# Function calls are not supported in decorators

I run this command in my GitLab CI pipeline because I like to know early if there are any `--prod` or `--aot` build issues.

```
ng build lms --aot --prod
```

I get the following error.

```
ERROR in Error during template compile of 'CrossCuttingModule'
  Function calls are not supported in decorators but 'ConfigurationModule' was
called.
: Unexpected value 'undefined' imported by the module 'CrossCuttingModule in
D:/development/gitlab/lms/workspace/apps/lms/src/app/modules/cross-cutting/cross-
cutting.module.ts'
Error during template compile of 'CrossCuttingModule'
  Function calls are not supported in decorators but 'ConfigurationModule' was
called.
```

- Angular 8 Workspace
- (1) Application Project
- (7) Library projects
- not publishing the libraries, referenced via the `tsconfig`:

```
"paths": {
      "@angularlicious/configuration": ["libs/configuration/src/index.ts"],
      "@angularlicious/error-handling": ["libs/error-handling/src/index.ts"],
      "@angularlicious/logging": ["libs/logging/src/index.ts"],
      "@angularlicious/rules-engine": ["libs/rules-engine/src/index.ts"],
      "@angularlicious/actions": ["libs/actions/src/index.ts"],
      "@angularlicious/http-service": ["libs/http-service/src/index.ts"],
      "@angularlicious/foundation": ["libs/foundation/src/index.ts"]
    }
```

## Workspace Details

```
ng version
```

```
                |___/


Angular CLI: 8.0.0
Node: 10.16.0
OS: win32 x64
Angular: 8.1.1
... animations, common, compiler, compiler-cli, core, forms
... language-service, platform-browser, platform-browser-dynamic
... router
------------------------------------------------------------
@angular-devkit/architect          0.800.6
@angular-devkit/build-angular      0.800.6
@angular-devkit/build-ng-packagr   0.800.6
@angular-devkit/build-optimizer    0.800.6
@angular-devkit/build-webpack      0.800.6
@angular-devkit/core               8.0.1
@angular-devkit/schematics         8.0.1
@angular/cli                       8.0.0
@ngtools/webpack                   8.0.6
@schematics/angular                8.0.0
@schematics/update                 0.800.0
ng-packagr                         5.5.0
rxjs                               6.4.0
typescript                         3.4.5
webpack                            4.30.0
```

## Source

I have a library project in my Angular 8 Workspace.

```
import { NgModule, ModuleWithProviders } from "@angular/core";
import { CommonModule } from "@angular/common";
import { ConfigurationContext } from "./configuration-context";

@NgModule({
  imports: [CommonModule]
})
export class ConfigurationModule {
  static forRoot(configContext: ConfigurationContext): ModuleWithProviders {
    console.log(`Preparing to handle configuration context.`);
    return {
      ngModule: ConfigurationModule,
      providers: [
        {
          provide: ConfigurationContext,
          useValue: configContext
        }
      ]
    };
```

```
  }
}
```

## Application Module

I have an application module that has the responsibility to initialize some of the cross-cutting concerns. It starts by using the `forRoot(..)` static method of the `ConfigurationModule`. This is what is causing the error.

```
  imports: [CommonModule, ErrorHandlingModule, LoggingModule,
ConfigurationModule.forRoot({ config: environment.appConfig })],
```

Here is the source for the `CrossCuttingModule` in the sample application.

```
import { NgModule, APP_INITIALIZER, ErrorHandler } from "@angular/core";
import { CommonModule } from "@angular/common";
import {
  LoggingService,
  LoggingModule,
  LogglyWriter
} from "@angularlicious/logging";
import {
  ConfigurationService,
  ConfigurationModule
} from "@angularlicious/configuration";
import { ConsoleWriter } from "@angularlicious/logging";
import {
  ErrorHandlingModule,
  ErrorHandlingService
} from "@angularlicious/error-handling";
import { environment } from "./../../../environments/environment";

/**
 * The factory function to initialize the logging service and writer for the
 * application.
 *
 * @param loggingService
 * @param consoleWriter
 */
export function initializeLogWriter(consoleWriter: ConsoleWriter) {
  console.log(`Initializing [Console Writer] from [AppModule]`);
  return () => {
    return consoleWriter;
  };
}
@NgModule({
  declarations: [],
  imports: [
    CommonModule,
```

```
    ErrorHandlingModule,
    LoggingModule,
    ConfigurationModule.forRoot({ config: environment.appConfig })
  ],
  providers: [
    ConfigurationService,
    LoggingService,
    ConsoleWriter,
    LogglyWriter,
    {
      provide: ErrorHandler,
      useClass: ErrorHandlingService
    },
    {
      provide: APP_INITIALIZER,
      useFactory: initializeLogWriter,
      deps: [LoggingService, ConsoleWriter, LogglyWriter],
      multi: true
    },
    {
      provide: ErrorHandler,
      useClass: ErrorHandlingService,
      deps: [ConfigurationService, LoggingService]
    }
  ]
})
export class CrossCuttingModule {}
```