

Sign in to your account (`de_@w_.net`) for your personalized experience.[Sign in with Google](#)Not you? [Sign in](#) or [create an account](#)

How Do You Speed Up Your Angular App?

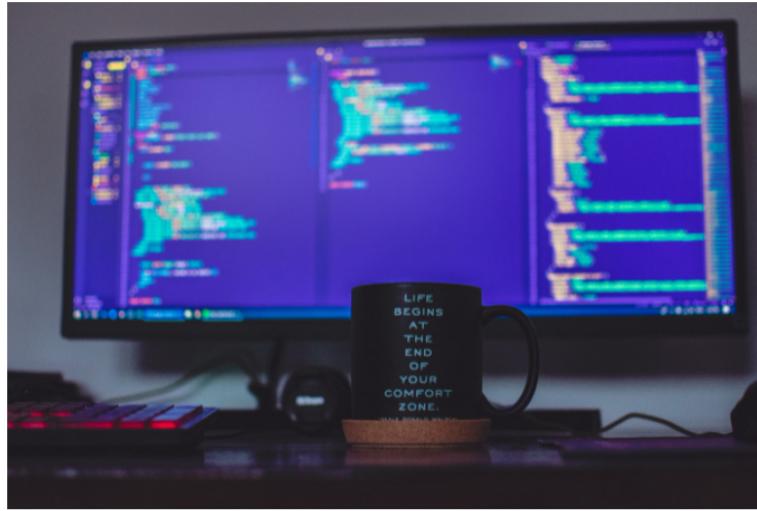
Here's how to fix Angular performance issues and optimize your Angular load time.



Daniel Kreider

[Follow](#)

Apr 27 · 7 min read

 Photo by [Tudor Baciu](#) on [Unsplash](#)

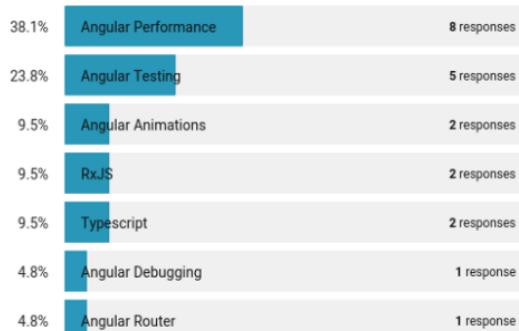
Angular performance is a common issue. It's probably Angular greatest weakness. How do I know?

I recently surveyed several dozen Angular developers, asking them their biggest frustration with Angular. Here's what they told me.

Response summary

1 What's your biggest frustration as an Angular developer?

21 out of 26 people answered this question





Not to mention an article I recently read that was written by an Angular consultant where he explained how he optimized an Angular application that took almost an entire minute to load. Yup! One. Whole. Minute.

Just think about it. You spend all this time building an Angular app, but as soon as it hits production, people start complaining that it's way too turtle-slow.



Now that's obviously an exaggerated example, but the Angular performance question is real.

Not to mention the frustrated Angular developers that don't know where to start when their boss demands a better performing Angular application.

How do you avoid this kind of situation? How do you optimize your Angular application? What can you do to improve the performance of your Angular application? And fix Angular performance issues?

Aside from [hiring an Angular consultant to help you](#), here's the in-depth-guide to improve and optimize your Angular performance and making sure it loads blazingly fast. Every. Single. Time.

What is loading performance versus runtime performance?

When we talk about fixing Angular performance and making our Angular app faster, there are two distinct areas to wrestle with and improve. There's a difference between making your Angular app **load fast** and making it **run fast**. This is known as the difference between the loading performance and runtime performance.

This guide focuses on how to make your Angular app **load fast**.

But if you need to optimize your runtime performance and make your Angular app **run fast** there are all kinds of Angular optimizations that you can consider like [using a trackBy function to improve ngFor performance](#).

Or [building a fast-performing Angular pipe](#).

As well as [tweaking Angular's change detection settings to avoid unnecessary change detection cycles](#).

And if you're suspicious that API calls could be choking your Angular runtime performance then here's [how to monitor the slowness of your HTTP calls](#).

Now that we understand the difference between loading performance and runtime performance, how do we improve it?

Getting started: Here's why you must profile Angular load time.

The first step in improving the performance of your Angular app is to measure how long it takes to load and bootstrap inside the browser.

Why?

Some developers know that they need to optimize their Angular app but they have no idea how much or where exactly to start. And so they make these wild stabs at the code or build configuration and then refresh the app to see if it appears to load faster than it did before. This kind of optimization approach is nothing but a clown show and deserves every kind of criticism it can get. It's like trying to bake a cake without the necessary measuring cups. Or re-arranging the chairs on the deck of the Titanic before it sank.

So what's the smarter approach?

Measure. Measure. Measure.

Somehow we as developers don't always notice the extra second or two that will bother the users. At least that's been my experience.

For example, one of my developer buddies is a Django developer. He kept insisting that his Django application was faster than my Angular app. When we were together, he'd pull out a browser and show me just how much faster it was and how sluggish my Angular app was until we started **actually counting the milliseconds** by using the browser's developer tool. After profiling the performance of my Angular app vs. his, we discovered that they're averaging about the same load time.

Obviously, my friend's internal timer wasn't properly counting and proves why a developer must carefully profile the performance of his Angular app and know exactly how many milliseconds it takes to load the thing. By counting the milliseconds that it takes to load your Angular app you will be able to know exactly how much you're improving load speed and performance as you attempt to optimize. Yup, **count the milliseconds**.

How to decrease the Angular bundle size?

There are all kinds of tweaks and wildly optimizing tree-shaking stuff floating around on the web about how to decrease the size of your Angular compiled app.

Some of the suggested performance tweaks are impressive, like [this one about using the Google Closure Compiler to decrease the size of your Angular app by 2.5x](#), and I'm glad for the effort that others have made to shave a few KB's. But I find that a lot of the suggestions are like kicking a dead donkey — especially if your Angular app has been poorly engineered.

So how do you design your Angular app to perform faster?

Unless I miss my guess, for every 8 out of 10 Angular projects the quickest and easiest way to shave the size of your Angular app and make it load faster is by splitting your main `AppModule` into smaller modules and lazy loading the other modules when you choose to.

In other words, make sure the size of your `AppModule` is as small as possible. Modularize everything you don't need right away and then lazy load them later because lazy loading helps keep the initial size of your Angular app smaller and makes for a speedier Angular load performance.

Speedy hosting options for your Angular app?

And after you've got your Angular app properly split into lazy-loaded modules, what's the next best step?

Deploy to a fast, global CDN.

And what are the fastest Angular hosting options? Where can you host your Angular app to guarantee the fastest load speed?

Here's a list of 3 different hosting options that specifically support Angular. This isn't a comprehensive list but each one of these Angular hosting options

- Has a free tier that allows you to get started and grow.
- Uses a global CDN that your Angular app is deployed too.
- Is a hosting solution that I've had success with. In other words, I know what I'm recommending.



The screenshot shows the Roast.io landing page for Angular hosting. The title is "Angular Hosting". It features a large "A" logo. Below the title, it says "Roast.io is a CDN static web host for Angular sites/apps. We focus on speed, simplicity, and sensible defaults so you can run `roast deploy` and never think about hosting again." A bulleted list of benefits follows, including automatic HTTPS, better SEO, social media unfurls, faster requests, and reduced bandwidth. At the bottom, there are instructions for installing the CLI tool with npm and deploying the build/dist directory, mentioning a `.redirects` file for rewriting 404s to /index.html.

If you want to start roasting an Angular application on Roast.io then here's the [getting started guide for Angular applications](#).

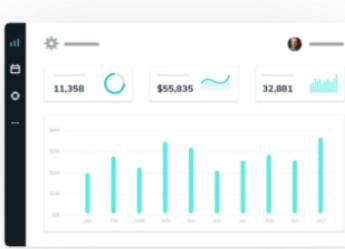
Netlify

Modern web apps shipped faster

An intuitive Git-based workflow and powerful serverless platform to build, deploy, and collaborate on web apps

[Get started for free](#)

Questions? [Talk to an expert](#)



Want to deploy your Angular app on Netlify? [Here's how to get started.](#)

Firebase Hosting

Firebase Hosting is a hosting service that was created by Google. If you want to know more than [here's how to deploy an Angular app to Firebase Hosting in 2021.](#)

Rolling our own Nginx server with Brotli support

But what if you have your own server with a VPS like Digital Ocean? And would rather not deploy to a CDN?

Then Brotli is here to help you.

[Brotli](#) is a compression algorithm developed by Google and considered the successor to [gzip](#). I've seen it cut an entire 1000ms and more off of the load time for Angular apps.

If you're using a pre-backed hosting solution like Firebase Hosting, Microsoft Azure Hosting, or another option that I mentioned above then you're outta luck. But if you have your own server that you can tweak and customize then you can compile Nginx from source with Brotli support.

Why compile it from the source?

Because Nginx doesn't come built with Brotli support. You have to use Google's Nginx Brotli module.

But the good news is I have a script that will do the job for you.

Here it is.

```
#Install deps
sudo apt install unzip zlib1g-dev libpcre3-dev libssl-dev brotli -y

# Pull nginx
mkdir /tmp/nginx-brotli
cd /tmp/nginx-brotli
wget http://nginx.org/download/nginx-1.18.0.tar.gz
tar -xvf nginx-1.18.0.tar.gz

# Pull brotli module
git clone https://github.com/google/ngx_brotli.git
cd ngx_brotli
git submodule update --init

# Build nginx
cd ../nginx-1.18.0
./configure --add-module=../ngx_brotli

./configure --sbin-path=/usr/local/nginx/nginx --conf-path=/usr/local/nginx/nginx.conf --pid-path=/usr/local/nginx/nginx.pid --with-http_ssl_module --with-brotli --with-pcre --with-pcre-jit
```

```
http_v2_module --add-module=.../ngx_protocols  
make  
sudo make install
```

Conclusion

Angular is a fast-evolving framework — which is one of the reasons I enjoy Angular — and it's built by a great group of people that want to know the frustrations Angular developers have. The Angular team has already shed a lot of sweat to improve the size of Angular applications and Angular's performance and I expect that they'll continue to do so.

If you enjoyed this article and found it useful please feel free to hammer that clap button and follow me for more cool articles like this one.

Follow Me: [GitHub](#), [Medium](#), [Personal Blog](#)

Look at you go!

Your download is hurtling its way to your inbox. See ya there...



7 Money Murdering Mistakes That Only Angular Developers Make

Grab your FREE copy, will ya?

SEND IT TO ME

I agree to leave javascript.plainenglish.io and submit this information, which will be collected and used accordi. [Powered by Unsubscribe policy.](#)

A handwritten signature in black ink that reads "Daniel Kreider".

• • •

Originally published at <https://danielk.tech>.

[More content at plainenglish.io](#)

Sign up for Last Week in Plain English

By JavaScript in Plain English

Updates from the world of programming, and In Plain English. Always written by our Founder, Sunil Sandhu. [Take a look.](#)

Get this newsletter



WRITTEN BY
Daniel Kreider
<https://danielk.tech>

[Follow](#)

JavaScript in Plain English

[Follow](#)

New JavaScript and Web Development content every day.
Follow to join our +2M monthly readers.

More From Medium

Installing AngularJS!
Sakshi Saxena



Making an Awesome
Pagination Hook with
React Query
Andrew Eames



Testing with Dojo
Rene Rubalcava



Passport-Local
Authentication
James Kaviyil Jose



4 Easy Steps To Building
Your First Twitter Bot
With JavaScript
Dennis O'Keeffe



Vuetify—Text Styles
John Au-Yeung



JavaScript Best Practices
—Arrow Functions and
Constructors
John Au-Yeung in Level Up
Coding



Do You Like React? Then
Why Don't You Configure
Your Infrastructure With
It?
Frank Zickert | Quantum
Machine Learning in DailyJS



Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox.
[Explore](#)

Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. [Start a blog](#)