# 403.https://stackoverflow.com/questions/71210272/error-ts1252-function-declarations-are-not-allowed-inside-blocks-in-strict-mode

**T:**error TS1252: Function declarations are not allowed inside blocks in strict mode when targeting 'ES3' or 'ES5'. Stacks.js

**Q:**I created typescript using typeORM to connect mySQL. seemed to work fine with the below config  
  
 { "compilerOptions": { "lib": [ "es5", "es6" ], "target": "es5", "module": "commonjs", "moduleResolution": "node", "outDir": "./build", "emitDecoratorMetadata": true, "experimentalDecorators": true, "esModuleInterop": true, "forceConsistentCasingInFileNames": true, "strict": true, "skipLibCheck": true, "strictPropertyInitialization": false }, "files": [ "./src/index copy.ts", "./src/index.ts", "./src/entity/Catalog.ts", "./src/entity/CatalogTimestamp.ts", "./src/entity/User.ts", "./src/entity/contracts.ts", "./src/entity/metadata.ts", "./src/entity/networks.ts", "./src/entity/parcels.ts", "./src/entity/tokens.ts" ], "exclude": [ "./build" ] }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
But, I am trying to add it to an existing typescript where I am using stacks.js to call functions on a smartcontract. This code worked fine (see [https://stackoverflow.com/questions/71146879/error-ts1378-top-level-await-expressions-are-only-allowed-when-the-module-o][1])Now when I run  
  
npm start  
  
I get the above error referencing this part of the code where run is redlined.  
  
 async function run() { const transaction = await makeContractCall(txOptions); const broadcastResponse = await broadcastTransaction(transaction, network); const txId = broadcastResponse.txid; console.log(broadcastResponse) } run().then(console.log).catch(console.error);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
This part of the code works fine when the target is set to es2020.Can you use both the es5 target and the ES2020? I a trying to read/write from a mySQL file to/from a Stacks smart contract NFT.

**C1:**Does doing const run = () => { instead fix the issue?

0 **Answer**