**Angular v12**

**Latest Version:** 12.0 **Release Date**: 12-05-2021

1. **Angular v12: “Ivy Everywhere” concept**

Angular v12 has finally deprecated the View Engine. The community has been working over recent releases towards the goal of converging the Angular ecosystem on Ivy. They call this approach “Ivy Everywhere”.

* As the View Engine is deprecated, so it will be removed in a future major release.
* Current libraries are using View Engine and it will still work with Ivy apps (no work is required by developers), but library authors should start planning to transition to Ivy.

1. **For forms,**[**min and max validators**](https://github.com/angular/angular/pull/39063)**are being introduced.**

This commit adds the missing min and max validators.

BREAKING CHANGE:

Previously min and max attributes defined on the <input type="number">

were ignored by Forms module. Now presence of these attributes would

trigger min/max validation logic (in case formControl, formControlName

or ngModel directives are also present on a given input) and

corresponding form control status would reflect that.

Fix: <https://github.com/angular/angular/issues/16352>

1. **Remove unused methods from DomAdapter**

The DomAdapter is present in all Angular apps and its methods aren't tree shakeable. These changes remove the methods that either aren't being used anymore or were only used by our own tests. Note that these changes aren't breaking, because the adapter is an internal API.

The following methods were removed:

**getProperty** - only used within our own tests.

**log** - Guaranteed to be defined on console.

**logGroup** and **logGroupEnd** - Only used in one place. It was placed in the DomAdapter for built-in null checking.

**performanceNow** - Only used in one place that has to be invoked through the browser console.

**supportsCookies** - Unused.

**getCookie** - Unused.

**getLocation** and **getHistory** - Only used in one place which appears to have access to the DOM already, because it had direct accesses to window. Furthermore, even if this was being used in a non-browser context already, the DominoAdapter was set up to throw an error.

The following APIs were changed to be more compact:

**supportsDOMEvents** - Changed to a readonly property.

**remove** - No longer returns the removed node.

1. **Exporting of a list of**[**HTTP status codes**](https://github.com/angular/angular/pull/23548)**.**

In Angular v12, **@angular/common/http** now exports a list of http status codes.

Issue: <https://github.com/angular/angular/issues/23543>

New feature: <https://www.npmjs.com/package/http-status-codes>

# **Passing Context to HTTP Interceptors**

# Passing metadata to HTTP interceptors. No more dirty hacks, such as using an HTTP header to pass custom data that our interceptor needs.

One common use case is to notify the interceptor that a request is [cacheable](https://github.com/ngneat/cashew). Let’s see how we use it.

First, update to the latest version of Angular. Next, create a new interceptor:

**ng g interceptor cache**

@Injectable()

export class CacheInterceptor implements HttpInterceptor {

intercept(request: HttpRequest, next: HttpHandler): Observable<HttpEvent> {

return next.handle(request);

}

}

Next, we need to create a unique token using **HttpContextToken** and provide a default value for it. A token is used to manipulate and access values stored in **HttpContext**.

**HttpContext** stores arbitrary user-defined values and ensures type safety without actually knowing the types.

The context is mutable and is shared between cloned requests unless explicitly specified.

const CACHE\_IT = new HttpContextToken<boolean>(() => false);

export function cacheIt() {

return new HttpContext().set(CACHE\_IT, true);

}

@Injectable()

export class CacheInterceptor implements HttpInterceptor {

intercept(request: HttpRequest, next: HttpHandler): Observable<HttpEvent> {

if(request.context.get(CACHE\_IT)) {

return ...

}

return next.handle(request);

}

}

Finally, we can provide it in our **HTTP** request:

import { cacheIt } from './cache.interceptor';

@Injectable()

export class UsersService {

constructor(private http: HttpClient) {}

getUsers() {

return this.http.get('....', {

context: cacheIt()

})

}

}

# **Nullish Coalescing**

The nullish coalescing operator (??) has been helping developers write cleaner code in TypeScript classes for a while now. We’re thrilled to announce that you can bring the power of nullish coalescing to Angular templates in v12!

Now, in templates, developers can use the new syntax to simplify complex conditionals.

For example:

{{age !== null && age !== undefined ? age : calculateAge() }}

Becomes:

{{ age ?? calculateAge() }}

1. **Support for inline Sass in the styles field of the @Component decorator**

Both Angular CDK and Angular Material expose a new Sass API surface designed for consumption with the new @use syntax. When updating to Angular 12, an app will automatically switch to the new API by updating via ng update.

1. **Angular v12 supports TypeScript 4.2. Support for TypeScript 4.0 and TypeScript 4.1 has been dropped.**

## Support For IE11 Has Been Deprecated

An update will start including a new deprecation warning message in Angular v12 — and remove support for IE11 in Angular v13.

1. **Upgrades the repository to use Node 14. Additionally, removes support for Node 10 as it has**

**reached end of life.**

BREAKING CHANGE: Angular no longer maintains support for Node v10

**Angular v12** includes a lot of **Bug Fixes**, **New Features** and **Performance Improvements**:

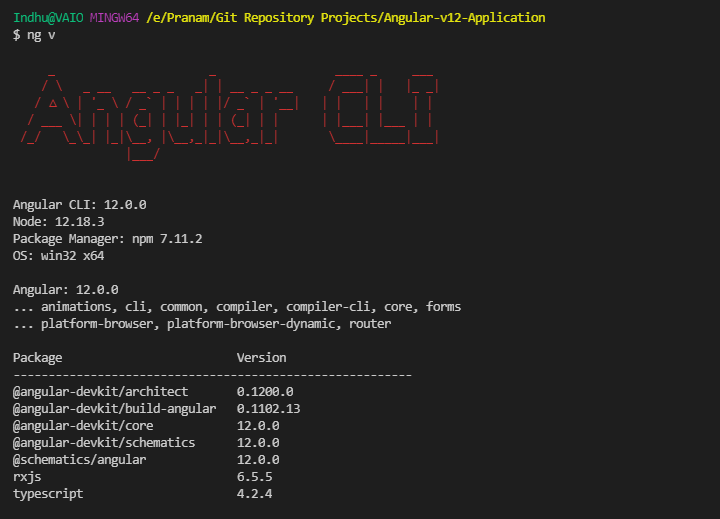
<https://github.com/angular/angular/releases>

**Steps to upgrade Angular 2/4/5/6/7/8/9/10/11 to Angular 12**:

* Make sure you are using **Node v12** (>Node v10) and **TypeScript 4.2**.

**Execute these commands:**

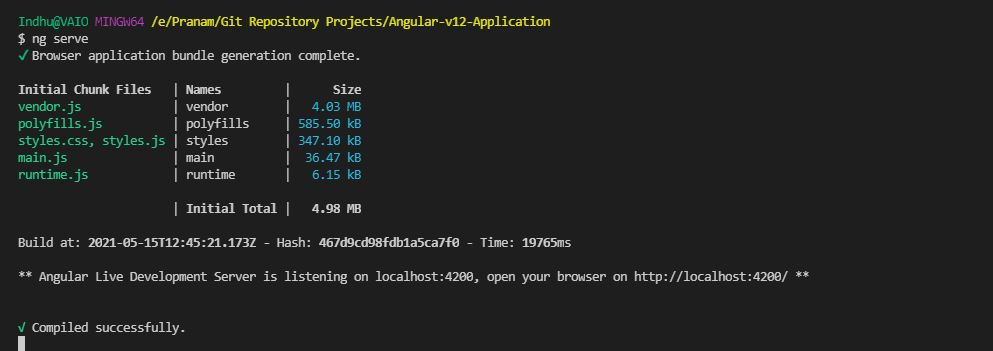
* ng update @angular/cli
* ng update @angular/core
* ng update --all –force
* npm install --force @angular/cli@12 @angular/core@12
* npm audit fix –force



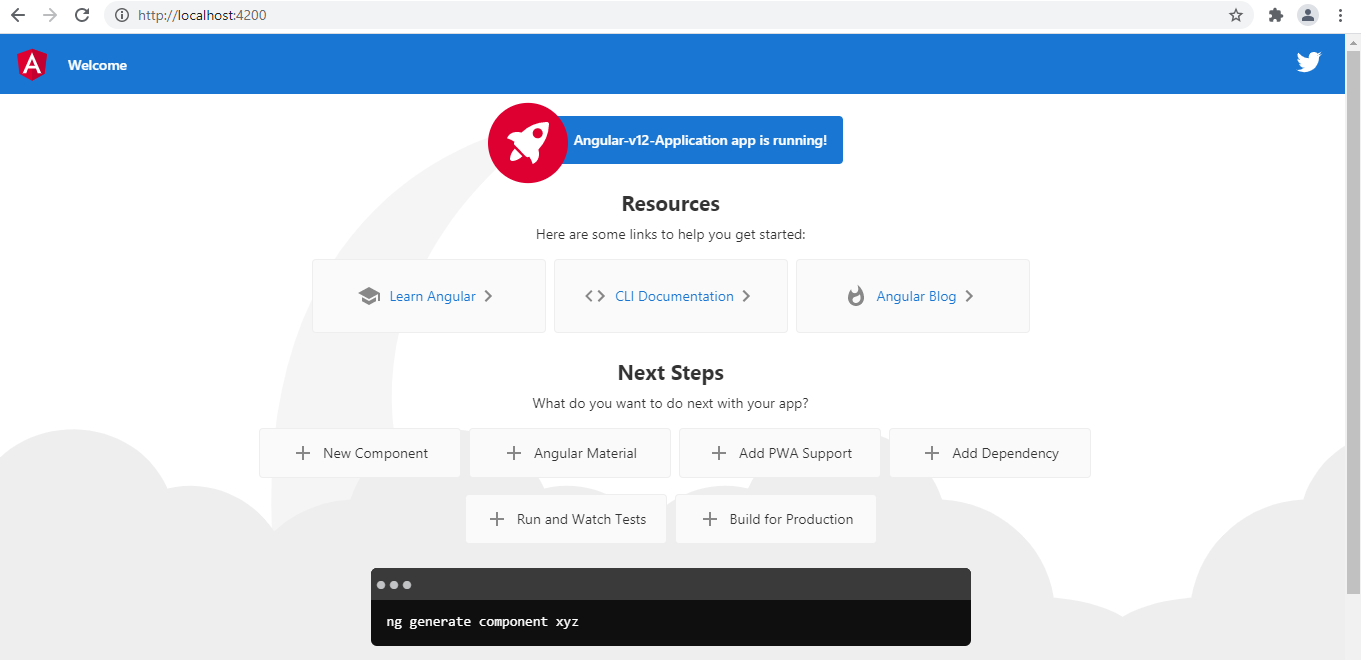
* Make sure you have to set “**aot**” to “**false**” in **angular.json** file. By default, it will be “**true**”.

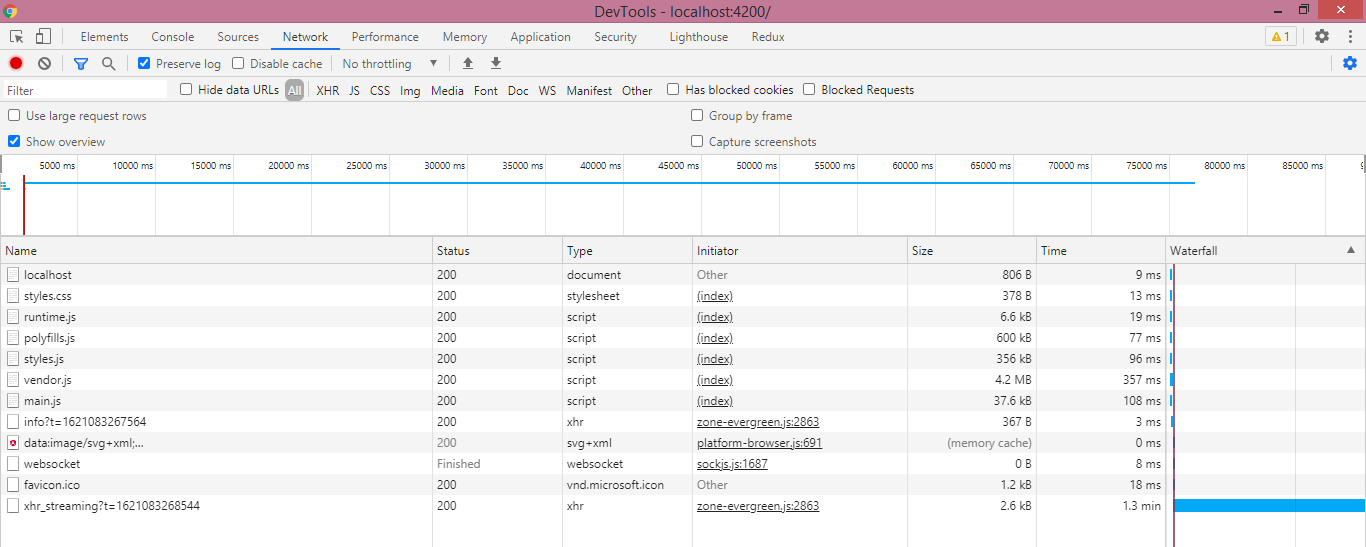
  "aot": false,

* ng serve



* Go to <http://localhost:4200/>





**Follow** me on **GitHub**: <https://github.com/PranamBhat>

**Connect** me on **LinkedIn**: <https://www.linkedin.com/in/pranam-bhat-11670689/>