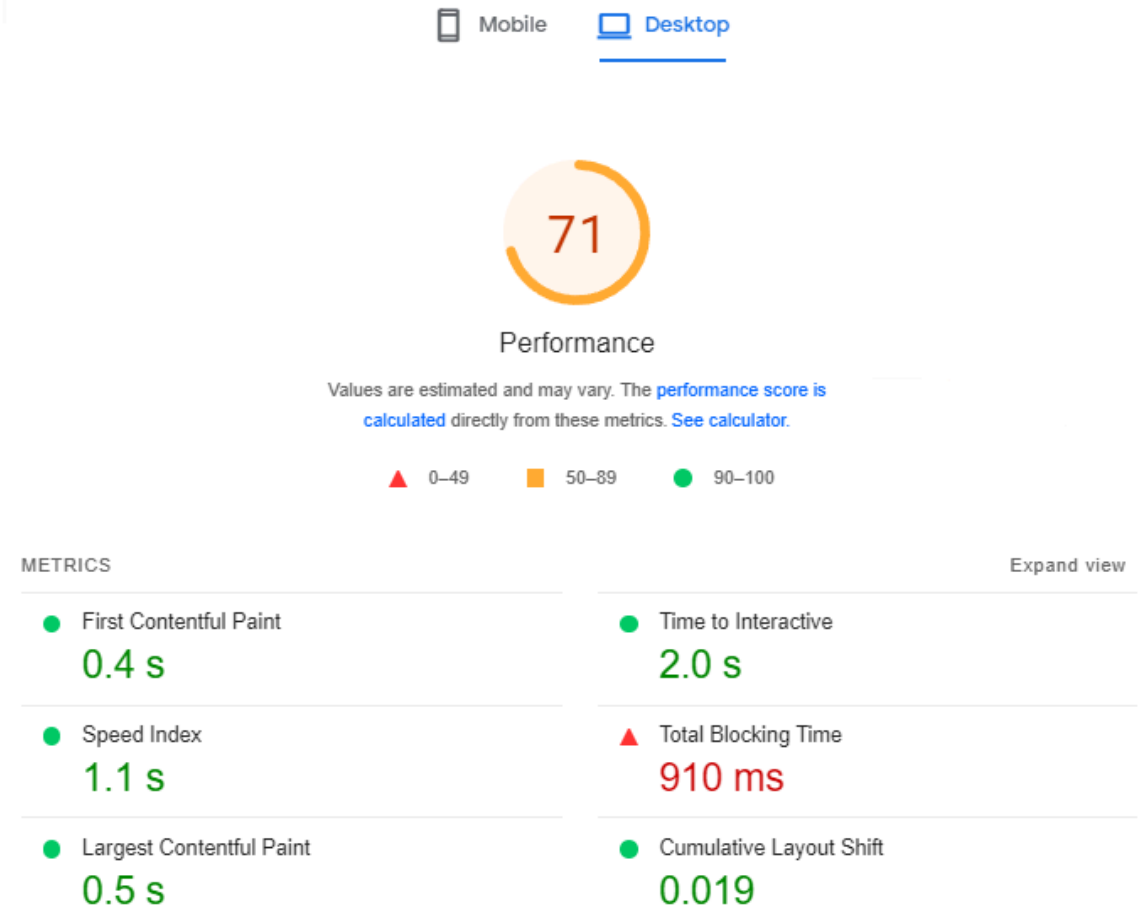
NextJs - Improve "Total Blocking Time" on Google Pagespeed

Asked 2 years, 5 months ago Modified 2 years ago Viewed 17k times

4

I'm using NextJS and i'm pretty new on that. I'm trying to increase my ranking on Google Pagespeed, and i already did some good progress on that.
As you guys can see on the screenshot, the only bad metric is the "Total Blocking Time":



If you guys want to try the page speed, thats the link: Google PageSpeed

Right now i'm running out of options on how to make that one better, i'm alredy dynamically importing my components, removed unused JS, i'm using the NextJs best practices.

I'll really appreciate any help that you guys could have

Thanks in advance

javascript reactjs next.js pagespeed

Share

Improve this question

Follow

edited Jun 28, 2022 at 12:08

asked Jun 27, 2022 at 14:44

GuiPab

475 ● 3 ● 7 ● 18

1 At the first look i see that you have some google maps javascript that you can load when it's need it. I don't see any google maps on visible part so may be load that when it's need it. You can read more about TBT here – angel.bonev Jun 28, 2022 at 12:26

I'm using the google maps api at the "Search By Location" input (which is a Google Place Autocomplete). But i'm already dynamically loading that component, so i think that's not suppose to be the problem – GuiPab Jun 28, 2022 at 12:49

3 It's . can you try loading it on focus an try again. It's loaded between `DOMcontent Loaded` and `Load`. Everithing that

can be loaded later must be loaded later. Everithing that is not on the visible part MUST be loaded later. May be when the element is visible or something. If you want to lower you TBT , everything that can be delayed must be delayed – angel.bonev Jun 28, 2022 at 13:02

1 Thats an interesting approach (just load the maps api when the user clicks on the input), I going to try implement that, and then i'll be back to tell the results. Thanks @angel.bonev – GuiPab Jun 28, 2022 at 13:35

2 may be use `IntersectionObserver` and load some css and javascript when they are need it ? I've posted an answer. In my case that was enough to hit 100%. But make sure you can't optimize you core more. This is not the holy grail. You need to provide best experience for your user, not to sutisfive google page speed test. Cheers and good luck – angel.bonev Jun 30, 2022 at 8:57

2 Answers

Sorted by: Highest score (default)

8

So lets start from what is TBT as the docs says

> The Total Blocking Time (TBT) metric measures the total amount of time between First Contentful Paint (FCP) and Time to Interactive (TTI) where the main thread was blocked for long enough to prevent input responsiveness.

How to improve your TBT score You need to start with a Lighthouse performance audit

So if you find some action that aren't necessary you need to delay it. Most common are :

> 1. Unnecessary JavaScript loading, parsing, or execution. While analyzing your code in the Performance panel you might discover that the main thread is doing work that isn't really necessary to load the page.
> 2. Inefficient JavaScript statements. For example, after analyzing your code in the Performance panel, suppose you see a call to document.querySelectorAll('a') that returns 2000 nodes. Refactoring your code to use a more specific selector that only returns 10 nodes should improve your TBT score.

For example you have some element in the footer or the hole footer it self. You can split your `javascript` and `css` and loaded that part dynamically. You can use IntersectionObserver , the support for that is good enough, but it's allways a good practice to ensure if not supported:

Example code (*in vanilla javascript*):

```
let IOObjects = [
    {"idToObserve": "myFooter", functionToTrigger: "so
];
let someFinction = () =>{
    //load some events, another js or another css
}
```

```
IOObjects.forEach(function (e) {
    if (document.getElementById(e.idToObserve)) {
        if (!window.IntersectionObserver) {
            window[e.functionToTrigger]();//Trigger th
supported
        } else {
            let observer = new IntersectionObserver(fu
                entries.forEach(function (entry) {
                    if (entry.intersectionRatio > 0) {
                        observer.unobserve(entry.targe
                        window[e.functionToTrigger]();
                    }
                });
            }, {rootMargin: '50px 0px', threshold: 0.0
            observer.observe(document.getElementById(e
the element if it's visible
        }
    }
});
```

You can read more about rootMargin

Share Improve this answer Follow

edited Jun 30, 2022 at 9:22

answered Jun 30, 2022 at 8:53

angel.bonev

2,232 ● 4 ● 23 ● 32

0

Actually, dynamically render everything is an bad practice, hence more bad performance. You will want to dynamic import component that doesnt include in the first build time, for example, a footer or chatbox.

Share Improve this answer

answered Nov 28, 2022 at 10:47

Follow

An2ta

19 ● 1