

Angular Update Guide

Select the options that match your update

Angular versions

From v. To v.

Warning: Be sure to follow the guide below to migrate your application to the new version. You can't run ng update to update Angular applications more than one major version at a time.

Application complexity

Basic	Medium	Advanced
-------	--------	----------

Shows information for all Angular developers.

Other dependencies

- ☐ I use ngUpgrade to combine AngularJS & Angular
- ☐ I use Angular Material
- ☒ I use Windows

Show me how to update!

Guide to update your Angular application v7.2 -> v12.0 for basic applications

Before you update

- ☐ If you use the legacy `HttpModule` and the `Http` service, switch to `HttpClientModule` and the `HttpClient` service. `HttpClient` simplifies the default ergonomics (you don't need to map to JSON anymore) and now supports typed return values and interceptors. Read more on [angular.io](https://update.angular.io/?v=7.2-12.0).

- ☐ Once you and all of your dependencies have updated to RxJS 6, remove `rxjs-compat`.
- ☐ If you use the Angular Service worker, migrate any `versionedFiles` to the `files` array. The behavior is the same.

Update to the new version

Review these changes and perform the actions to update your application.

- ☐ Update to version 8 of the core framework and CLI by running

```
cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@8 update
@angular/cli@8 @angular/core@8"
```

in your terminal and review and commit the changes.
- ☐ Replace `/deep/` with `::ng-deep` in your styles, [read more about angular component styles and ::ng-deep](#). `/deep/` and `::ng-deep` both are deprecated but using `::ng-deep` is preferred until the shadow-piercing descendant combinator is [removed from browsers and tools](#) completely.
- ☐ Angular now uses TypeScript 3.4, [read more about errors that might arise from improved type checking](#).
- ☐ Make sure you are using [Node 10 or later](#).
- ☐ The CLI's build command now automatically creates a modern ES2015 build with minimal polyfills and a compatible ES5 build for older browsers, and loads the appropriate file based on the browser. You may opt-out of this change by setting your `target` back to `es5` in your `tsconfig.json`. Learn more on [angular.io](#).
- ☐ When using new versions of the CLI, you will be asked if you want to opt-in to share your CLI usage data. You can also add your own Google Analytics account. This lets us make better decisions about which CLI features to prioritize, and measure the impact of our improvements. Learn more on [angular.io](#).
- ☐ If you use `ViewChild` or `ContentChild`, we're updating the way we resolve these queries to give developers more control. You must now specify that change detection should run before results are set. Example: `@ContentChild('foo', {static: false}) foo !: ElementRef; .ng_update` will update your queries automatically, but it will err on the side of making your queries `static` for compatibility. Learn more on [angular.io](#).

- ☐ For lazy loaded modules via the router, make sure you are [using dynamic imports](#). Importing via string is removed in v9. `ng update` should take care of this automatically. Learn more on [angular.io](#).
- ☐ Make sure you are using [Node 10.13 or later](#).
- ☐ Run

```
cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@8 update
@angular/cli@8 @angular/core@8"
```

in your workspace directory to update to the latest 8.x version of `@angular/core` and `@angular/cli` and commit these changes.
- ☐ You can optionally pass the `--create-commits` (or `-C`) flag to [ng update](#) commands to create a git commit per individual migration.
- ☐ Run

```
cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@9 update
@angular/cli@9 @angular/core@9"
```

which should bring you to version 9 of Angular.
- ☐ Your project has now been updated to TypeScript 3.8, read more about new compiler checks and errors that might require you to fix issues in your code in the [TypeScript 3.7](#) or [TypeScript 3.8](#) announcements.
- ☐ If your project depends on other Angular libraries, we recommend that you consider updating to their latest version. In some cases this update might be required in order to resolve API incompatibilities. Consult `ng update` or `npm outdated` to learn about your outdated libraries.
- ☐ During the update to version 9, your project was transformed as necessary via code migrations in order to remove any incompatible or deprecated API calls from your code base. You can now review these changes, and consult the [Updating to version 9 guide](#) to learn more about the changes.
- ☐ Bound CSS styles and classes previously were applied with a "last change wins" strategy, but now follow a defined precedence. Learn more about [Styling Precedence](#).
- ☐ Remove any `es5BrowserSupport` flags in your `angular.json` and set your `target` to `es2015` in your `tsconfig.json`. Angular now uses your `browserslist` to determine if an ES5 build is needed. `ng update` will migrate you automatically.

- ☐ If you use `ngForm` element selector to create Angular Forms, you should instead use `ng-form`.
- ☐ With Angular 9 Ivy is now the default rendering engine, for any compatibility problems that might arise, read the [Ivy compatibility guide](#).
- ☐ Angular 9 introduced a global `$localize()` function that needs to be loaded if you depend on Angular's internationalization (i18n). Run `ng add @angular/localize` to add the necessary packages and code modifications. Consult the [\\$localize Global Import Migration guide](#) to learn more about the changes.
- ☐ In your application projects, you can remove `entryComponents` `NgModules` and any uses of `ANALYZE_FOR_ENTRY_COMPONENTS`. They are no longer required with the Ivy compiler and runtime. You may need to keep these if building a library that will be consumed by a View Engine application.
- ☐ If you use `TestBed.get`, you should instead use `TestBed.inject`. This new method has the same behavior, but is type safe.
- ☐ If you use [Angular's i18n support](#), you will need to begin using `@angular/localize`. Learn more about the [\\$localize Global Import Migration](#).
- ☐ Make sure you are using [Node 12 or later](#).
- ☐ Run `npx @angular/cli@10 update @angular/core@10 @angular/cli@10` which should bring you to version 10 of Angular.
- ☐ New projects use the filename `.browserslistrc` instead of `browserslist`. `ng update` will migrate you automatically.
- ☐ Angular now requires `tslint` v6, `tslib` v2, and [TypeScript 3.9](#). `ng update` will migrate you automatically.
- ☐ In version 10, classes that use Angular features and do not have an Angular decorator are no longer supported. [Read more](#). `ng update` will migrate you automatically.
- ☐ As of Angular 9, enforcement of `@Injectable` decorators for DI is stricter and incomplete provider definitions behave differently. [Read more](#). `ng update` will migrate you automatically.

- ☐ If you use Angular forms, inputs of type `number` no longer listen to [change events](#) (these events are not necessarily fired for each alteration of the value), instead listen for an [input event](#).
- ☐ For Angular forms validation, the `minLength` and `maxLength` validators now verify that the form control's value has a numeric length property, and only validate for length if that's the case.
- ☐ The [Angular Package Format](#) has been updated to remove `esm5` and `fesm5` formats. These are no longer distributed in our npm packages. If you don't use the CLI, you may need to downlevel Angular code to ES5 yourself.
- ☐ Warnings about unknown elements are now logged as errors. This won't break your app, but it may trip up tools that expect nothing to be logged via `console.error`.
- ☐ You may see `ExpressionChangedAfterItHasBeenChecked` errors that were not detected before when using the `async` pipe. The error could previously have gone undetected because two `WrappedValues` are considered "equal" in all cases for the purposes of the check, even if their respective unwrapped values are not. In version 10, `WrappedValue` has been removed.
- ☐ For more details about deprecations, automated migrations, and changes visit the [guide](#) [angular.io](#)
- ☐ For Angular Universal users, if you use `useAbsoluteUrl` to setup `platform-server`, you now need to also specify `baseUrl`.
- ☐ Run `ng update @angular/core@11 @angular/cli@11` which should bring you to version 11 of Angular.
- ☐ Angular now requires [TypeScript 4.0](#). `ng update` will migrate you automatically.
- ☐ Support for IE9, IE10, and IE mobile has been removed. This was announced in the [v10 update](#).
- ☐ You can now opt-in to use webpack 5 by using Yarn and adding `"resolutions": {"webpack": "^5.0.0"}` to your `package.json`.
- ☐ When generating new projects, you will be asked if you want to enable strict mode. This will configure TypeScript and the Angular compiler for stricter type checking, and apply smaller bundle budgets by default. You can use the `--strict=true` or `--strict=false` to skip the prompt.

- ☐ In the Angular Router's `routerLink` , `preserveQueryParams` has been removed, use `queryParamsHandling="preserve"` instead.
- ☐ The `uppercase` and `lowercase` pipes no longer let falsy values through. They now map both `null` and `undefined` to `null` and raise an exception on invalid input (`0` , `false` , `NaN`). This matches other Angular pipes.
- ☐ In your tests if you call `TestBed.overrideProvider` after `TestBed` initialization, provider overrides are no longer applied. This behavior is consistent with other override methods (such as `TestBed.overrideDirective` , etc) but they throw an error to indicate that. The check was previously missing in the `TestBed.overrideProvider` function. If you see this error, you should move `TestBed.overrideProvider` calls before `TestBed` initialization is completed.
- ☐ If you use the Router's `RouteReuseStrategy`, the argument order has changed. When calling `RouteReuseStrategy#shouldReuseRoute` previously when evaluating child routes, they would be called with the `future` and `current` arguments swapped. If your `RouteReuseStrategy` relies specifically on only the future or current snapshot state, you may need to update the `shouldReuseRoute` implementation's use of `future` and `current` `ActivateRouteSnapshots` .
- ☐ If you use Angular Forms with async validators defined at initialization time on class instances of `FormControl` , `FormGroup` or `FormArray` , the status change event was not previously emitted once async validator completed. This has been changed so that the status event is emitted into the `statusChanges` observable. If your code relies on the old behavior, you can filter/ignore this additional status change event.
- ☐ Run `ng update @angular/core@12 @angular/cli@12` which should bring you to version 12 of Angular.
- ☐ Angular now requires [TypeScript 4.2](#). `ng update` will update you automatically.
- ☐ IE11 support has been deprecated. Find details in the [RFC for IE11 removal](#).
- ☐ You can no longer use Angular with Node.js version 10 or older
- ☐ Change the import of `XhrFactory` from `@angular/common/http` to `@angular/common` .
- ☐ If you rely on legacy i18n message IDs use the `localize-migrate` tool to [move away from them](#).

- ☐ If you are using `emitDistinctChangesOnly` to configure `@ContentChildren` and `@ViewChildren` queries, you may need to update its value to `false` to align with its previous behavior. In v12 `emitDistinctChangesOnly` has default value `true`, and in future releases we will remove this configuration option to prevent triggering of unnecessary changes.
- ☐ You can run the optional migration for enabling production builds by default
`ng update @angular/cli@12 --migrate-only production-by-default`.

After you update

You don't need to do anything after moving between these versions.