

# Update Angular

This guide contains information to update to Angular version 13.

## Update Angular CLI applications

For step-by-step instructions on how to update to the latest Angular release and leverage the Angular automated migration tools, use the interactive update guide at [update.angular.io](https://update.angular.io).

## Changes and deprecations in version 13

For information about the deprecation and removal practices of Angular, see [Angular Release Practices](#).

- **Removal of View Engine**

Requires all applications and libraries to build using Ivy. See the [Upcoming improvements to Angular library distribution blog](#).

- **Modernization of the Angular Package Format (APF)**

Removed older output formats, including View Engine specific metadata.

- **Removal of IE11 Support**

Removes all support for Microsoft Internet Explorer 11 (IE11). See [Issue #41840](#).

- **Testbed module teardown**

Adds the option in `initTestEnvironment` to completely remove test environments from an application. See the [Improving Angular tests by enabling Angular testing module teardown article](#).

- **\$localize tagged message strings**

Adds documentation for the Angular `$localize` API and tagged message strings.

- **Disk Cache**

Enables the persistent build cache by default for all applications. See [Issue #21545](#).

## Breaking changes in Angular version 13

{@a breaking-changes}

	Details
PR #43642	TypeScript versions older than 4.4.2 are no longer supported.
PR #43740	NodeJS versions older than v12.20.0 are no longer supported. The Angular packages now use the NodeJS package exports feature with subpath patterns and requires a NodeJS version above 14.15.0 or 16.10.0.
PR #31187	Previously, the default url serializer dropped everything after and including a question mark in query parameters. That is, for a navigation to <code>/path?q=hello?&amp;other=123</code> , the query parameters parsed to just <code>{q: 'hello'}</code> . This is incorrect, because the URI spec allows for question mark characters in query data. This change now correctly parses the query parameters for <code>/path?q=hello?&amp;other=123</code> as <code>{q: 'hello?', other: '123'}</code> .
PR #41730	The behavior of the <code>SpyLocation</code> used by the <code>RouterTestingModule</code> has changed to match the behavior of browsers. It no longer emits a <code>popstate</code> event when <code>Location.go</code> is called. In addition, <code>simulateHashChange</code> now triggers <i>both</i> a <code>hashchange</code> event and a <code>popstate</code> event. Tests that use <code>location.go</code> and expect the changes to be picked up by the <code>Router</code> should migrate to <code>simulateHashChange</code> . Each test is different in what it attempts to assert, so there is no single change that works for all tests. Each test that uses the <code>SpyLocation</code> to simulate changes in the browser URL should be evaluated on a case-by-case basis.

	Details
PR #42952	<p>A new type called <code>FormControlStatus</code> has been introduced, which is a union of all possible status strings for form controls. <code>AbstractControl.status</code> has been narrowed from <code>string</code> to <code>FormControlStatus</code>, and <code>statusChanges</code> has been narrowed from <code>Observable&lt;any&gt;</code> to <code>Observable&lt;FormControlStatus&gt;</code>. Most applications should consume the new types seamlessly. Any breakage caused by this change is likely due to one of the following two problems:</p> <p>Previously <code>,null</code> and <code>undefined</code> inputs for <code>routerLink</code> were equivalent to empty string and there was no way to disable the navigation of the link. In addition, the <code>href</code> is changed from a property <code>HostBinding()</code> to an attribute binding <code>(HostBinding('attr.href'))</code>. The effect of this change is that <code>DebugElement.properties['href']</code> now returns the <code>href</code> value returned by the native element which is the full URL rather than the internal value of the <code>RouterLink href</code> property.</p>
PR #43087	

	Details
PR #43496	The router no longer replaces the browser URL when a new navigation cancels an ongoing navigation. The replacement of the browser URL often caused URL flicker and was only in place to support some AngularJS hybrid applications. Hybrid applications which rely on the presence of <code>navigationId</code> on each initial navigation handled by the Angular router should instead subscribe to <code>NavigationCancel</code> events and manually perform the <code>location.replaceState</code> to add <code>navigationId</code> to the Router state. In addition, tests that assert <code>urlChanges</code> on the <code>SpyLocation</code> should be adjusted to account for the lack of the <code>replaceState</code> trigger.
PR #43507	The <code>WrappedValue</code> class is no longer imported from <code>@angular/core</code> . This change may result in compile errors or failures at runtime, if outdated libraries are used that rely on <code>WrappedValue</code> . Dependency on <code>WrappedValue</code> should be removed since no replacement is available.
PR #43591	It is no longer possible to use <code>Route.loadChildren</code> with a string value. The following supporting classes were removed from <code>@angular/core</code> :
PR #43668	The return type of <code>SwUpdate#activateUpdate</code> and <code>SwUpdate#checkForUpdate</code> changed to <code>Promise&lt;boolean&gt;</code> . Although unlikely, this change may cause TypeScript type-checking to fail in some cases. If necessary, update your types to account for the new return type.

	Details
PR #43791	When storing and retrieving a <b>DetachedRouteHandle</b> , the Router traverses the <b>Route</b> children in order to correctly allow storing a parent route when there are several possible child <b>Route</b> configs that can be stored. This allows a <b>RouteReuseStrategy</b> to store a parent <b>Route</b> and a child, while preserving the ability to change the child route while restoring the parent. Some implementations of <b>RouteReuseStrategy</b> will need to be updated to correctly store and retrieve the <b>DetachedRouteHandle</b> of descendants as well as the stored parent <b>ActivatedRouteSnapshot</b> . Previously, the Router would only store the parent, making it impossible to change descendant paths when a parent stored was retrieved. See <b>Issue #20114</b>
Issue #22159	Scripts that load via dynamic <b>import()</b> are now treated as ES modules (meaning they must be strict mode-compatible).

## New deprecations

{@a deprecations}

Removed	Replacement	Details
<b>getModuleFactory</b>	<b>getNgModuleById</b>	
Factory-based signature of <b>ApplicationRef.bootstrap</b>	Type-based signature of <b>ApplicationRef.bootstrap</b>	Use the Type-based signature in place of the Factory-based signature.
<b>PlatformRef.bootstrapModuleFactory</b>	<b>PlatformRef.bootstrapModule</b>	
<b>ModuleWithComponentFactories</b>	<b>Module</b>	
<b>Compiler</b>	none	
<b>CompilerFactory</b>	none	

Removed	Replacement	Details
<code>NgModuleFactory</code>	Non-factory based framework APIs	Use the non-factory based framework APIs, such as <code>PlatformRef.bootstrapModule</code> and <code>createNgModuleRef</code> .
Factory-based signature of <code>ViewContainerRef.createComponent</code>	Type-based signature of <code>ViewContainerRef.createComponent</code>	Use the Type-based signature in place of the Factory-based signature.
<code>aotSummaries</code> parameter of the <code>TestBed.initTestEnvironment</code> method	none	
<code>aotSummaries</code> parameter of the <code>TestModuleMetadata</code> type	none	
<code>renderModuleFactory</code>	<code>renderModule</code>	
<code>SwUpdate#activated</code>	<code>SwUpdate#activateUpdate()</code>	Use the return value of <code>SwUpdate#activateUpdate()</code> .
<code>SwUpdate#available</code>	<code>SwUpdate#versionUpdates</code>	
<code>bind-input="value"</code>	<code>[input]="value"</code>	
<code>bind-animate-trigger="value"</code>	<code>[animateTrigger]="value"</code>	
<code>on-click="onClick()"</code>	<code>(click)="onClick()"</code>	
<code>bindon-ngModel="value"</code>	<code>[(ngModel)]="value"</code>	
<code>ref-templateRef</code>	<code>#templateRef</code>	

@reviewed 2021-11-01