Angular Error: 'Component 'X' is not included in a module…' when declared in a sub module

I'm trying to consolidate my dialogs into an Angular module, but I'm getting a linting error in the IDE:

14

> Component 'X' is not included in a module and will not be available inside a template. Consider adding it to an NgModule declaration.

Despite this error the application still loads and runs successfully.

Example Component Definition

2

```typescript
import { Component, Inject, OnInit, ViewEncapsulation } from '@angular/core';
import { MAT_DIALOG_DATA, MatDialogRef } from '@angular/material';

export interface AlertDialogData {
  titleText: string;
  dismissalText: string;
  contentComponent: string;
}

@Component({
  selector: 'app-alert-dialog',
  templateUrl: './alert-dialog.component.html',
  styleUrls: ['./alert-dialog.component.scss'],
  encapsulation: ViewEncapsulation.None
})
export class AlertDialogComponent implements OnInit {

  constructor(private dialogRef: MatDialogRef<AlertDialogComponent>,
@Inject(MAT_DIALOG_DATA) public data: any) { }

  ngOnInit() {
  }

  handleCloseClick(): void {
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

Sub module making Declaration/Export

```typescript
import { CUSTOM_ELEMENTS_SCHEMA, NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { ZipLocatorDialogComponent } from './zip-locator-dialog/zip-locator-
dialog.component';
import { AlertDialogComponent } from './alert-dialog/alert-dialog.component';
import { HelperDialogComponent } from './helper-dialog/helper-dialog.component';
import {
  MatAutocompleteModule, MatButtonModule, MatDialogModule, MatFormFieldModule,
MatInputModule,
  MatSelectModule
} from '@angular/material';
import { FlexLayoutModule } from '@angular/flex-layout';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

@NgModule({
  imports: [
    CommonModule,
    FlexLayoutModule,
    FormsModule,
    ReactiveFormsModule,
    MatDialogModule,
    MatInputModule,
    MatFormFieldModule,
    MatSelectModule,
    MatAutocompleteModule,
    MatButtonModule
  ],
  exports: [
    ZipLocatorDialogComponent,
    HelperDialogComponent,
    AlertDialogComponent
  ],
  declarations: [
    ZipLocatorDialogComponent,
    HelperDialogComponent,
    AlertDialogComponent
  ],
  entryComponents: [
    ZipLocatorDialogComponent,
    HelperDialogComponent
```

```
})
export class AppDialogsModule { }
```

App Module

```
// <editor-fold desc="Global Application Imports">
import { BrowserModule } from '@angular/platform-browser';
import { NgModule, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HTTP_INTERCEPTORS, HttpClientModule } from '@angular/common/http';
import { RouterModule } from '@angular/router';
import { RouteDefinitions } from './app.routes';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { FlexLayoutModule } from '@angular/flex-layout';
import { WebWrapperModule } from 'web-wrapper';
import { UiComponentsModule } from './ui-components.module';
import { AppComponent } from './app.component';


// OPERATORS
import './rxjs-operators';

// SERVICES
import { LoginManagerService } from './services/login-manager.service';
import { UtilsService } from './services/utils.service';
import { DataManagerService } from './services/data-manager.service';
import { ReferenceDataManagerService } from './services/reference-data-manager.service';
import { InfrastructureApiService } from './services/infrastructure-api.service';
// </editor-fold>

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    FormsModule,
    FlexLayoutModule,
    HttpClientModule,
    WebWrapperModule,
```

```
        LoginManagerService,
        DataManagerService,
        InfrastructureApiService,
        ReferenceDataManagerService
    ],
    schemas: [
        CUSTOM_ELEMENTS_SCHEMA
    ],
    bootstrap: [AppComponent]
})
export class AppModule { }
```

Versions

```
Angular CLI: 1.5.0
Node: 7.2.1
OS: win32 x64
Angular: 4.4.6
... animations, common, compiler, compiler-cli, core, forms
... http, language-service, platform-browser
... platform-browser-dynamic, router, tsc-wrapped

@angular/cdk: 2.0.0-beta.12
@angular/cli: 1.5.0
@angular/flex-layout: 2.0.0-beta.11-b01c2d7
@angular/material: 2.0.0-beta.12
@angular-devkit/build-optimizer: 0.0.32
@angular-devkit/core: 0.0.20
@angular-devkit/schematics: 0.0.35
@ngtools/json-schema: 1.1.0
@ngtools/webpack: 1.8.0
@schematics/angular: 0.1.2
typescript: 2.4.2
webpack: 3.8.1
```

angular   typescript   angular-cli   lint

edited Jan 23 at 19:10   asked Nov 30 '17 at 23:31

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook

23:44

It's not literally 'X', but it appears for every dialog component declared in the `AppDialogsModule`, i.e. `HelperDialogComponent`. The exports are only included, because I saw that was provided in a different SO answer however it makes no difference. The `AppDialogsModule` is already included as an import to the `AppModule` which is all that should be required correct? — rawkfist0215 Nov 30 '17 at 23:46

1   There are quite a few rules here on how this works. Consider checking out my video on the topic from ngConf here: youtube.com/watch?v=ntJ-P-Cvo7o&t=1s — DeborahK Dec 1 '17 at 7:09

Regarding your error, have you seen this: github.com/angular/vscode-ng-language-service/issues/10. It appears that this may be a known error with the VS Code language service. Are you using that tool? — DeborahK Dec 1 '17 at 7:12

I'm using the npm package `@angular/language-service": "^4.3.6` tool that ships with the CLI. Maybe it's the same issue the VS Code version has though. Thanks for sharing the link to the talk. While I think my modules were already configured correctly, (I removed exporting the components) you gave an excellent presentation! If anything I gained more understanding into the "why" instead of employing blind copy/paste tactics. — rawkfist0215 Dec 1 '17 at 18:11 ✏

4 Answers

I had this same issue and this is how it got resolved :

31

1) Goto Intellij / IDE settings and tick(set) the Recompile on changes to :

✔
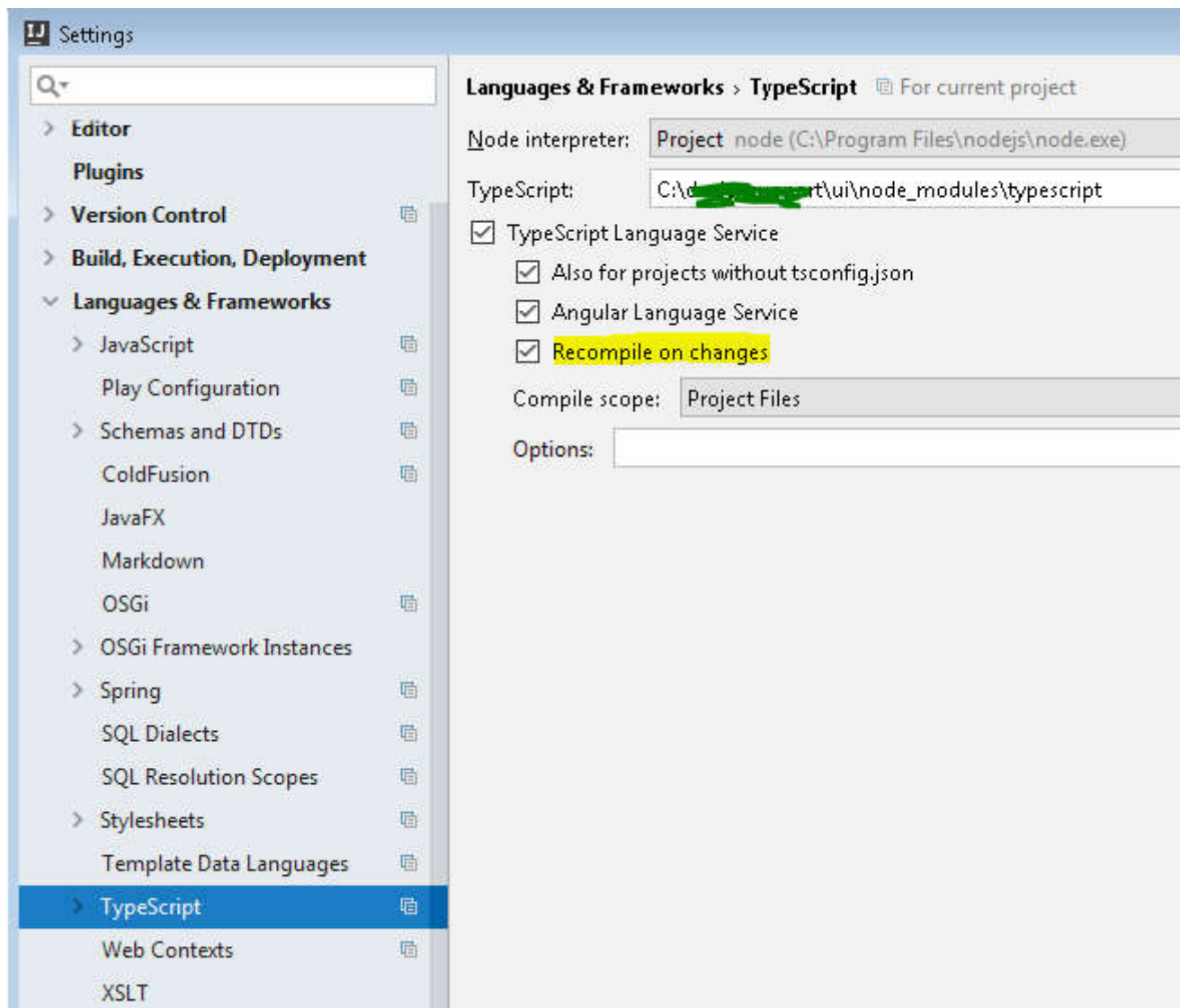
2) Goto *tsconfig.json* and set the *compileOnSave* to true :

Now go and remove the @Component that's causing the issue, and retype @Component.

This worked for me :) Good Luck.

answered Jun 4 '18 at 10:55

Praveesh P
632   9   24

It worked for me just by doing the first step – Ahmed AMMOURI Feb 27 at 14:45

1   god I am happy I found this answer! The webstorm site doesn't even have this there. – Mark Mar 3 at 21:16

Add `@angular/language-service` as a dev-dpendency

1   or

do `npm install @angular/language-service`

Can confirm this is fixed in v `5.2.9` . YMMV with earlier versions.

answered Apr 6 '18 at 16:16

AnthonyW
1,232   2   21   40

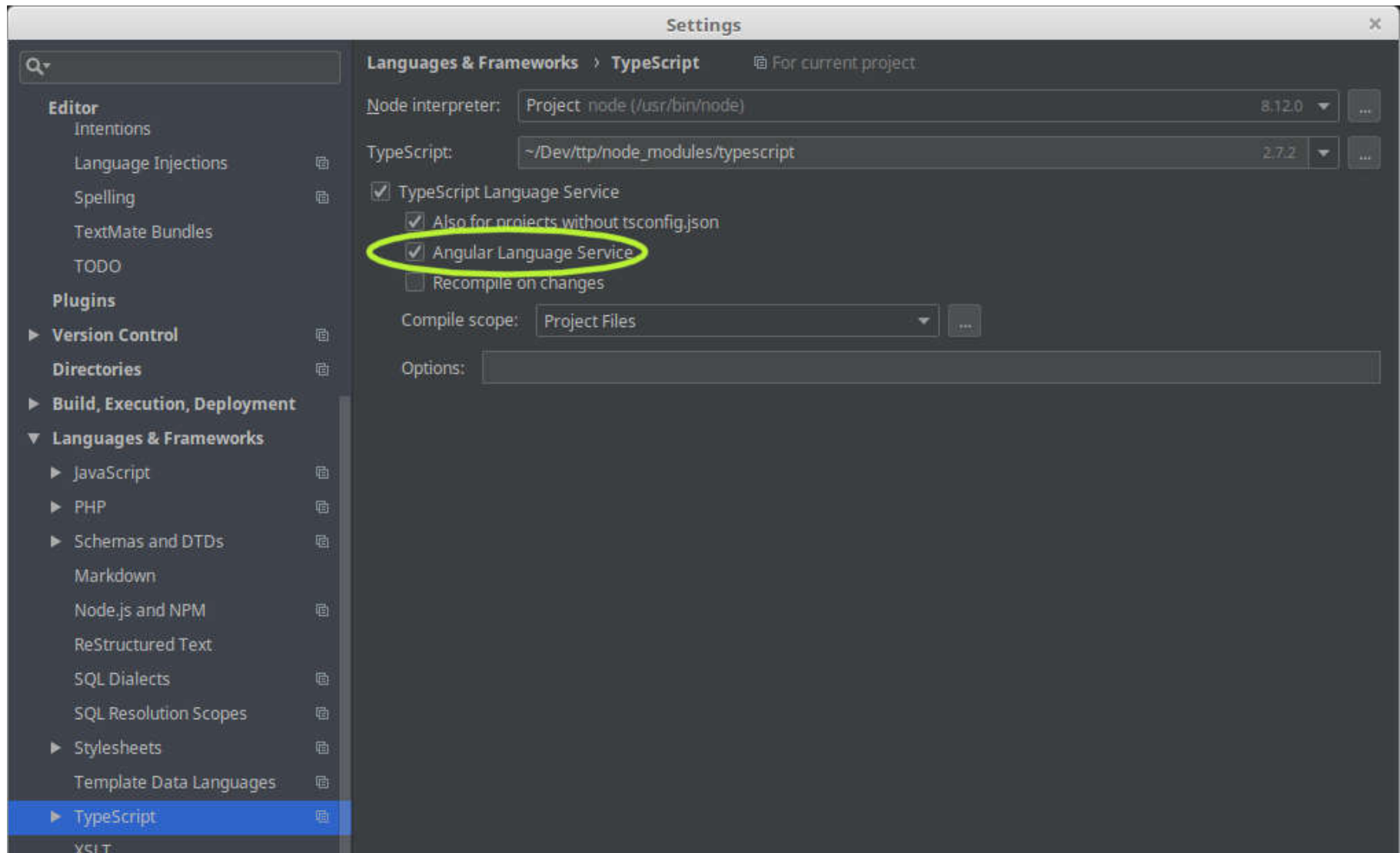**Join Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook

Angular Language Service checkbox in Settings | Languages & Frameworks | Typescript

1 https://intellij-support.jetbrains.com/hc/en-us/community/posts/360000014820-Component-is-not-includd-in-a-module-and-will-not-be-available-inside-a-template

answered Oct 10 '18 at 19:04

Stack Underflow
573   1   8   23

---

1   Or in Visual Studio Code just uninstall the Angular Language Service extension. — SushiGuy Mar 6 at 22:36

First: declare all your components in **declarations** section (`app.module.ts`).

1

If the problem persist, I remember that is a problem with the beta angular-cli versions.

> The issue you are running into is a variant of the baseUrl issue. The language service does not correctly respect the baseUrl option. For example, if you change the import of the shared module from app/shared/shared.module to ../shared/shared.module then the errors go away.

edited Feb 28 at 17:38   answered Nov 30 '17 at 23:46

Pipo   alvaropanizo
1,856   16   28   **107**   5

Doing so leads to the following error: `Uncaught Error: Type ZipLocatorDialogComponent is part of the declarations of 2 modules: AppDialogsModule and AppModule! Please consider moving ZipLocatorDialogComponent to a higher module that imports AppDialogsModule and AppModule. You can also create a new NgModule that exports and includes ZipLocatorDialogComponent then import that NgModule in AppDialogsModule and AppModule.` Which feels ironic, because their suggestion sums up what I'm trying to do in the first place. — rawkfist0215 Dec 1 '17 at 0:00

Try to remove from **imports** and only put this module in **declarations** — alvaropanizo Dec 1 '17 at 0:02

2   Sorry, but the error message still displays. It's only the linting tool that's confused, because Angular still compiles the app and runs without exceptions. — rawkfist0215   Dec 1 '17 at 0:24