Browser support

Contents ›

Angular supports most recent browsers. This includes the following specific versions:

Browser	Supported versions
Chrome	latest
Firefox	latest and extended support release (ESR)
Edge	2 most recent major versions
IE	11, 10*, 9* ("compatibility view" mode not supported) *deprecated in v10, see the deprecations guide.
IE Mobile*	11 *deprecated in v10, see the deprecations guide.
Safari	2 most recent major versions
iOS	2 most recent major versions
Android	X (10.0), Pie (9.0), Oreo (8.0), Nougat (7.0)

Angular's continuous integration process runs unit tests of the framework on all of these browsers for every pull request, using SauceLabs and Browserstack.

Polyfills

Angular is built on the latest standards of the web platform. Targeting such a wide range of browsers is challenging because they do not support all features of modern browsers. You compensate by loading polyfill scripts ("polyfills") for the browsers that you must support. The table below identifies most of the polyfills you might need.

> The suggested polyfills are the ones that run full Angular applications. You may need additional polyfills to support features not covered by this list. Note that polyfills cannot magically transform an old, slow browser into a modern, fast one.

In Angular CLI version 8 and higher, applications are built using *differential loading*, a strategy where the CLI builds two separate bundles as part of your deployed application.

- The first bundle contains modern ES2015 syntax, takes advantage of built-in support in modern browsers, ships less polyfills, and results in a smaller bundle size.
- The second bundle contains code in the old ES5 syntax, along with all necessary polyfills. This results in a larger bundle size, but supports older browsers.

This strategy allows you to continue to build your web application to support multiple browsers, but only load the necessary code that the browser needs. For more information about how this works, see Differential Loading in the Deployment guide.

Enabling polyfills with CLI projects

The Angular CLI provides support for polyfills. If you are not using the CLI to create your projects, see Polyfill instructions for non-CLI users.

When you create a project with the `ng new` command, a `src/polyfills.ts` configuration file is created as part of your project folder. This file incorporates the mandatory and many of the optional polyfills as JavaScript `import` statements.

- The npm packages for the *mandatory* polyfills (such as `zone.js`) are installed automatically for you when you create your project with `ng new`, and their corresponding `import` statements are already enabled in the `src/polyfills.ts` configuration file.
- If you need an *optional* polyfill, you must install its npm package, then uncomment or create the corresponding import statement in the `src/polyfills.ts` configuration file.

For example, if you need the optional web animations polyfill, you could install it with `npm`, using the following command (or the `yarn` equivalent):

```
# install the optional web animations polyfill
npm install --save web-animations-js
```

You can then add the import statement in the `src/polyfills.ts` file. For many polyfills, you can simply un-comment the corresponding `import` statement in the file, as in the following example.

src/polyfills.ts

```
/**
* Required to support Web Animations `@angular/platform-browser/animations`.
* Needed for: All but Chrome, Firefox and Opera. http://caniuse.com/#feat=web-animation
**/
import 'web-animations-js';  // Run `npm install --save web-animations-js`.
```

If the polyfill you want is not already in `polyfills.ts` file, add the `import` statement by hand.

Mandatory polyfills

These are the polyfills required to run an Angular application on each supported browser:

Browsers (Desktop & Mobile)	Polyfills Required
Chrome, Firefox, Edge, Safari, Android, IE 10+	ES2015
IE 9	ES2015 classList

Optional browser features to polyfill

Some features of Angular may require additional polyfills.

Feature	Polyfill	Browsers (Desktop & Mobile)
AnimationBuilder. (Standard animation support does not require polyfills.)	Web Animations	If AnimationBuilder is used, enables scrubbing support for IE/Edge and Safari. (Chrome and Firefox support this natively).
NgClass on SVG elements	classList	IE 10, IE 11
Http when sending and receiving binary data	Typed Array Blob FormData	IE 9
Router when using hash-based routing	ES7/array	IE 11

Suggested polyfills

The following polyfills are used to test the framework itself. They are a good starting point for an application.

Polyfill	License	Size*
ES7/array	MIT	0.1KB
ES2015	MIT	27.4KB
classList	Public domain	1KB
Intl	MIT / Unicode license	13.5KB
Web Animations	Apache	14.8KB
Typed Array	MIT	4KB
Blob	MIT	1.3KB
FormData	MIT	0.4KB

* Figures are for minified and gzipped code, computed with the closure compiler.

Polyfills for non-CLI users

If you are not using the CLI, add your polyfill scripts directly to the host web page (`index.html`).

For example:

src/index.html

```
<!-- pre-zone polyfills -->
<script src="node_modules/core-js/client/shim.min.js"></script>
<script src="node_modules/web-animations-js/web-animations.min.js"></script>
<script>
  /**
   * you can configure some zone flags which can disable zone interception for some
   * asynchronous activities to improve startup performance - use these options only
   * if you know what you are doing as it could result in hard to trace down bugs..
   */
  // __Zone_disable_requestAnimationFrame = true; // disable patch
requestAnimationFrame
  // __Zone_disable_on_property = true; // disable patch onProperty such as onclick
```

```
  // __zone_symbol__UNPATCHED_EVENTS = ['scroll', 'mousemove']; // disable patch
specified eventNames

  /*
   * in IE/Edge developer tools, the addEventListener will also be wrapped by zone.js
   * with the following flag, it will bypass `zone.js` patch for IE/Edge
   */
  // __Zone_enable_cross_context_check = true;
</script>
<!-- zone.js required by Angular -->
<script src="node_modules/zone.js/dist/zone.js"></script>

<!-- application polyfills -->
```