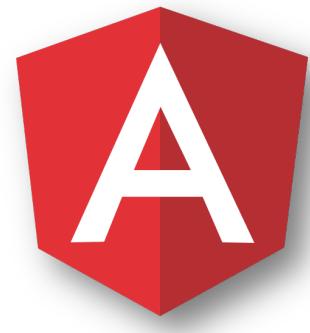


State of Angular

Uncovering Exciting New Features and Enhancements

Thirumoorthy Ponnusamy
Trustrace





About me

Thirumooorthy Ponnusamy

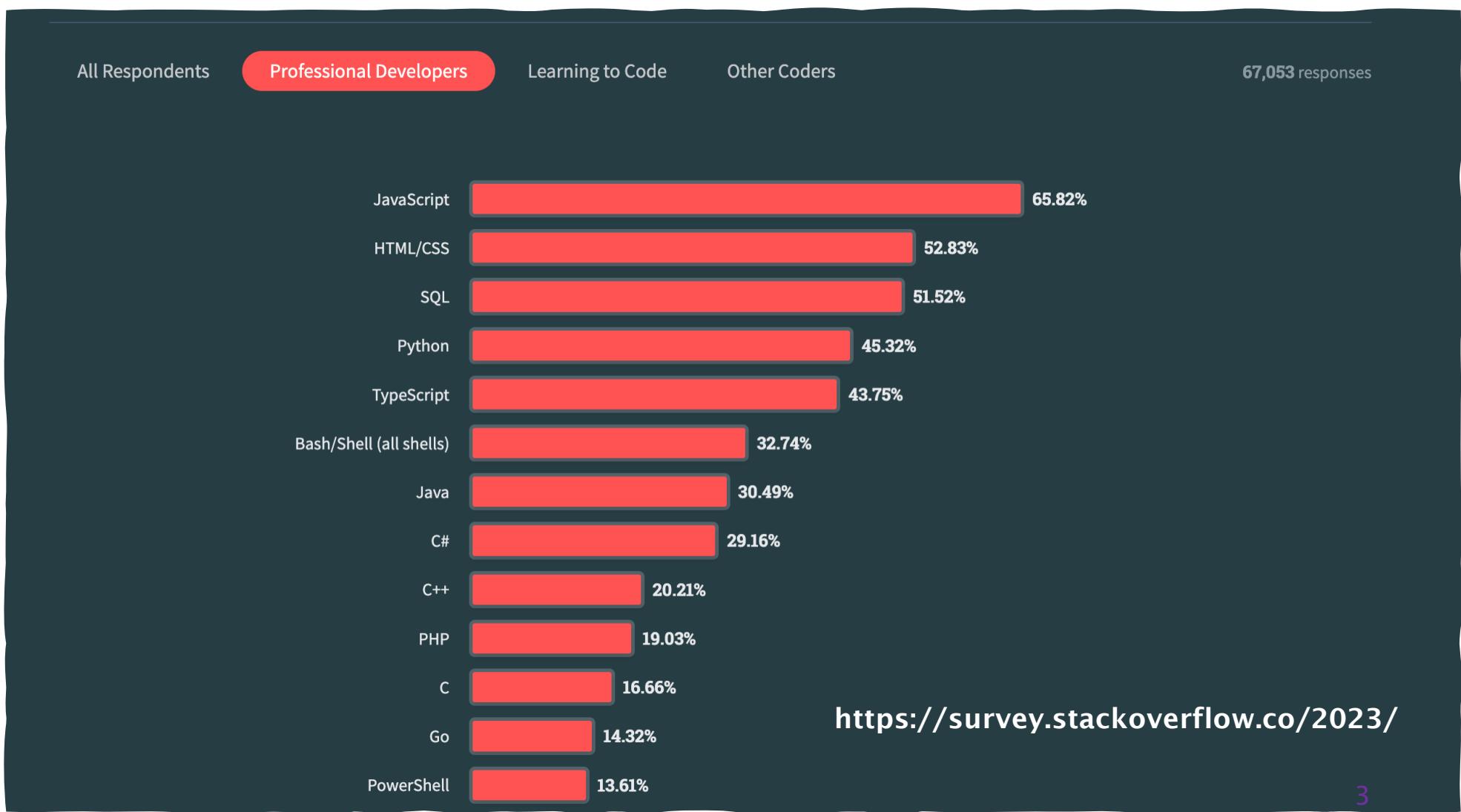
Technical Lead

Trustrace

 <https://www.linkedin.com/in/moorthyptm/>

 <https://github.com/moorthyptm>





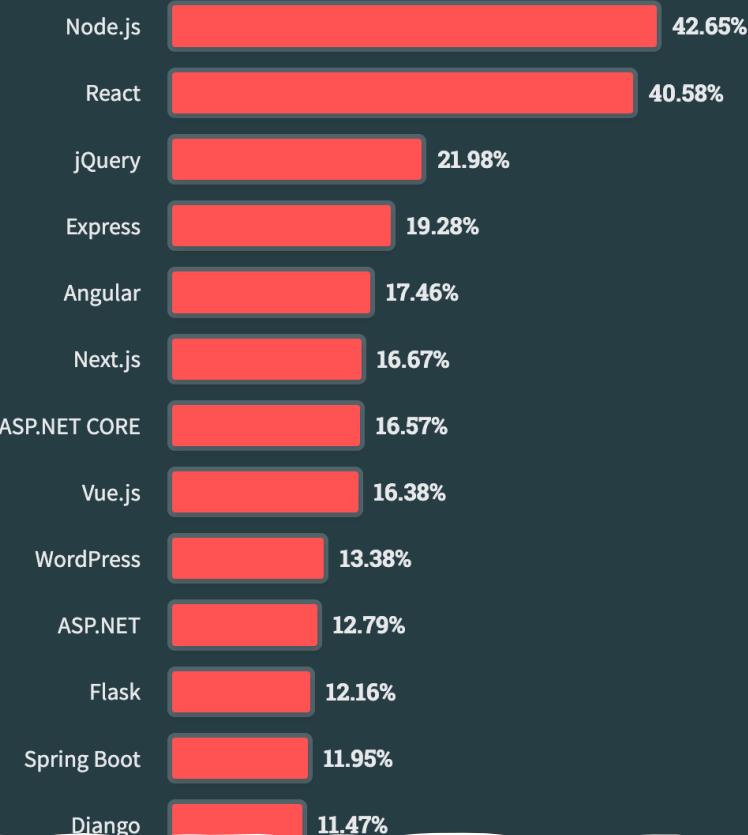
All Respondents

Professional Developers

Learning to Code

Other Coders

71,802 responses



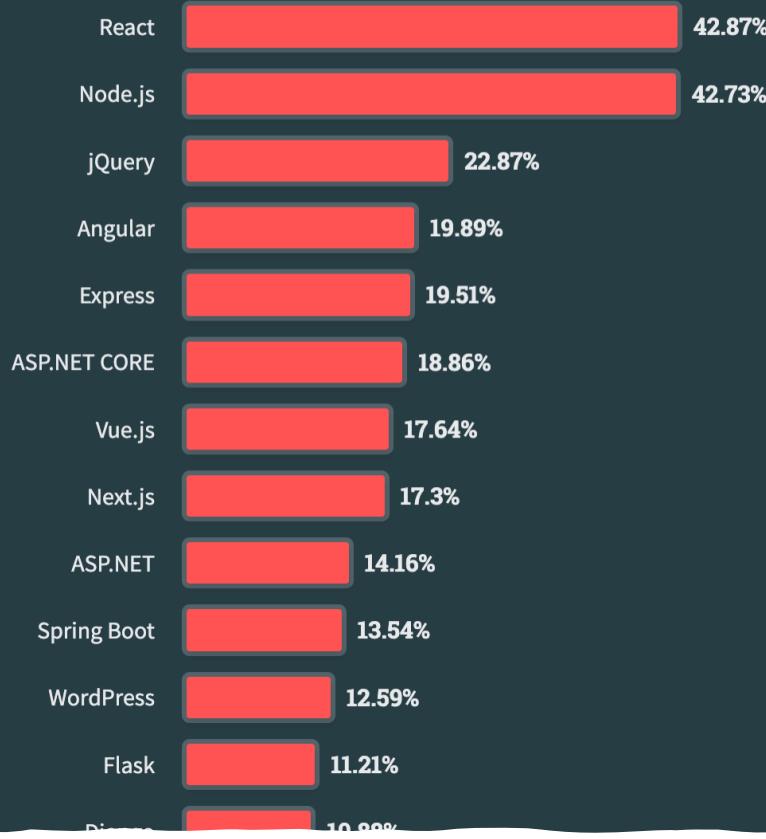
All Respondents

Professional Developers

Learning to Code

Other Coders

56,742 responses



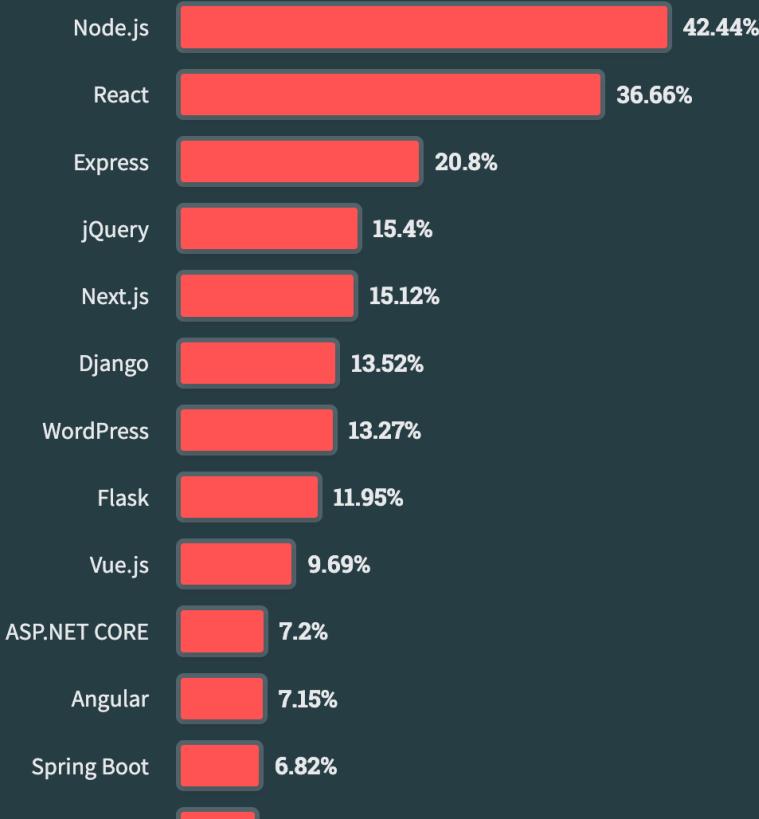
All Respondents

Professional Developers

Learning to Code

Other Coders

3,942 responses



State of JS - 2022

RATIOS OVER TIME 

[Get JSON Data](#) [Get GraphQL Query](#) [Percentages](#) [Rankings](#)

Label	Awareness %								Usage %								Interest %								Retention %									
	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023		
Year	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023	2016	2017	2018	2019	2020	2021	2022	2023		
React	99.5	99.7	99.9	100	100	99.9	99.9	52.7	62	71.6	80.3	80.1	80	81.8	75.4	71.3	67.5	60.9	58.2	48.4	47.2	92.6	93.1	90.6	89.4	87.5	83.7	83						
Vue.js	72.9	94.6	98.7	99.6	99.1	99.7	99.5	10.1	21.6	31.7	46.5	49	51.2	46.2	51.2	70.4	69.6	64.3	62.5	50	50.7	87.3	91.1	91.2	87.2	85.2	80	77.4						
Angular	98.9	99.1	99.9	99.9	99.7	99.6	99.6	20	28.5	57.8	57.8	55.9	54	48.8	50.4	36.3	24.6	23.1	20.8	16.3	20.4	68	66.1	41.4	38	41.7	44.8	42.7						
Preact			71.9	75.7	76.9	79.6	76.1		7.5	12.2	13.5	14.1	12.8				42.7	40.5	39.6	31.7	35.6				83.5	77.8	78.1	73.5	74.3					
Ember	95.6	94.3	92.8	90.9	88.1	86.8	77	14	11.4	11.3	12	11.1	9.2	7.5	27.4	23.6	17.4	18.2	15.5	11.6	13.4	47	41.4	44.7	30.5	27.1	21	17.1						
Svelte				75.4	86.5	93.6	93.5			7.8	14.6	19.9	21.2				66.6	66.1	68.1	69.8					87.7	88.9	89.5	89.7						
Alpine.js					34.1	48.1	51.6				3.4	5.7	6.2					37.6	33.2	32.7						82.3	79.2	76.1						
Lit					27	37.9	41.7				5.4	7.1	6.2					39.5	40.1	36.8						77.6	77.3	70.8						
Solid						38	65.9				2.8	6.3						55.5	65.7								89.5	90.9						
Qwik							48.5					1.8						67											84.6					

State of JS - 2022

Label	Would not use %	Not interested %	Would use again %	Interested %
React	11.8%	8.1%	57.4%	7.2%
Svelte	2%	19.5%	16.9%	45%
Vue.js	8.9%	22.3%	30.3%	22.9%
Solid	0.7%	24.2%	6.7%	46.2%
Qwik	0.4%	22.9%	2.3%	46.4%
Preact	3.5%	43.1%	10%	23.8%
Lit	3%	37.2%	7.2%	21.6%
Alpine.js	2.1%	43.6%	6.6%	21.1%
Angular	23.7%	34.3%	17.6%	8.8%
Stencil	3%	38.6%	4.9%	18.7%
Ember	6.5%	63%	1.3%	9.7%

Angular Latest version

v16.2.0 Latest Compare ▾

 dylhunn released this 2 days ago · 41 commits to main since this release ↗ 16.2.0 ⌂ 096fdd0

16.2.0 (2023-08-09)

benchpress

Commit	Description
dd850b2ab7 fix	correctly report GC memory amounts (#50760)

9

Angular team focus



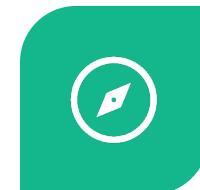
Simplicity



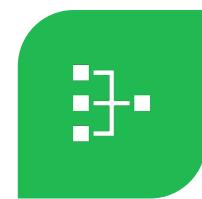
DX



Performance 🚀



SURVEY



RFC – REQUEST FOR
COMMENTS

Component API - old



```
1 import { ComponentFactoryResolver, Directive, ViewContainerRef } from '@angular/core';
2
3 @Directive({ selector: '[appSelector]' })
4 export class NameDirective {
5   constructor(
6     private viewContainerRef: ViewContainerRef,
7     private componentFactoryResolver: ComponentFactoryResolver
8   ) {}
9
10  createMyComponent(): void {
11    const componentFactory = this.componentFactoryResolver.resolveComponentFactory(MyComponent);
12    this.viewContainerRef.createComponent(componentFactory);
13  }
14}
15
```

Component API - New

```
1 import { Directive, ViewContainerRef } from '@angular/core';
2
3 @Directive({ selector: '[appSelector]' })
4 export class NameDirective {
5     constructor(private viewContainerRef: ViewContainerRef) {}
6     createMyComponent(): void {
7         this.viewContainerRef.createComponent(MyComponent);
8     }
9 }
10
```

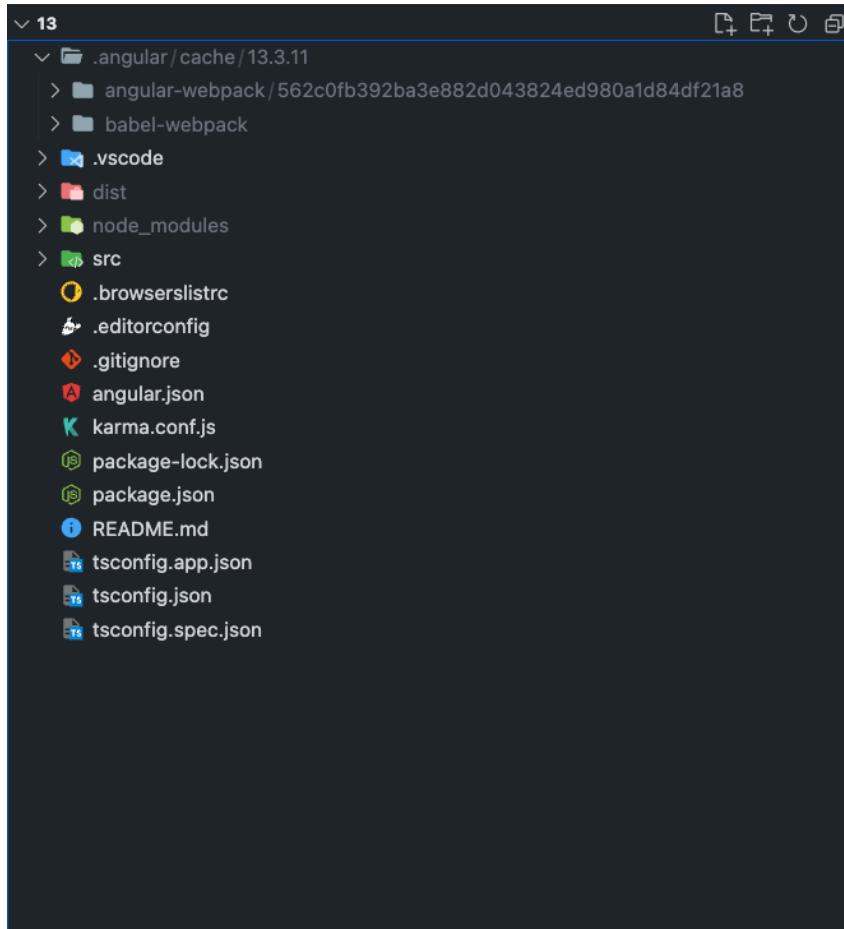
IE11 Support End

- IE-specific polyfills are removed
- So, Bundle size reduced



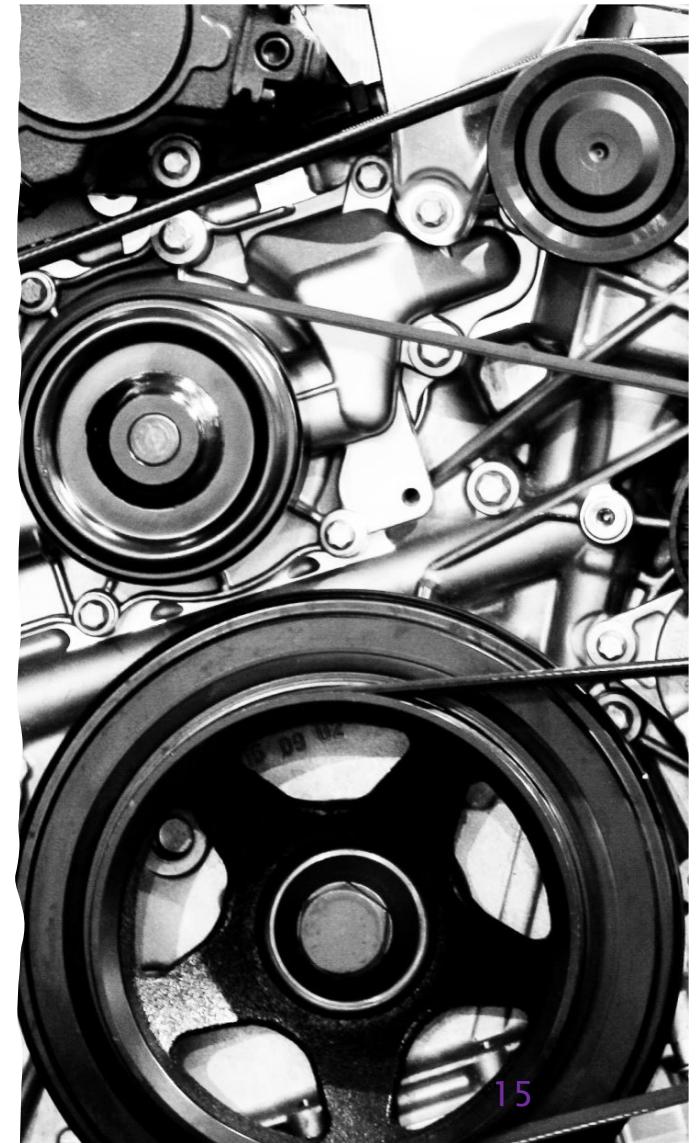
Enable cache Cli

- Ng cache -14



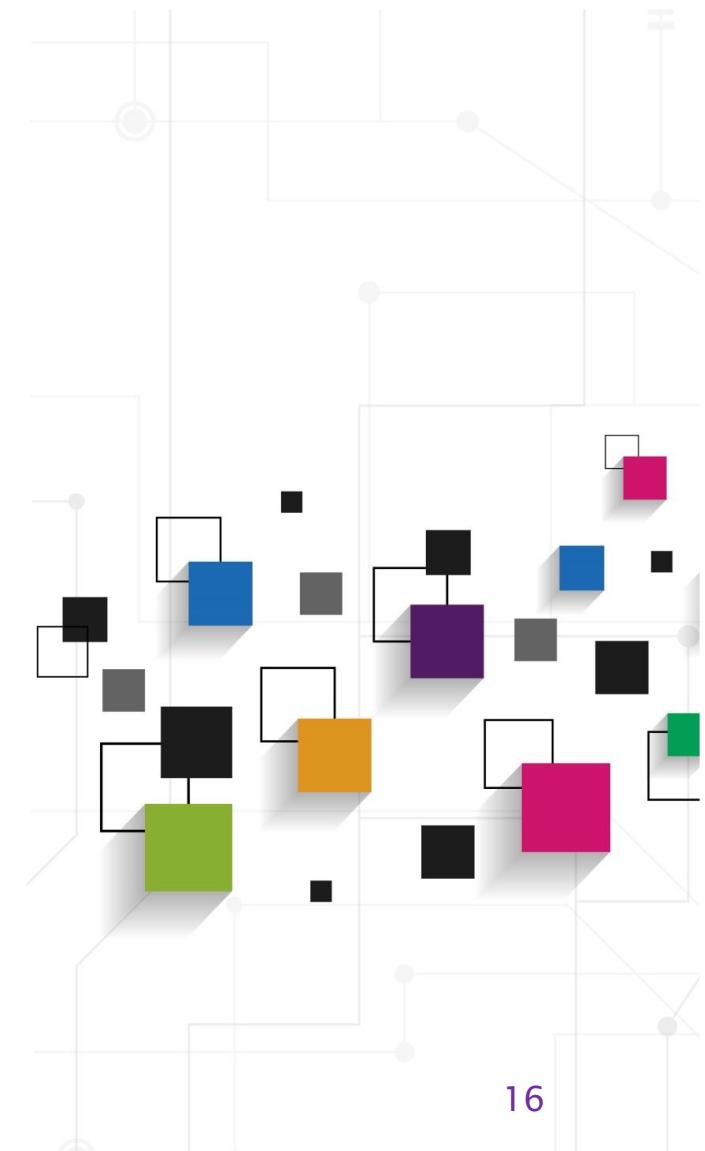
View engine support

- Starting from Angular v13, View Engine is no longer used, focusing solely on Ivy
- improved performance and developer experience
- Ngcc - Angular Compatibility Compiler



Changes to the Angular Package Format (APF)

- Removed older output formats - UMD bundles.
- Modern JS formats standardized ES2020 output.
- Newly developed libraries no longer required to use NGCC
- The node command-line tool exports all of its available outputs to files with a subpath pattern, making them easy for developers to use multiple versions or paths in their projects.



Build optimization -FCP

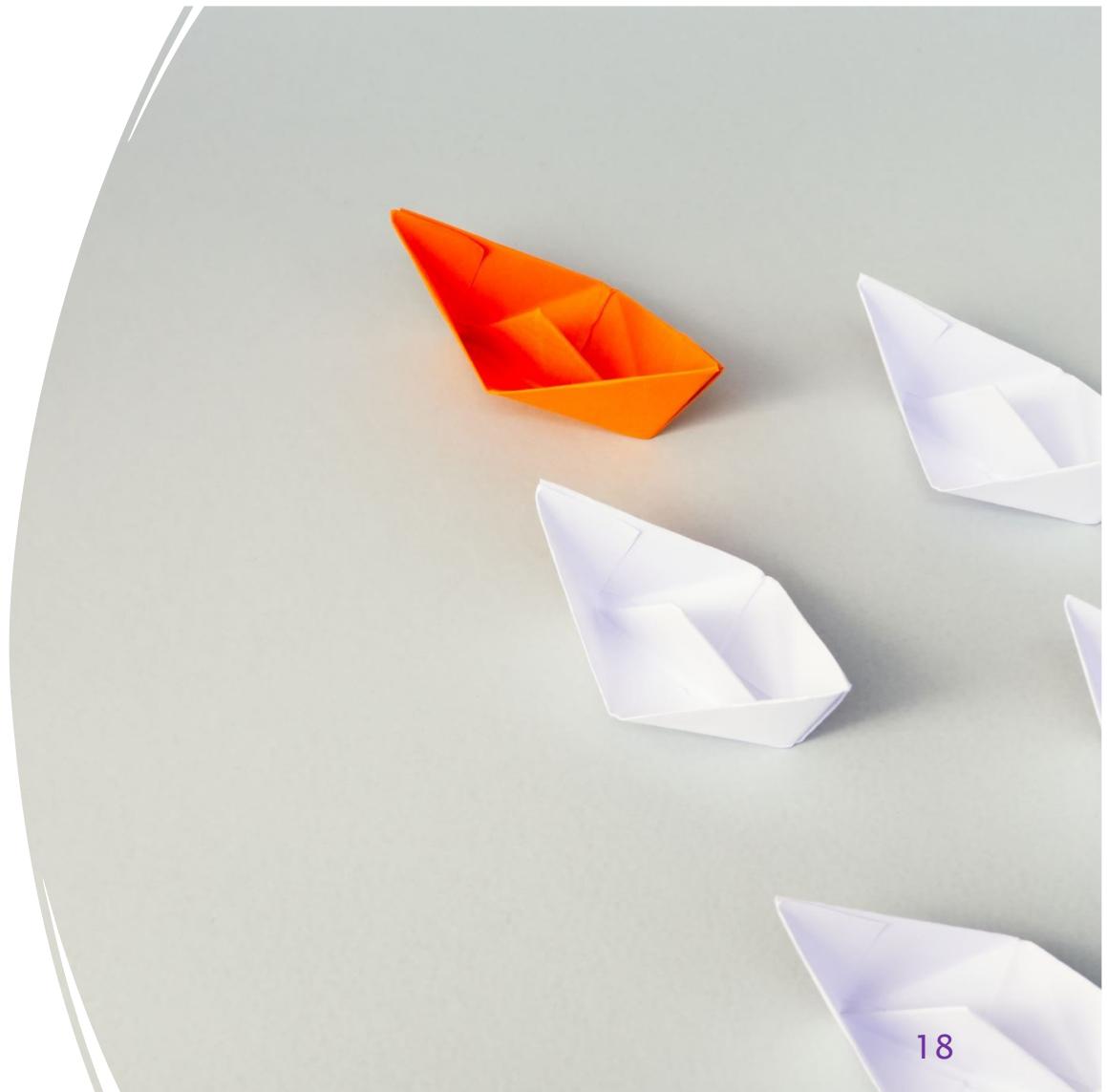
-
- Google fonts -Angular 11
 - Adobe fonts -Angular 13



```
1  "optimization": {  
2    "fonts": {  
3      "inline": true  
4    },  
5    "scripts": true,  
6    "styles": {  
7      "inlineCritical": true,  
8      "minify": true  
9    }  
10 }
```

Community contributions

- Dynamically enable/disable validators:
<https://github.com/angular/angular/pull/42565>
- Restore history after canceled navigation
<https://github.com/angular/angular/pull/38884>



Material - MDC

- Accessibility
 - Contrast
 - Touch targets
 - ARIA

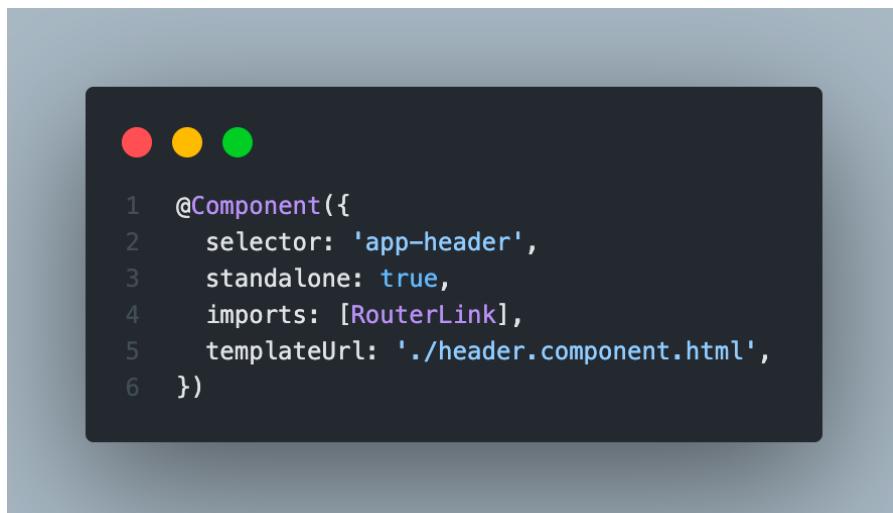


Typed forms and backward compactable

```
const party = new FormGroup({
  address: new FormGroup({
    house: new FormControl(123),
    street: new FormGroup({
      name: new FormControl('Powell'),
      kind: new FormControl('St'),
    }),
  }),
  formal: new FormControl(true),
  foodOptions: new FormArray([
    new FormControl('Soup'),
  ])
});
```

Standalone Components - Preview

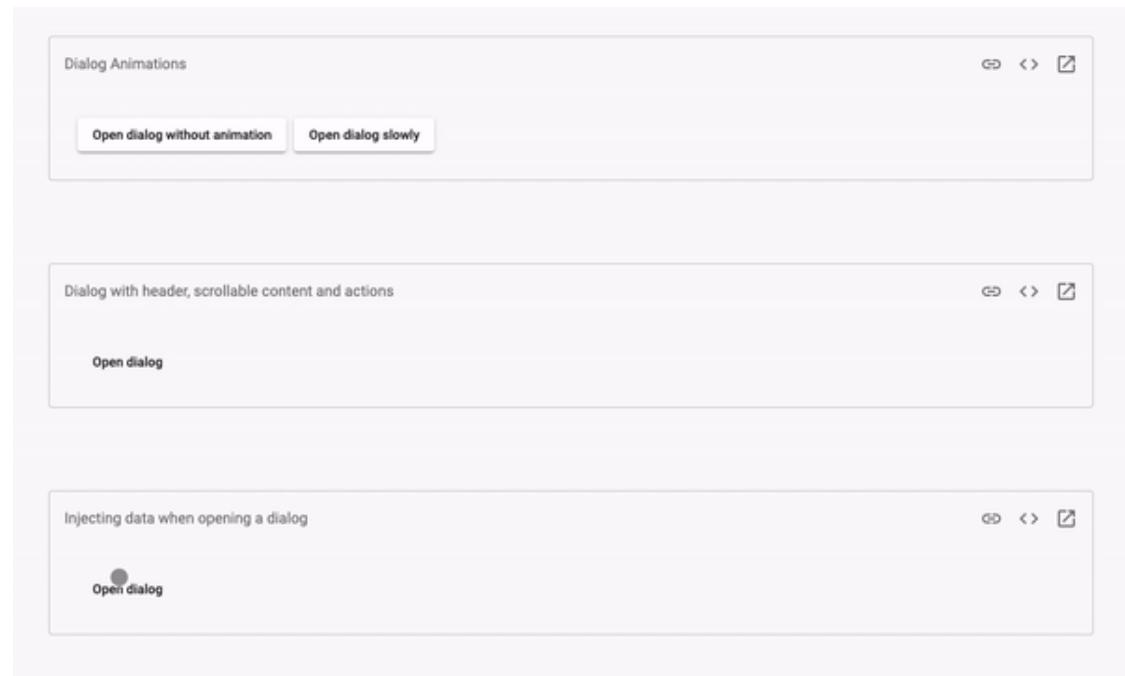
- Component
- Directive
- Pipe

A screenshot of a dark-themed code editor window. At the top, there are three circular icons: red, yellow, and green. Below them, a snippet of TypeScript code is displayed in a monospaced font:

```
1 @Component({
2   selector: 'app-header',
3   standalone: true,
4   imports: [RouterLink],
5   templateUrl: './header.component.html',
6 })
```

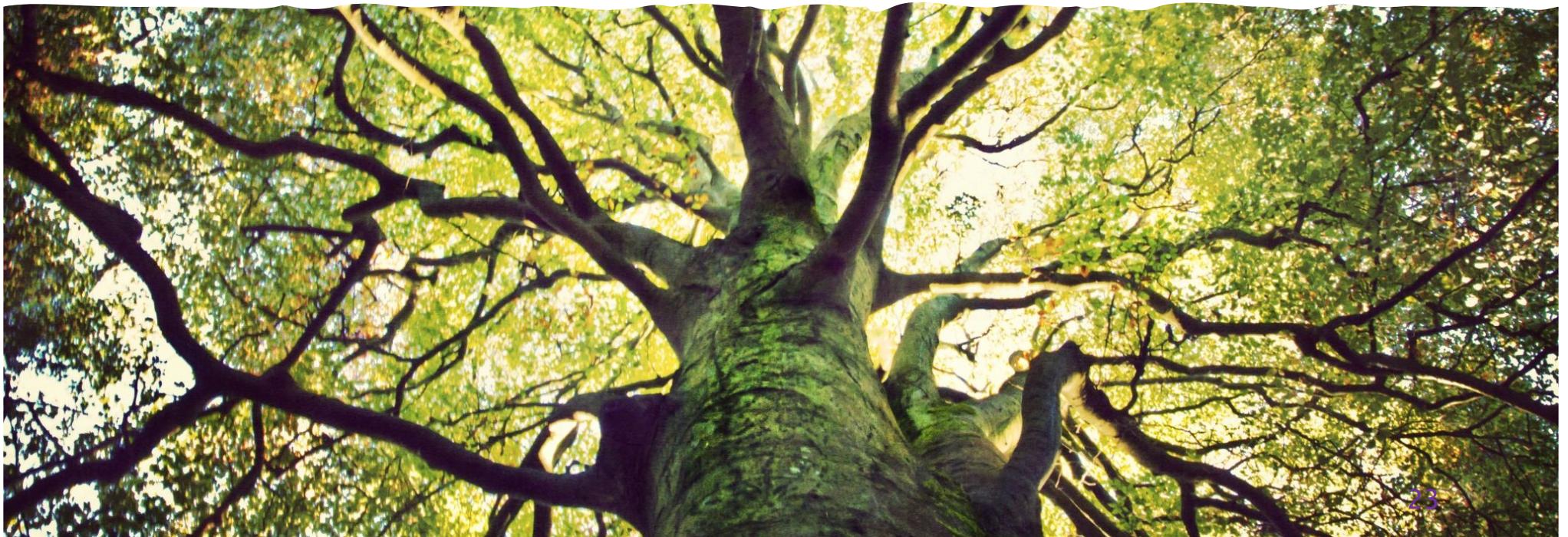
New primitives Angular CDK

- Modal
- Menu



Git: Main branch

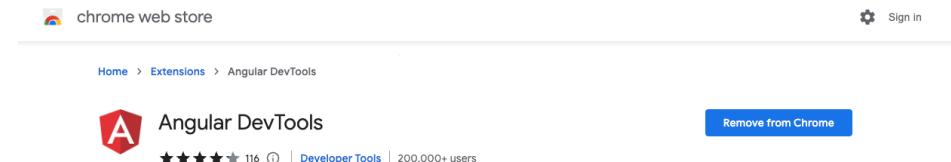
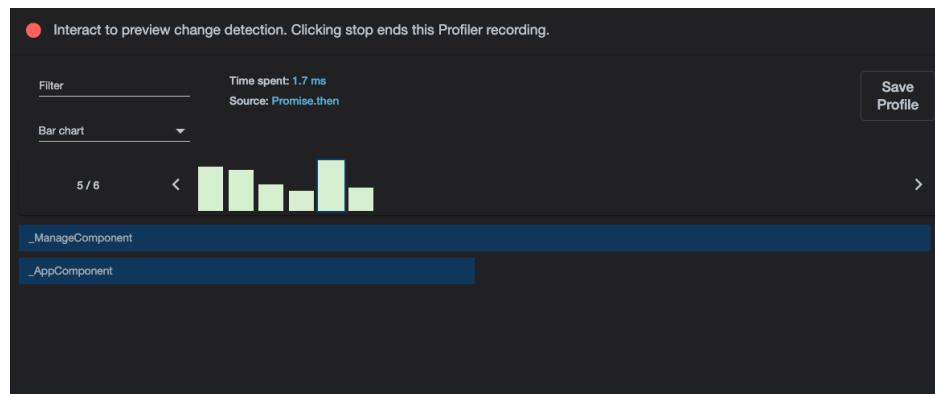
Inclusive community



Tree shakeable error messages

```
● ● ●  
1  @Component({...})  
2  class MyComponent {}  
3  
4  @Directive({...})  
5  class MyDirective extends MyComponent {} // throws an error at runtime  
6  
7  // Before v14 the error is a string:  
8  > Directives cannot inherit Components. Directive MyDirective is attempting to extend component MyComponent.  
9  
10 // Since v14 the error code makes this tree-shakeable:  
11 > NG0903: Directives cannot inherit Components. Directive MyDirective is attempting to extend component MyComponent.  
12  
13 // v14 production bundles preserve the error code, tree-shaking strings and making the bundle XX smaller:  
14 > NG0903
```

Angular change detection and runtime optimization

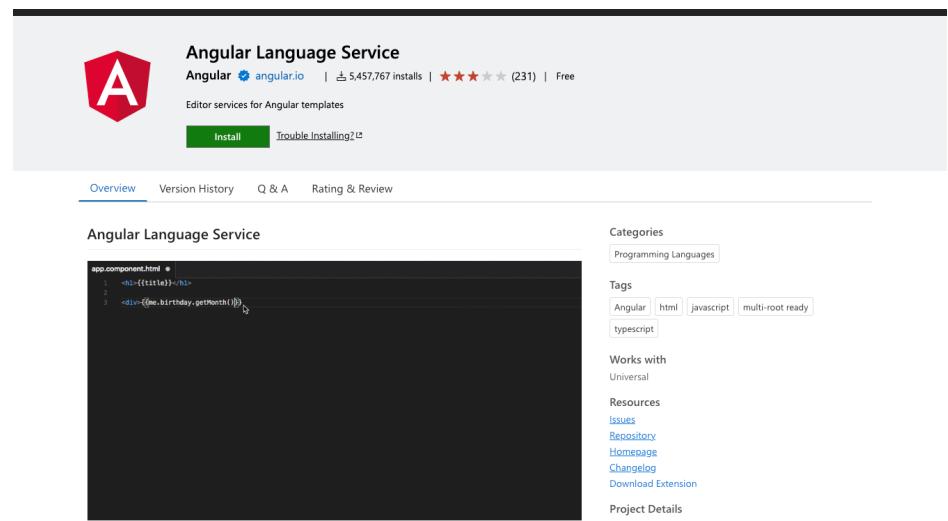


Angular language service



A screenshot of a code editor showing a configuration file for the Angular language service. The file contains the following JSON code:

```
1  {
2    "angularCompilerOptions": {
3      "extendedDiagnostics": {
4        // The categories to use for specific diagnostics.
5        "checks": {
6          // Maps check name to its category.
7          "invalidBananaInBox": "error"
8          "nullishCoalescingNotNullable": "warning"
9        },
10       // The category to use for any diagnostics not listed in `checks` above.
11       "defaultCategory": "suppress"
12     },
13   ...
14 },
15 ...
16 }
```



NgOptimizedImage

```
1 export const APP_ROUTES: Route[] = [
2   {
3     path: '',
4     loadComponent: () =>
5       import('./app/pokemon.component').then((m) => m.PokemonComponent),
6     providers: [
7       provideHttpClient(),
8       PokeService,
9       {
10         provide: IMAGE_LOADER,
11         useValue: (config: ImageLoaderConfig) => {
12           return `https://raw.githubusercontent.com/PokeAPI/
13             sprites/master/
14             sprites/pokemon/other/home/${config.src}`;
15         },
16       },
17     ],
18   },
19 ];
```

```
1 export const APP_ROUTES: Route[] = [
2   {
3     path: '',
4     loadComponent: () =>
5       import('./app/pokemon.component').then((m) => m.PokemonComponent),
6     providers: [
7       provideHttpClient(),
8       PokeService,
9       {
10         provide: IMAGE_LOADER,
11         useValue: (config: ImageLoaderConfig) => {
12           return `https://raw.githubusercontent.com/PokeAPI/
13             sprites/master/
14             sprites/pokemon/other/home/${config.src}`;
15         },
16       },
17     ],
18   },
19 ];
```

Accessibility: Page title

- TitleStrategy

```
1 import { Routes } from '@angular/router';
2
3 export const routes: Routes = [
4   {
5     path: '',
6     title: 'ToDo 📝 Home',
7     loadComponent: () =>
8       import('./home/home.component').then((m) => m.HomeComponent),
9   },
10  {
11    path: 'manage',
12    title: 'ToDo 📝 Manage',
13    loadComponent: () =>
14      import('./manage/manage.component').then((m) => m.ManageComponent),
15  },
16  {
17    path: 'list',
18    title: 'ToDo 📝 List',
19    loadComponent: () =>
20      import('./list/list.component').then((m) => mListComponent),
21  },
22];
23
```

Route providers

```
● ● ●  
1 import { NgModule } from '@angular/core';  
2 import { RouterModule, Routes } from '@angular/router';  
3 import { MockService } from './user/mock';  
4  
5 const routes: Routes = [  
6   {  
7     path: 'user/:orgId',  
8     loadComponent: () => import('./user/user.component'),  
9     providers: [MockService],  
10    },  
11  ];  
12  
13 @NgModule({  
14   imports: [RouterModule.forRoot(routes, { bindToComponentInputs: true })],  
15   exports: [RouterModule],  
16 })  
17 export class AppRoutingModule {}  
18
```

Component Protected props binding



```
1 @Component({
2   selector: 'app-component',
3   template: '{{ title }}', // Now compiles!
4 })
5 export class AppComponent {
6   protected title = 'Hello world';
7 }
```

```
Error: src/app/app.component.html:344:14 - error TS2445: Property 'title' is protected and only accessible within class 'AppComponent' and its subclasses.
344   <span>{{ title }} app is running!</span>
```

Inject



```
1 constructor(private todoService: TodoService) {}
```

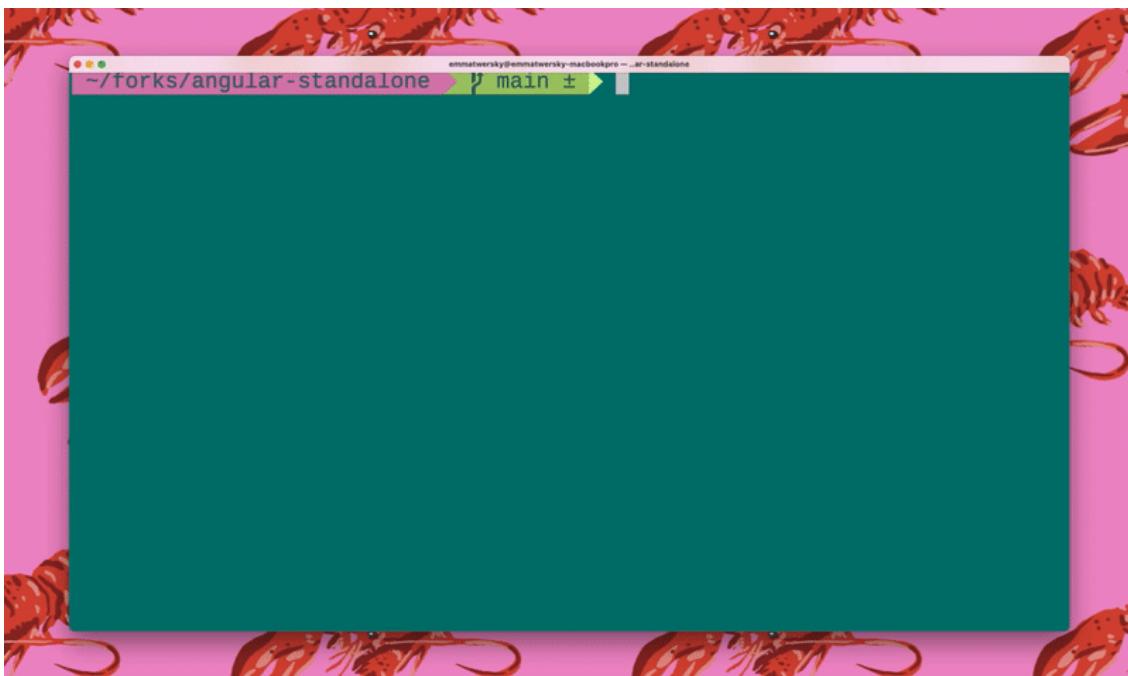


```
1  
2 todoService:TodoService;  
3 constructor(private injector:Injector) {  
4     this.todoService = injector.get<TodoService>(TodoService)  
5 }  
6
```

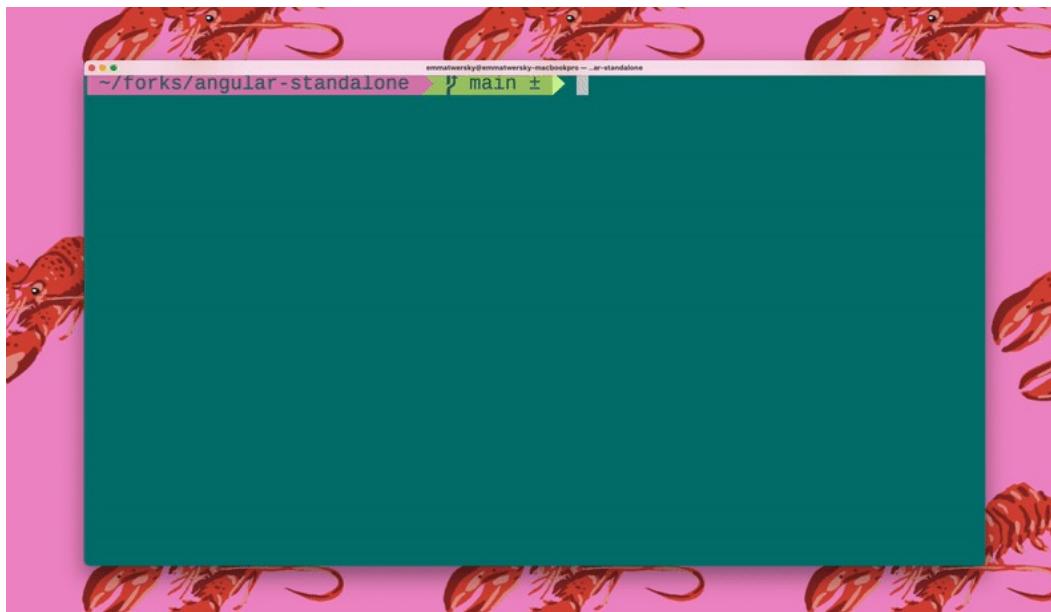


```
1 private todoService = inject(TodoService);
```

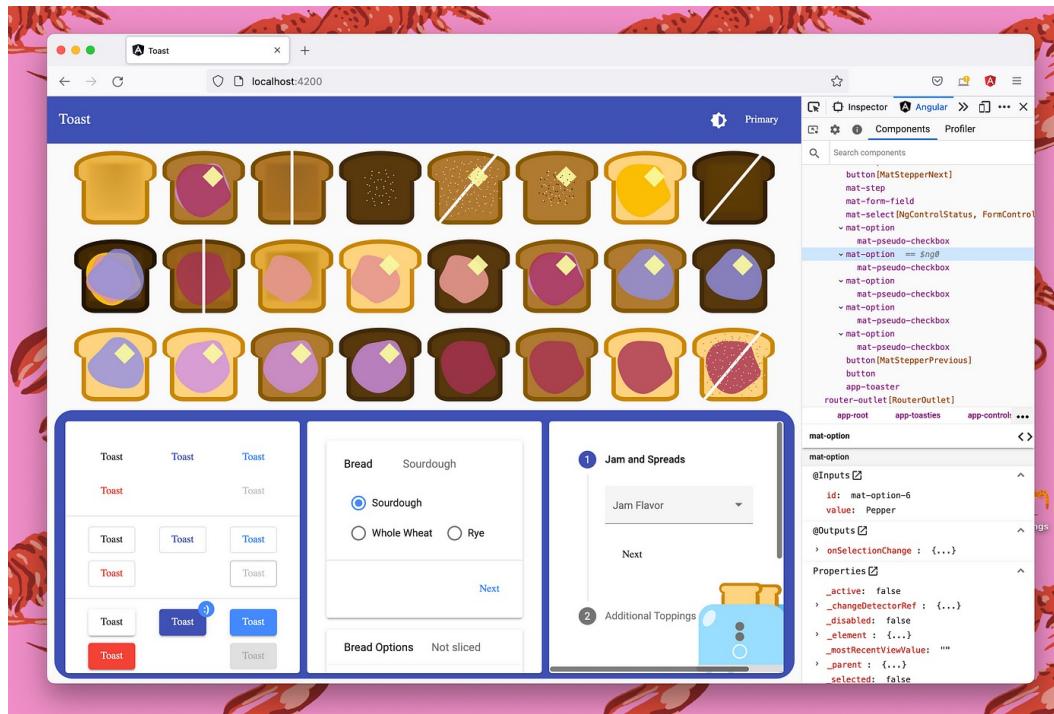
ng completion



ng analytics

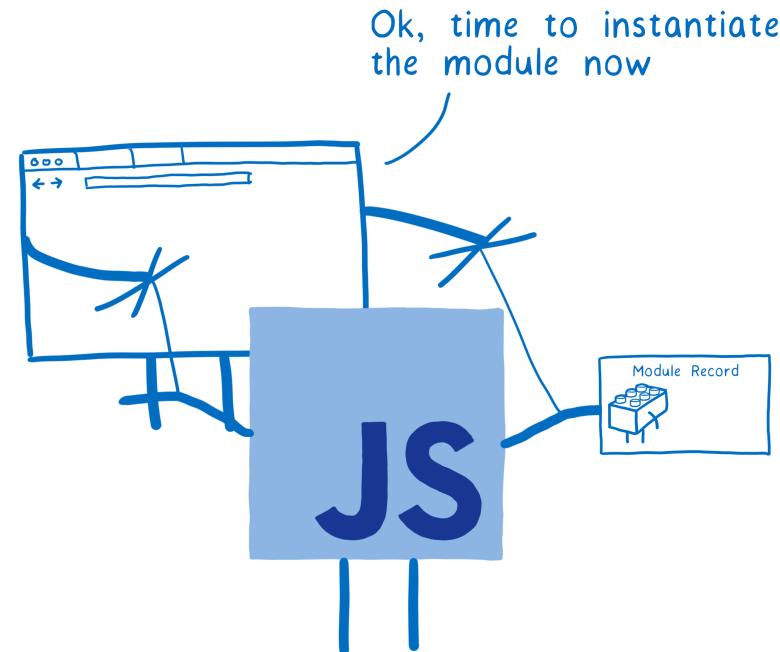


Angular DevTools is available offline and in Firefox

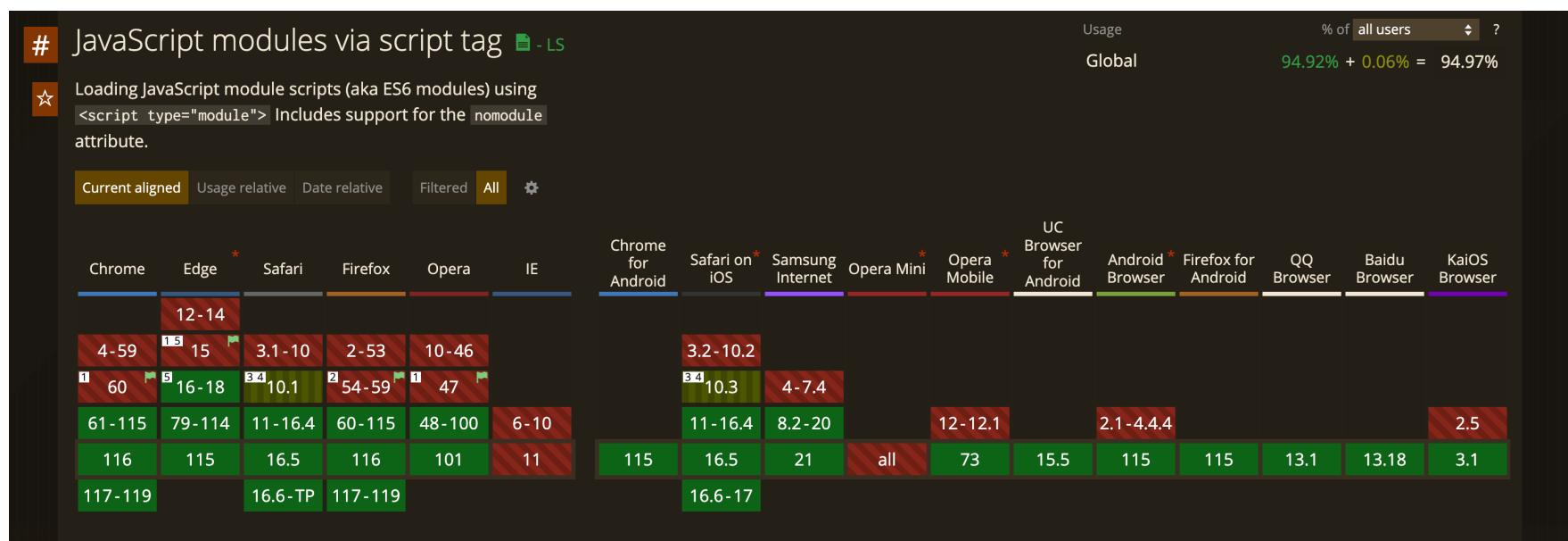


Experimental ESM Application Builds

- "builder": "@angular-devkit/build-angular:browser-esbuild"

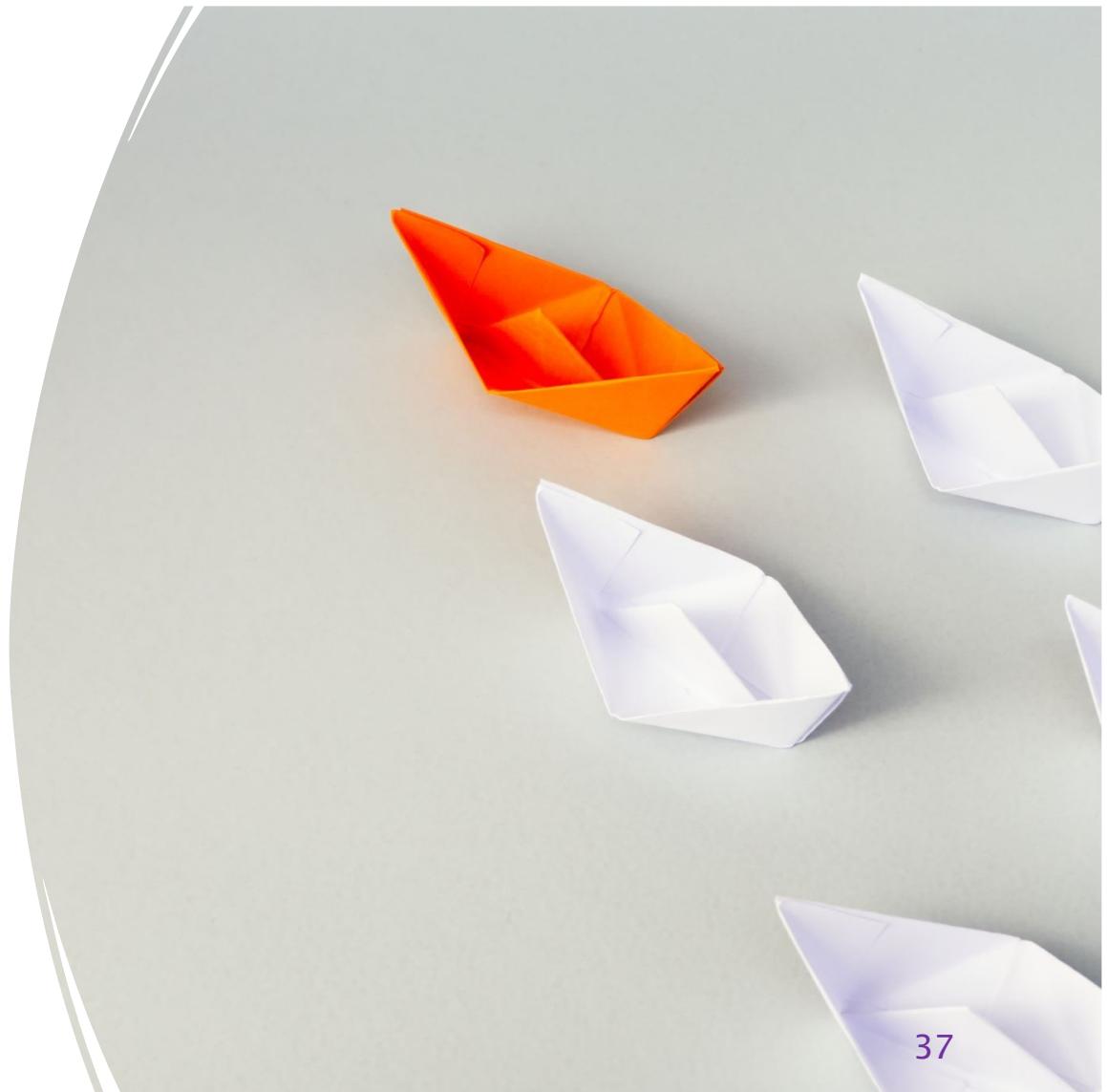


ESM



Community contribution

- NgModel changes are reflected in the UI for OnPush components
<https://github.com/angular/angular/issues/10816>
- Tree-shakeable error messages and etc..



15

- Standalone APIs are stable
- Image directive is now stable
- `ng new myAppName --standalone`
- `ng generate @angular/core:standalone -15.2.0 or later.`



Router and HttpClient tree-shakable standalone APIs

```
1 import {Routes} from '@angular/router';
2
3 import {LazyComponent} from './lazy.component';
4
5 export const lazyRoutes: Routes = [{path: '', component: LazyComponent}];
```

```
1 export const appRoutes: Routes = [
2   {
3     path: "lazy",
4     loadChildren: () =>
5       import("./lazy/lazy.routes").then((routes) => routes.lazyRoutes),
6   },
7 ];
8
```

```
1 bootstrapApplication(AppComponent, {
2   providers: [
3     provideRouter(appRoutes)
4   ]
5 });
6 );
```

Directive composition API

```
1 @Component({
2   selector: 'mat-menu',
3   hostDirectives: [HasColor, {
4     directive: CdkMenu,
5     inputs: ['cdkMenuDisabled: disabled'],
6     outputs: ['cdkMenuClosed: closed']
7   }]
8 })
9 class MatMenu {}
```

Functional router guards

```
● ● ●  
1 const route = {  
2   path: 'user',  
3   canActivate: [() => inject(AuthService).isLoggedIn()]  
4 };
```

default imports



```
1 const routes: Routes = [
2   {
3     path: 'app',
4     loadComponent:
5       () => import('./app-shell/app-shell.component'),
6   },
7 ];
```

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-app-shell',
5   templateUrl: './app-shell.component.html',
6   standalone: true,
7   styleUrls: ['./app-shell.component.scss'],
8 })
9 export default class AppShellComponent {}
```

MDC

- Material Design Components for Web (MDC)
- Range selection
- CDK Listbox



```
1  
2 import {MatLegacyButtonModule} from '@angular/material/legacy-button';  
3
```

Experimental esbuild support

- Sass, SVG template, file replacement
- ng build -watch support



Automatic imports in language service

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'standalone-boo',
  standalone: true,
  template: `
    <standalone-foo></standalone-foo>
    `,
  styles: [],
  imports: [CommonModule]
})
export class StandaloneBooComponent {}
```

Community contribution

```
● ● ●  
1  
2 bootstrapApplication(AppComponent, {  
3   providers: [  
4     {  
5       provide: DATE_PIPE_DEFAULT_OPTIONS,  
6       useValue: { dateFormat: 'shortDate' }  
7     }  
8   ]  
9 });  
10
```

Deprecations

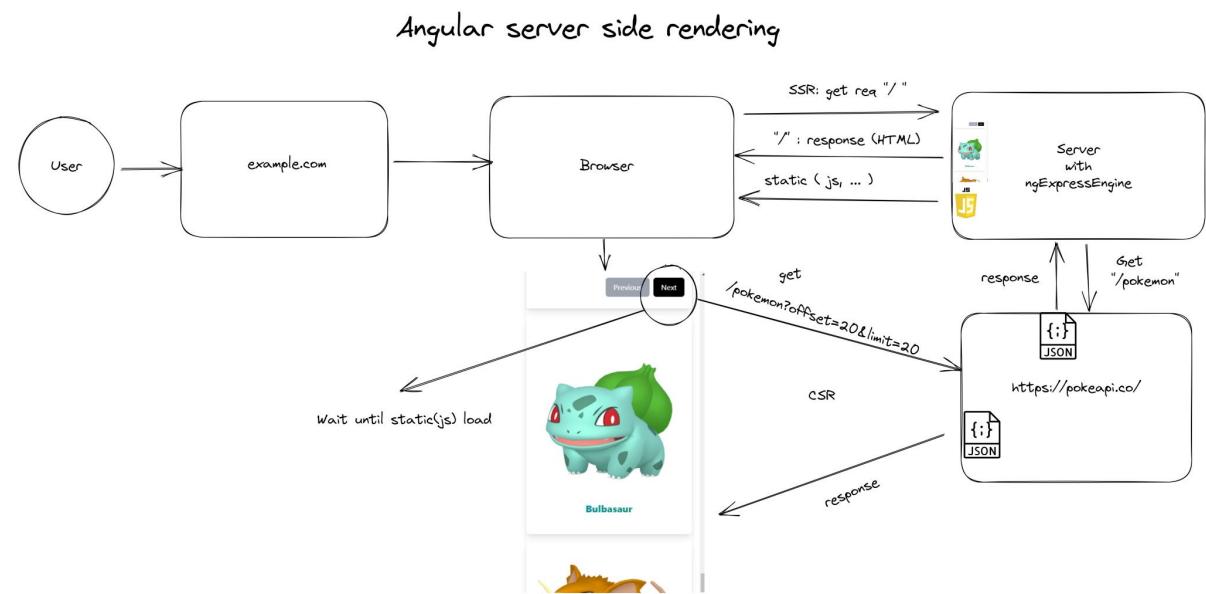
- providedIn: 'any' NgModule
- @angular/flex-layout – stop publishing new releases



Signal Preview

```
1 import { Component, computed, effect, signal } from '@angular/core';
2
3 @Component({
4   selector: 'app-counter',
5   template: `
6     <div class="h-[90ch] w-full flex flex-col items-center justify-center space-y-3" style="background-color: #f0f0f0; padding: 20px;">
7       <div>
8         <h1>Counter : {{ counter() }}</h1>
9         <h1>Doubled : {{ doubledCount() }}</h1>
10      </div>
11
12      <div style="text-align: center; margin-top: 20px;">
13        <button class="bg-black hover:bg-gray-700 text-white px-4 py-2 rounded-lg" (click)="increment()" style="color: white; border: none; font-weight: bold; font-size: 1em; width: fit-content; height: fit-content; padding: 10px; border-radius: 10px; transition: all 0.3s ease; margin-bottom: 10px;">
14          Increment
15        </button>
16      </div>
17    </div>
18  `,
19  styles: [``],
20})
21 export class CounterComponent {
22   counter = signal(0);
23   doubledCount = computed(() => this.counter() * 2);
24   constructor() {
25     effect(() => console.log(this.counter()));
26   }
27   increment() {
28     this.counter.set(this.counter() + 1);
29   }
30 }
31
32
33
34
35
36 }
```

Server-side rendering (SSR) with Angular Universal



16

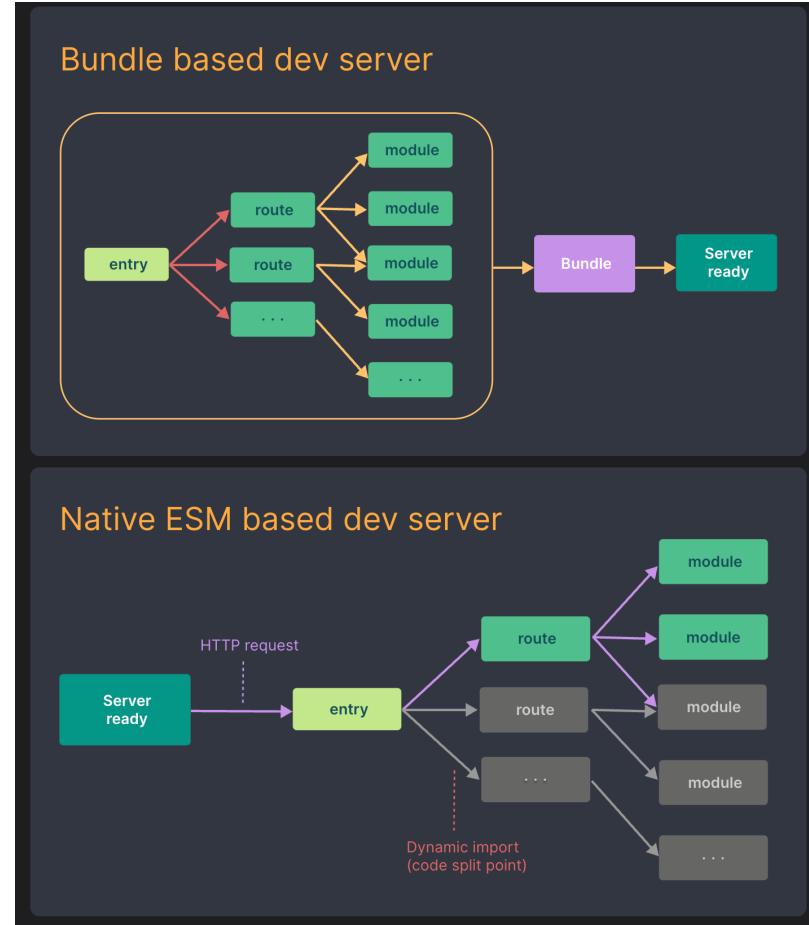
- Signals -Stable
- SSR- Hydration
- Simplified project structure
- CSP support for inline styles



zone configuration

```
1 import { bootstrapApplication } from '@angular/platform-browser';
2 import { AppComponent } from './app/app.component';
3 import { NgZone, provideZoneChangeDetection, eNoopNgZone } from '@angular/core';
4
5 // import { AppModule } from './app/app.module';
6 // import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
7
8 // platformBrowserDynamic()
9 //   .bootstrapModule(AppModule, {
10 //     //ngZone: 'noop',
11 //     ngZoneEventCoalescing: true,
12 //   })
13 //   .catch((err) => console.error(err));
14
15 class CustomZone extends eNoopNgZone {}
16 const instance = new CustomZone();
17
18 bootstrapApplication(AppComponent, {
19   providers: [
20     provideZoneChangeDetection({
21       eventCoalescing: true,
22     }),
23     // {
24     //   provide: NgZone,
25     //   useValue: instance,
26     // },
27   ],
28 });
29
```

Vite preview



Karma

Karma

code style standard npm v6.4.2 downloads 11M/month

maintainability F prs welcome david no longer available david no longer available semantic-release

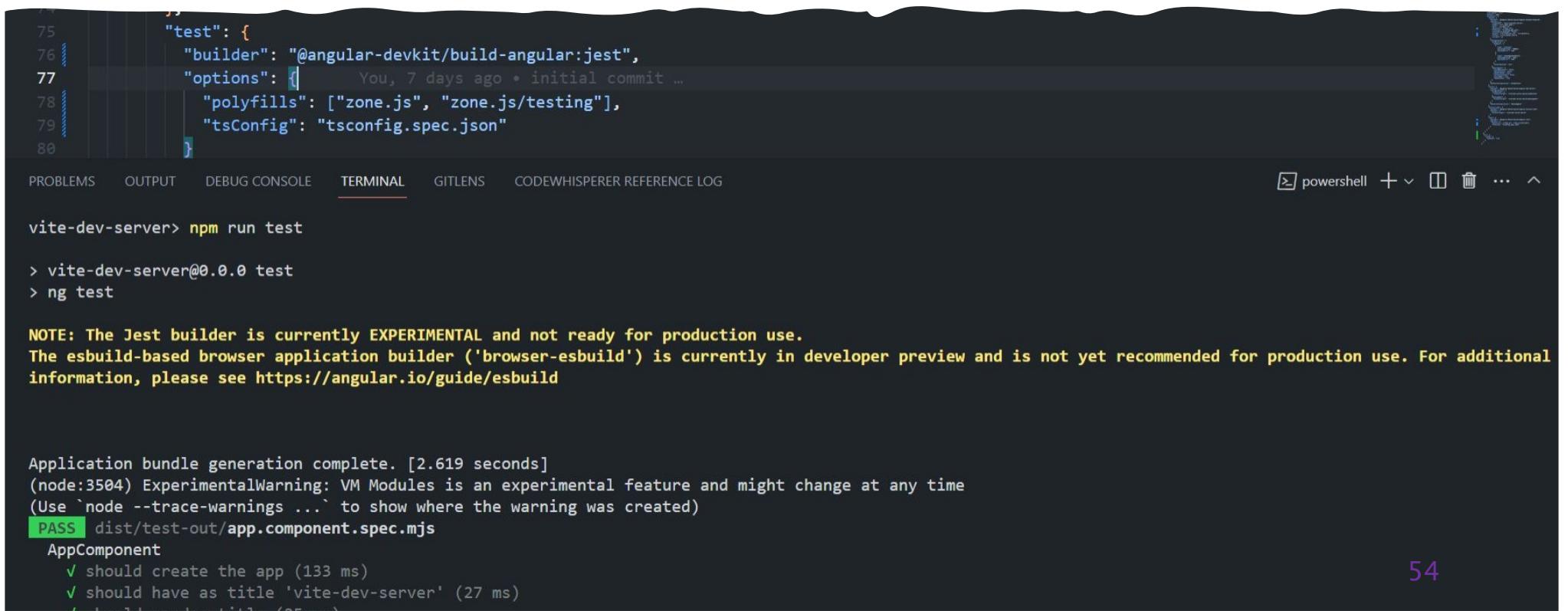
A simple tool that allows you to execute JavaScript code in multiple *real* browsers.

The main purpose of Karma is to make your test-driven development easy, fast, and fun.

Karma is deprecated and is not accepting new features or general bug fixes.

The web testing space has evolved significantly in the [10+ years](#) since Karma's creation. The web landscape looks very different today and new patterns and tools have emerged in the ecosystem. New test runners offer more performant alternatives, and Karma no longer provides clear unique value.

Jest



A screenshot of a terminal window within a code editor interface. The terminal shows the command `vite-dev-server> npm run test` being run, which triggers a build process. The output includes a note about the Jest builder being experimental and a warning about esbuild-based browser application builder. The test results show a single test named `AppComponent` with two passing tests: `should create the app` and `should have as title 'vite-dev-server'`. The terminal also displays some experimental warnings from Node.js.

```
75      "test": {  
76        "builder": "@angular-devkit/build-angular:jest",  
77        "options": { You, 7 days ago * initial commit ...  
78          "polyfills": ["zone.js", "zone.js/testing"],  
79          "tsConfig": "tsconfig.spec.json"  
80        }  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS CODEWHISPERER REFERENCE LOG powershell + × ☰ ... ^  
vite-dev-server> npm run test  
> vite-dev-server@0.0.0 test  
> ng test  
  
NOTE: The Jest builder is currently EXPERIMENTAL and not ready for production use.  
The esbuild-based browser application builder ('browser-esbuild') is currently in developer preview and is not yet recommended for production use. For additional information, please see https://angular.io/guide/esbuild  
  
Application bundle generation complete. [2.619 seconds]  
(node:3504) ExperimentalWarning: VM Modules is an experimental feature and might change at any time  
(Use `node --trace-warnings ...` to show where the warning was created)  
PASS dist/test-out/app.component.spec.mjs  
AppComponent  
  ✓ should create the app (133 ms)  
  ✓ should have as title 'vite-dev-server' (27 ms)
```

Vitest

Blazing Fast Unit Test Framework

A Vite-native unit test framework. It's fast!

[Get Started](#)[Features](#)[Why Vitest?](#)[View on GitHub](#)

Vite Powered

Reuse Vite's config, transformers, resolvers, and plugins - consistent across your app and tests.

Jest Compatible

Expect, snapshot, coverage, and more - migrating from Jest is straightforward.

Smart & instant watch mode

Only rerun the related changes, just like HMR for tests!

ESM, TypeScript, JSX

Out-of-box ESM, TypeScript and JSX support powered by esbuild

The screenshot shows an IDE interface with two code editors and a terminal. The left editor displays the template file `app.component.html`, which contains a single component definition:

```
src > app > app.component.html > app-demo
Go to component
1 <app-demo></app-demo>
```

The right editor displays the component file `demo.component.ts`:

```
src > app > demo > demo.component.ts > DemoComponent
1 import { Component, Input } from '@angular/core';
2
3 @Component({
4   selector: 'app-demo',
5   templateUrl: './demo.component.html',
6   styleUrls: ['./demo.component.scss'],
7   standalone: true
8 })
9 export class DemoComponent {
10   @Input({ required: true, alias: "title" }) testTitle!: string;
11
12
13 }
```

Below the editors, the IDE navigation bar shows tabs: PROBLEMS (1), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS (1), and COMMENTS.

The terminal window shows the following output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 1 PORTS 1 COMMENTS
6 | templateUrl: './app.component.html',
```

→ Local: http://localhost:4200/
Watch mode enabled. Watching for file changes...
X [ERROR] NG8008: Required input 'title' from component DemoComponent must be specified. [plugin angular-compiler]

```
src/app/app.component.html:0:0:
0 | ^
```

Error occurs in the template of component AppComponent.

```
src/app/app.component.ts:6:15:
6 |   templateUrl: './app.component.html',
```

Reloading client(s)... □

Mandatory inputs

self-closing tags

The screenshot shows an Angular 16 application running in VS Code. The code editor displays a component template with a self-closing tag:

```
1 <app-home-qwerty-abc />
2
3
4 <h1>Hello</h1>
5
6
7 <!-- <router-outlet></router-outlet> -->
8
```

The component is rendered in the browser as:

Test

Hello

The browser's developer tools show the rendered HTML structure, highlighting the self-closing tag in the component template.

Angular 16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS ...

+ v ... ^ X

| node

| cmd

✓ Compiled successfully.
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size

takeUntilDestroyed

```
1 import { Component } from '@angular/core';
2 import { takeUntilDestroyed } from '@angular/core/rxjs-interop';
3 import { interval } from 'rxjs';
4
5 @Component({
6   selector: 'app-home-qwerty-abc',
7   standalone: true,
8   template: `<p>Hello</p>`,
9 })
10 export class HomeComponent {
11   interval = interval(1000);
12   constructor() {
13     // Angular 16 : takeUntilDestroyed Example
14     this.interval.pipe(takeUntilDestroyed()).subscribe(console.log);
15   }
16 }
17
```

DestroyRef

```
● ● ●
1 import { Component, DestroyRef, inject } from '@angular/core';
2
3 @Component({
4   selector: 'app-home-qwerty-abc',
5   standalone: true,
6   template: `<p>Hello</p>`,
7 })
8 export class HomeComponent {
9   constructor() {
10     inject(DestroyRef).onDestroy(() => {
11       // TODO
12       console.log('HomeComponent Destroyed');
13     });
14   }
15 }
16
```

Input transformation

```
1 import { Directive, Input } from '@angular/core';
2
3 @Directive({ selector: 'mat-checkbox' })
4 export class MatCheckbox {
5   @Input()
6   get disabled() {
7     return this._disabled;
8   }
9   set disabled(value: any) {
10     this._disabled =
11       typeof value === 'boolean' ? value : value != null && value !== 'false';
12   }
13   private _disabled = false;
14 }
15
```

```
1 import { Directive, Input, booleanAttribute } from '@angular/core';          16.1.0
2
3 @Directive({ selector: 'mat-checkbox' })
4 export class MatCheckbox {
5   @Input({ transform: booleanAttribute }) disabled: boolean = false;
6 }
7
```

Fetch/xhr

The screenshot shows a browser window with two tabs: '16ExpStandalone' and another tab. The '16ExpStandalone' tab displays a JSON response from a Pokémon API endpoint. The code editor shows an Angular configuration file with providers for 'provideRouter' and 'provideHttpClient' with 'withFetch'. The browser's developer tools Network tab is open, showing two requests for 'pokemon' data. One request is labeled 'xhr - Default' and the other 'With fetch'. The 'With fetch' request is highlighted with a green border and has a shorter duration (2 ms) compared to the 'xhr' request (3 ms). The Network tab includes a timeline at the top and a detailed table below.

xhr - Default

With fetch

Name	Status	Type	Initiator	Size	Time	Waterfall
pokemon	200	xhr		(disk cache)	3 ms	
pokemon	200	fetch	app.component.html:1	(disk cache)	2 ms	

61

ngcc



bindToComponentInputs

```
✓ import { Component, Input, inject } from '@angular/core';
import { MockService } from './mock';
import { Observable } from 'rxjs';

✓ @Component({
  selector: 'app-user',
  standalone: true,
  templateUrl: './user.component.html',
  providers: [MockService],
})
✓ export default class UserComponent {
  ✓ @Input() set orgId(orgId: string) {
    console.log(orgId); ✘ 'abc123' ✘ 'abc123'
    this.users$ = this.mockService.getUserByOrgId(orgId);
  }
  mockService = inject(MockService);
  users$!: Observable<string[]>;
}
```

```
● ● ●

1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {
6     path: 'user/:orgId',
7     loadComponent: () => import('./user/user.component'),
8   },
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes, { bindToComponentInputs: true })],
13   exports: [RouterModule],
14 })
15 export class AppRoutingModule {}
```

MFE

npm Search packages

@angular-architects/native-federation TS

16.2.0 • Public • Published a day ago

[Readme](#) [Code](#) Beta [9 Dependencies](#) [0 Dependents](#) [12 Versions](#)

Native Federation for Angular

Native Federation is a "browser-native" implementation of the successful mental model behind webpack Module Federation for building Micro Frontends (Plugins, etc.).

Features 🔥

- Mental Model of Module Federation
- Future Proof: Independent of build tools like webpack
- Embraces Import Maps - an emerging web standard
- Easy to configure: We use the same API and Schematics as for our Module Federation plugin
- Blazing Fast: The reference implementation not only uses the fast esbuild; it also caches already built shared dependencies.

Prerequisite

Angular & Angular CLI 16.1 or higher

This package was successfully tested with Angular CLI projects and with Nx projects.

Versions

We will at least provide a new version of this package per Angular major. If necessary, we will also provide packages to adapt to Angular minors. To make the relationship between Angular

Install

```
> npm i @angular-architects/native-feder...
```

Repository

Homepage

Weekly Downloads

94

Version License

16.2.0 MIT

Unpacked Size Total Files

99.5 kB 73

Issues Pull Requests

138 25

Last publish

a day ago

RFC / Upcoming



Signal based input

```
● ● ●  
1  @Component({  
2    signals: true,  
3    selector: "user-profile",  
4    template: `  
5      <p>Name: {{ firstName() }} {{ lastName() }}</p>  
6      <p>Account suspended: {{ suspended() }}</p>  
7    `,  
8  })  
9  export class UserProfile {  
10   // Create an optional input without an initial value.  
11   firstName = input<string>(); // Signal<string|undefined>  
12  
13   // Create an input with a default value  
14   lastName = input("Smith"); // Signal<string>  
15  
16   // Create an input with options.  
17   suspended = input<boolean>(false, {  
18     alias: "disabled",  
19   }); // Signal<boolean>  
20 }  
21
```

Model inputs

```
● ● ●  
1  @Component({  
2    signals: true,  
3    selector: 'some-checkbox',  
4    template: `  
5      <p>Checked: {{ checked() }}</p>  
6      <button (click)="toggle()">Toggle</button>  
7    `,  
8  })  
9  export class SomeCheckbox {  
10   // Create a model, which is a *writable* signal that supports two-way binding.  
11   checked = model(false);  
12  
13   toggle() {  
14     // Normal inputs are read-only, but a model is writable  
15     // and propagates the value back to the parent.  
16     checked.update(c => !c);  
17   }  
18 }  
19  
20 @Component({  
21   signals: true,  
22   selector: 'some-page',  
23   template: `  
24     <!-- Note that the getter is *not* called here, the raw signal is passed -->  
25     <some-checkbox [(checked)]="isAdmin" />  
26   `,  
27 })  
28 export class SomePage {  
29   isAdmin = signal(false);  
30 }
```

Signal-based queries

```
● ● ●  
1  @Component({  
2    signals: true,  
3    selector: 'form-field',  
4    template: `  
5      <field-icon *ngFor="let icon of icons()"> {{ icon }} </field-icon>  
6      <div class="focus-outline">  
7        <input #field>  
8      </div>  
9    `  
10 })  
11 export class FormField {  
12   icons = viewChildren(FieldIcon); // Signal<FieldIcon[]>  
13   input = viewChild<ElementRef>('field'); // Signal<ElementRef>  
14  
15   someEventHandler() {  
16     this.input().nativeElement.focus();  
17   }  
18 }
```

Output in signal- based components

```
● ● ●  
1 @Component({  
2   signals: true,  
3   selector: 'simple-counter',  
4   template: `  
5     <button (click)="save()">Save count</button>  
6     <button (click)="reset()">Reset count</button>  
7   `,  
8 })  
9 export class SimpleCounter {  
10   saved = output<number>(); // EventEmitter<number>  
11   cleared = output<number>({alias: 'reset'});  
12  
13   save() {  
14     this.saved.emit(123);  
15   }  
16  
17   reset() {  
18     this.cleared.emit(456);  
19   }  
20 }
```

Lifecycle - 16.2 -preview

- **afterNextRender**- schedules a function to execute after the next rendering operation (that is, change detection cycle) is complete. This is useful whenever you want to read or write from the DOM manually.
- **afterRender** -schedules a function to be executed after any time the framework performs DOM updates during rendering.
- **afterRenderEffect** - is a special kind of effect which, when triggered, executes with afterRender timing.

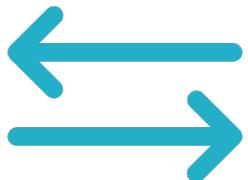
```
1  @Component({
2    template: `
3      <p #p>{{ longText() }}</p>
4    `,
5  })
6  export class AfterRenderCmp {
7    constructor() {
8      afterNextRender(() => {
9        console.log('text height: ' + p().nativeElement.scrollHeight);
10       })
11     }
12   }
13   p = viewQuery('p');
14 }
```

Component lifecycle and effects

Signal-based components will retain the following lifecycle methods:

ngOnInit

ngOnDestroy



The remaining lifecycle methods will not be available in signal components. For all of these methods, signals provide new patterns to achieve the same use cases:

ngOnChanges - is used to observe changes to inputs. As inputs are signal-based, computed can be used to derive new values, or effect to react side-effectfully.

ngDoCheck - typically this hook was used to implement custom change detection. effect is a likely (and more performant) replacement.

ngAfterViewInit is often used to perform some action after initial rendering. afterNextRender can be used instead (and is more correct).

ngAfterContentInit, ngAfterViewChecked and ngAfterContentChecked are often used to observe query results. Since queries are also signal-based and therefore reactive by default, signals can be used directly.



SIGNAL-BASED COMPONENTS
COMPLETELY DROP
THE CHANGEDETECTION SETTING.



ZONE LESS



REACTIVITY



```
1 <header-cmp />
2 @if showBody {
3   <body-cmp />
4 } @else if showSummary {
5   <summary-cmp />
6 } @else {
7   Nothing to see here...
8 }
9 <footer-cmp />
```



```
1 @switch kind {
2   @case 'human' {
3     <human-cmp />
4   }
5   @case 'robot' {
6     <robot-cmp />
7   }
8   @default {
9     <alien-cmp />
10  }
11 }
```



```
1 <ul>
2   @for item of item; track item.id {
3     <li>{{ item.name }}</li>
4   } @empty {
5     <li>No items...</li>
6   }
7 </ul>
```



```
1 @defer on viewport {
2   @main {
3     <!-- this block will be deferred until the placeholder is visible -->
4     <heavy-cmp />
5   }
6   @placeholder {
7     
8   }
9   @loading {
10    <loading-spinner />
11  }
12 }
```

RFC

Control flow

NGRX

```
● ● ●  
1 import { signalState } from '@ngrx/signals';  
2  
3 const state = signalState({  
4   user: {  
5     firstName: 'John',  
6     lastName: 'Smith',  
7   },  
8   foo: 'bar',  
9   numbers: [1, 2, 3],  
10});  
11  
12 console.log(state()); // { user: { firstName: 'John', lastName:  
13 console.log(state.user()); // { firstName: 'John', lastName: 'Sm  
14 console.log(state.user.firstName()); // 'John'
```

```
● ● ●  
1 // passing a partial state object  
2 state.$update({ foo: 'baz', numbers: [10, 20, 30] });  
3  
4 // passing an updater function  
5 state.$update((state) => ({  
6   user: { ...state.user, firstName: 'Peter' },  
7   foo: 'bar',  
8 }));  
9  
10 // passing a sequence of partial state objects and/or updater functions  
11 state.$update(  
12   (state) => ({  
13     numbers: [...state.numbers, 4],  
14     user: { ...state.user, lastName: 'Ryan' }  
15   }),  
16   { foo: 'bar' },  
17 );
```

signalStore

```
● ● ●  
1 import { signalStore, withState, withMethods } from '@ngrx/signals';
2 import { withCallState, setLoaded } from './call-state.feature';
3
4 type UsersState = { users: User[] };
5
6 const UsersStore = signalStore(
7   { providedIn: 'root' },
8   withState<UsersState>({ users: [] }),
9   withCallState(),
10  withMethods((store, userService = inject(UserService)) => ({
11    async loadUsers() {
12      // updating the state:
13      store.$update({ callState: 'loading' });
14      const users = await userService.getAll();
15      // or we can use reusable updater:
16      store.$update({ users }, setLoaded());
17    }
18  }))
19);
20
21 @Component({ /* ... */})
22 export class UsersComponent implements OnInit {
23   readonly usersStore = inject(UsersStore);
24   // available signals:
25   // - usersStore.users: Signal<User[]>
26   // - usersStore.callState: Signal<CallState>
27   // - usersStore.loading: Signal<boolean>
28   // - usersStore.loaded: Signal<boolean>
29   // - usersStore.error: Signal<string | null>
30
31   ngOnInit(): void {
32     this.usersStore.loadUsers();
33   }
34 }
```

signalStore



```
1 import { signalStore, withMethods } from '@ngrx/signals';
2 import { withEntities, setAll, deleteOne } from '@ngrx/signals/entities';
3 import { rxMethod } from '@ngrx/signals/rxjs-interop';
4 import { withCallState, setLoading, setLoaded } from './call-state.feature';
5
6 const UsersStore = signalStore(
7   withEntities<User>(),
8   withCallState(),
9   withMethods((store, userService = inject(UserService)) => ({
10     loadUsers: rxMethod<void>(
11       pipe(
12         tap(() => store.$update(setLoading())),
13         exhaustMap(() => userService.getAll()),
14         tap((users) => store.$update(setAll(users), setLoaded()))
15       )
16     ),
17   }))
18 );
19
20 @Component({
21   template: `
22     <p>Users: {{ usersStore.entities() | json }}</p>
23     <p *ngIf="usersStore.loading()">Loading ...</p>
24     <button (click)="onDeleteOne()">Delete One</button>
25   `,
26   providers: [UsersStore],
27 })
28 export class UsersComponent implements OnInit {
29   readonly usersStore = inject(UsersStore);
30
31   ngOnInit(): void {
32     this.usersStore.loadUsers();
33   }
34
35   onDeleteOne(): void {
36     this.usersStore.$update(deleteOne(1));
37   }
38 }
```

Thanks

