Lets load are app faster

Hi all.

I am a front-end developer , while building one of our (great 😊 ) Apps I got a task: “Make our loading performance better in our app”. With the help of my colleagues we did some good work (in my eyes). I will try to summarize some ideas that can my help you also to decrease your app loading time.

I will write here all of the areas that need to be fixed and there is a GitHub repo with the code. The “starting point” will be the code that is in the “master/main” branch, and the “better code” will be in different branches.

The goal of this article is to list our actions to load our app faster.

This will be divided to several areas: how to analyze and detected the problems, and how to try to fix them.

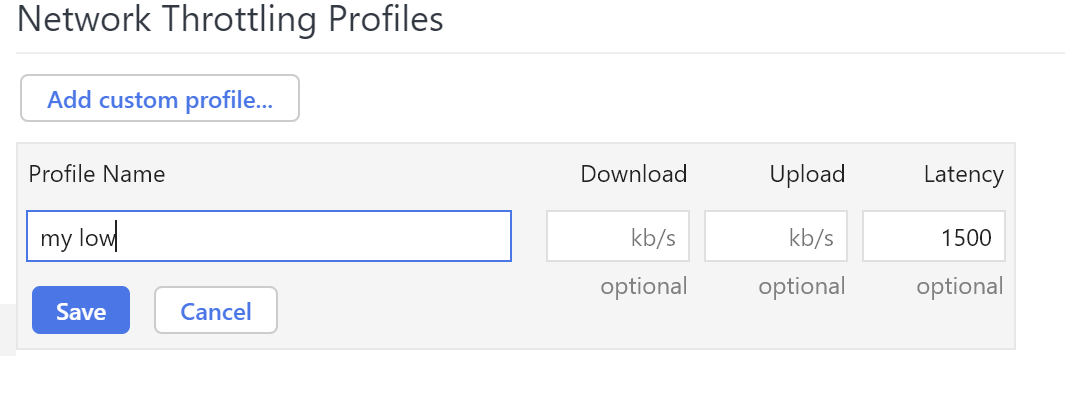
Analyzing:

To analyze and detect the “problems” that we will want to fix for getting a faster loading app some actions are determand:

* Use chrome developers tools: the light house and record the loading with the performance tab
* Run a “ng build” with the webpack analyzer

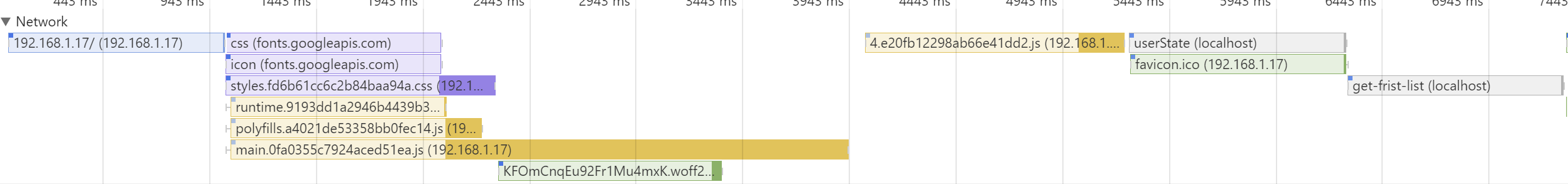
Get details from chrome tools

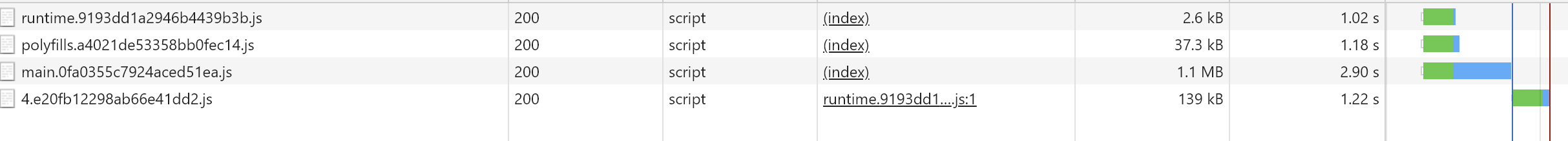
Before running the tools I demonstrated a server that its latency is low (can be for any reason). You must have the best severs you can get but sometimes this is not your team to decide and the teem can play with the “cards” that they get and make the best out of it (just like my team had).

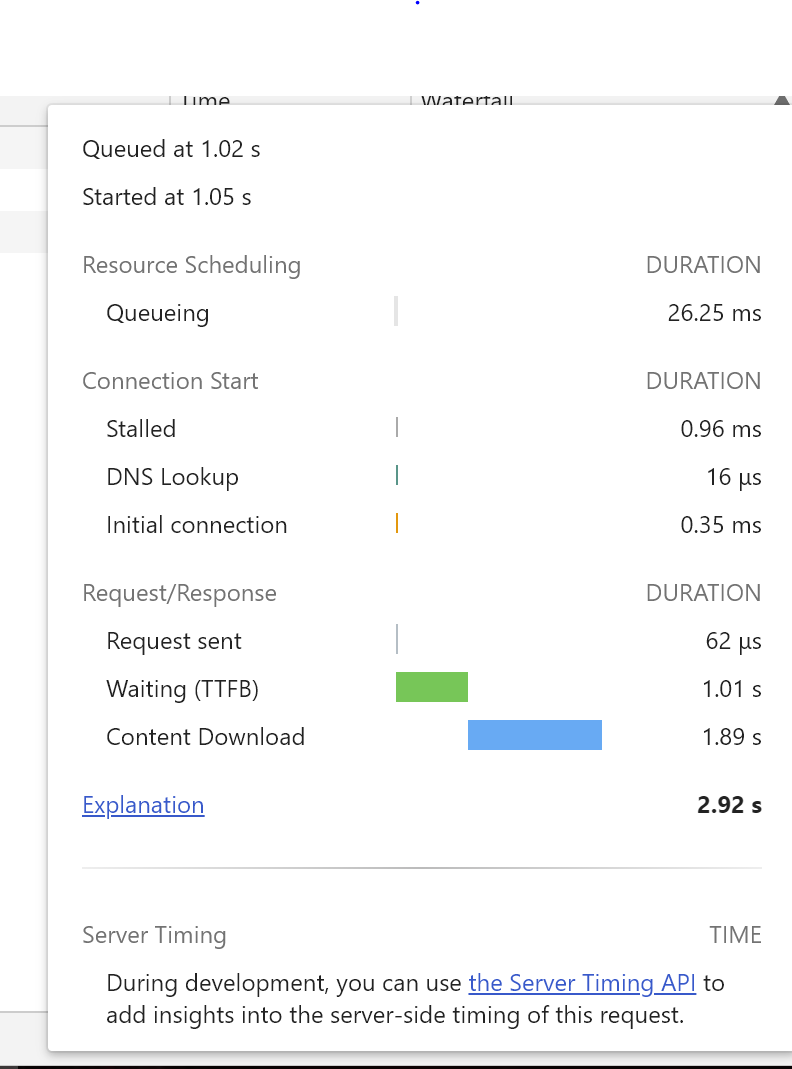


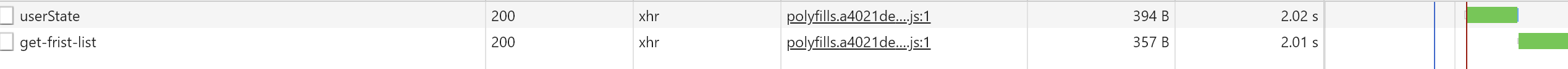
After defining the “server”:

* Start the node server in the project path: “src/server/run.js”.
* Run “ng build –prod”
* Install in your local machine the node package “http-server” ([link to package](https://www.npmjs.com/package/http-server))
* Open in chrome and “record” the loading. This is what I got









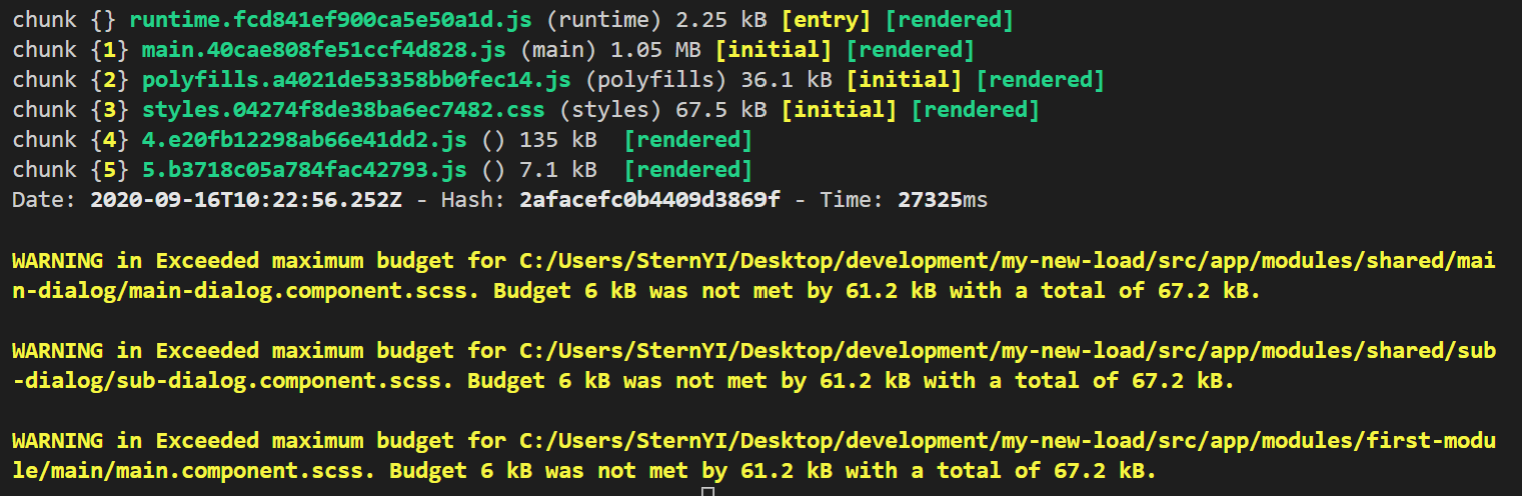
After looking in the images we can list some “red flags” that help us to focus in the bad areas in our code:

* The whole process takes 7 seconds which is a very long time for a simple process.
* The “main” chunk is very big and it takes 3 seconds to be downloaded.
* the app has a “buttle neck” with the path: loading main.js => loading module (4…..js) => login api (userState call) => list api call.
* Api calls take 2 seconds.

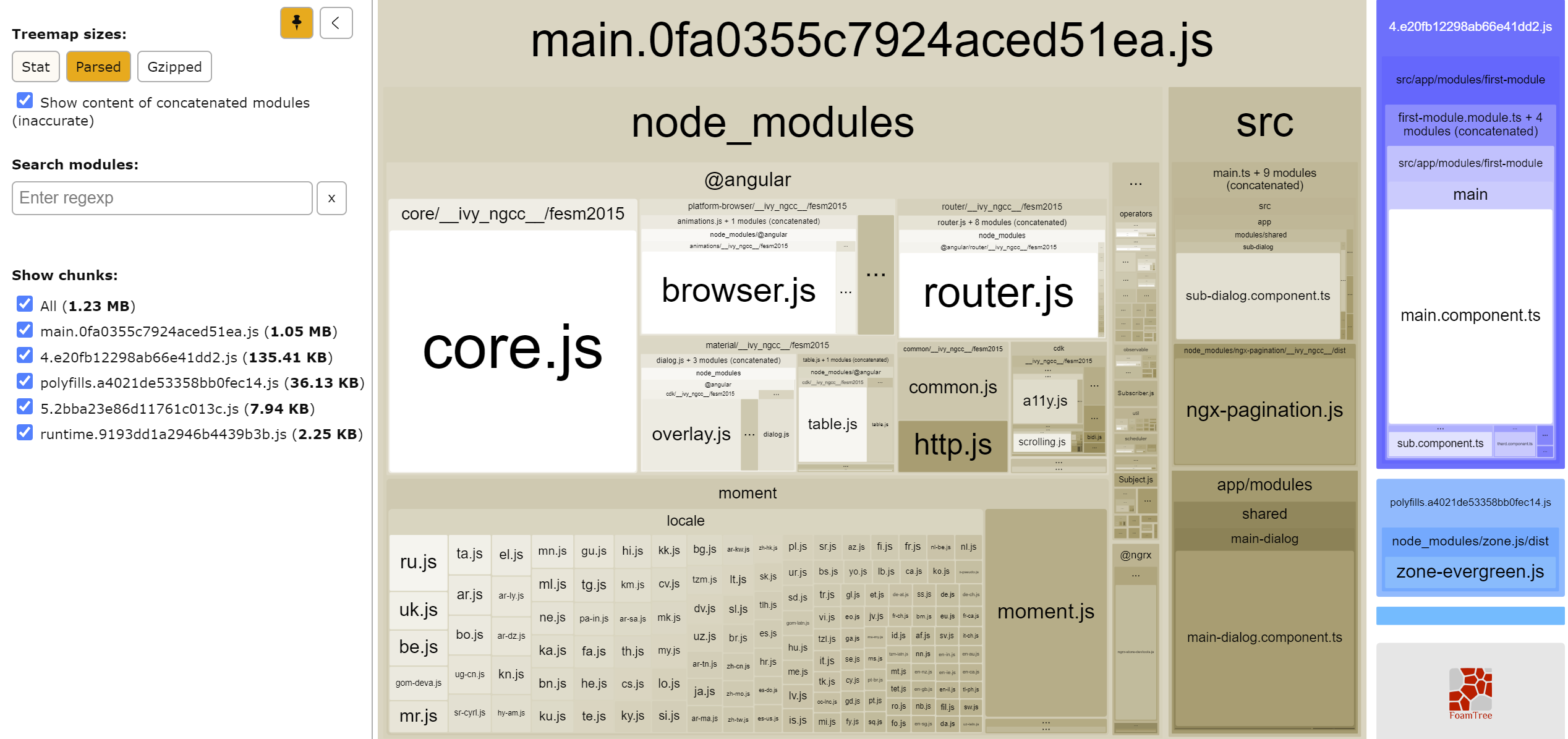
Analyze our bundles

After the recording we know that we have big bundules and to undesrstand there structure we will use a packahe called : “webpack-bundle-analyzer”.

To run this package look at its [ducomentaion,](https://www.npmjs.com/package/webpack-bundle-analyzer) after rhat run the “ng build --prod --stats-json” and this is what we get: first the output

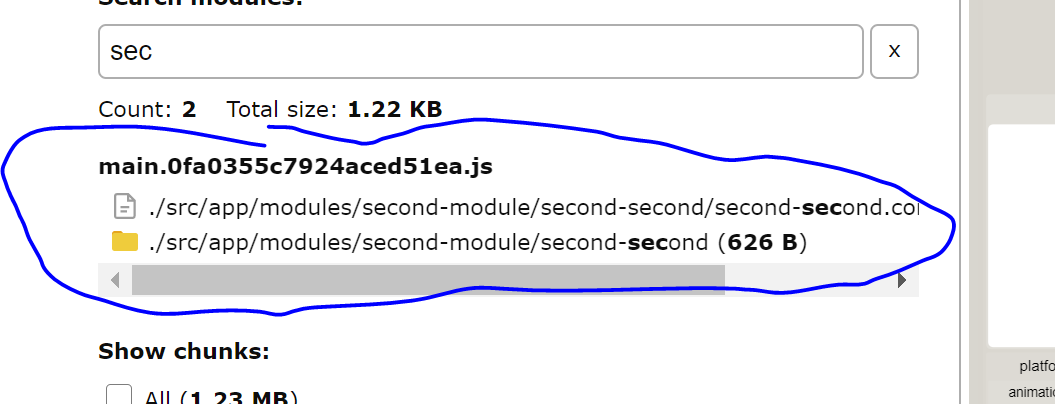


Than run the “npm run bundle-report” and this is what we get



After analyzing the chunks there are some questions:

* Why is main.js so big
* Do we need al the theard party packges (moment, pagining, etc…)
* Why components are so big
* Why components that are defined in a module is defined in the main chunk



In the next articles we will do the next actions:

1. Minimize bundles as much as possible
2. Optimize API calls