + Function and inline comments should be used wherever needed.
  + Statement should be ended with a semicolon.

**let values = ["Volvo", "Saab", "Fiat"];**

**let person = {**

**firstName: "John",**

**lastName: "Doe",**

**age: 50,**

**eyeColor: "blue"  
 }**;

* + A short description header should be there at the top of every code file as File Header.

e.g.

**/\*\*  
 \* filename  
 \*   
 \* Description   
 \*/**

* + Properties should be up top followed by methods.
  + Private members should be after public members and alphabetized.
  + @Input() and @Output() class decorators should be used instead of the inputs and outputs properties of the @Directive and @Component metadata
  + A code file should not have more than 300-400 lines of code.
  + A function should do only "one job."
  + A function should not be more than 25-30 lines large.
  + There should be one and only one single blank line between each method inside the class.
  + All member variables should be declared at the top of a class, with static variables at the very top.
  + Templates and styles should be extracted to their own files.
  + One empty line should be there between third party imports and application imports.
  + Files should be imported in the following order :-
    1. Angular library.
    2. Third party library.
    3. Application imports.

# **UI Design Guidelines**

## **HTML :-**

* + Lower camelcase should be used for Element Names.

e.g. **<section>   
 <p>This is a paragraph.</p>  
 </section>**

* + All the HTML elements should be closed, even the Empty HTML Elements.

e.g. **<meta charset="utf-8" />**

* + Lower camelcase should be used for attributes.

e.g. **<div class="menu"></div>**

* + **Attribute values should be quoted.**

**e.g. <table class="striped">**

* + Long Code Lines should be avoided.
  + Unnecessary blank lines should not be there and code should be properly Indented.
  + Inline css should be avoided.

## **CSS :-**

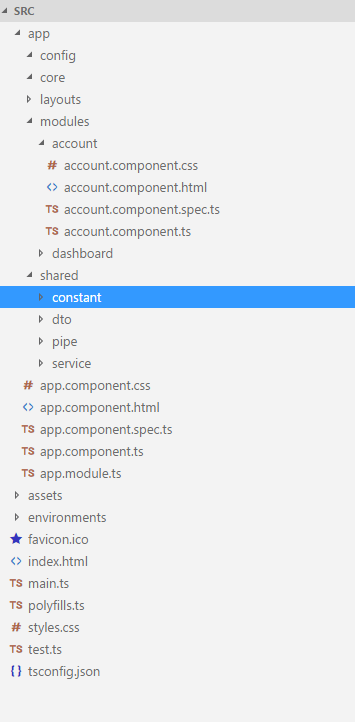
* + Stylesheet should be organised in a Top-down Structure e.g.
    1. **Generic classes (body, a, p, h1, etc.)**
    2. **#header**
    3. **#nav-menu**
    4. **#main-content**
  + All strings in class name should be delimited with a hyphen (-).
  + CSS properties should be grouped by type
  + CSS should be written across multiple lines.

e.g. **.icon {  
 display: inline-block;  
 width: 16px;  
 height: 16px;  
 background-image: url(/img/sprite.svg);  
 }**

* + Related selectors should be on the same line and unrelated ones on new lines.
  + A space should be there before opening brace ({).
  + Properties and values should be on the same line.
  + A space should be there after property–value delimiting colon (:).
  + Each declaration should be on its own new line.
  + The opening brace ({) should be on the same line as the last selector.
  + First declaration should be on a new line after opening brace ({).
  + Closing brace (}) should be on its own new line.
  + Each declaration should be indented by two (2) spaces.
  + A trailing semi-colon (;) should be there on the last declaration.
  + Each major section should be titled.

# **Angular 2 Project Folder Structure**

* **/src/ :-** contains all the code required for the application.
* **/src/app/ :-** root folder and contains a root module.
* **/src/app/modules/ :-** contains modules of the application. Each module within contains a components folder that will have the file related to that component like .html, .css, .component.ts etc.
* **/src/app/shared/ :-** contains the components, directives, and pipes in a shared module when those items will be re-used and referenced by the components declared in other feature modules.
* **/src/app/core/** :- contains feature module named **CoreModule**, that have a singleton service whose instance will be shared throughout the application in the CoreModule (e.g. **ExceptionService** and **LoggerService**).
* **/src/app/config/ :-** contains the configuration related files like for paths etc.
* **/src/app/assets/ :-** contains external css, fonts, images etc.
* /src/app/modules/services/ :- will contain the service related to the respective module/component**.**
* **/src/app/modules/dtos/ :-**  will contain the DTO(Data Transfer Object), related to the respective module/component**.**
* **/src/app/modules/component/ :-**  will contain the components related to the respective module**.**



**Folder structure for Angular2 project for CCL**