

tslib direct dependency migration

✎

What does this migration do?

If you have any libraries within your workspace, this migration will convert `tslib` peer dependencies to direct dependencies for the libraries. TypeScript uses the `tslib` package to provide common helper functions used in compiled TypeScript code. The `tslib` version is also updated to `2.0.0` to support TypeScript 3.9.

Before:

```
{
  "name": "my-lib",
  "version": "0.0.1",
  "peerDependencies": {
    "@angular/common": "^9.0.0",
    "@angular/core": "^9.0.0",
    "tslib": "^1.12.0"
  }
}
```

📄

After:

```
{
  "name": "my-lib",
  "version": "0.0.1",
  "peerDependencies": {
    "@angular/common": "^9.0.0",
    "@angular/core": "^9.0.0"
  },
  "dependencies": {
    "tslib": "^2.0.0"
  }
}
```

📄

Why is this migration necessary?

The `tslib` [🔗](#) is a runtime library for Typescript. The version of this library is bound to the version of the TypeScript compiler used to compile a library. Peer dependencies do not accurately represent this relationship between the runtime and the compiler. If `tslib` remained declared as a library peer dependency, it would be possible for some Angular workspaces to get into a state where the workspace could not satisfy `tslib` peer dependency requirements for multiple libraries, resulting in build-time or run-time errors.

As of TypeScript 3.9 (used by Angular v10), `tslib` version of 2.x is required to build new applications. However, older libraries built with previous version of TypeScript and already published to npm might need `tslib` 1.x. This migration makes it possible for code depending on incompatible versions of the `tslib` runtime library to remain interoperable.

Do I still need tslib as a dependency in my workspace package.json?

Yes. The `tslib` dependency declared in the `package.json` file of the workspace is used to build applications within this workspace, as well as run unit tests for workspace libraries, and is required.