How to solve Next.js window is not defined

Written by Seth Corker on Benevolent Bytes

React is a client side library which means you can access everything you're familiar with on the web including `document` and `window`. If you're coming to Next.js from Create React App or using React as a single page app, there are some pitfalls to beware of.

Next.js is capable of rendering React on the client, just as before, and rendering React on the server. Even if you don't explicitly use `getServerSideProps`, Next.js will still pre-render the page using the node server. You can learn more about this pre-rendering in the Next.js docs. In this environment, we don't have access to `window`. So, what happens?

What happens when accessing `window` on the server

When Next.js pre-renders the page, it generates the HTML then sends that to the client. When a user visits the page, it will load the HTML then rehydrate (the process of running React to allow the page to become interactive). This is where the issue can occur. On the server, `window` is `undefined` and accessing anything on `window` will result in the familiar error thrown from `ReactDOMServerRenderer.render`:

```
ReferenceError: window is not defined
```

Now we know what can cause this error, why does it only show up sporadically? ## When does this error occur? The challenge with this error – it's inconsistent because Next.js may render your app differently depending on how the user navigates to a given page.

Take a look at the CodeSandbox to see the issue in action. In the example, we'll use a the Geolocation API but any other browser specific APIs will present the same problem.

Here are a few scenarios:

Using the next/link component

1. The user clicks on an internal link which uses the `next/link` component
2. The page doesn't error and loads just fine

What happened? Because the navigation took place via a `next/link` component, the routing happens client-side, so Next.js didn't pretender the page.

Navigation from an external page

1. The user is on another site and click a link to the page
2. The page shows an error

The error occurred because the request went to the server directly and Next.js decided to pre-render the page when `window` doesn't exist.

Page Refresh

1. The user clicks on an internal link which uses the `next/link` component
2. The page doesn't error and loads just fine
3. The user refreshes the page
4. The page shows an error

The first navigation happens on the client, but the refresh makes a request to the server directly, triggering the error.

We know when it happens and why, let's see how we can mitigate the issue.

How do you fix window is not defined?

One way is to not use `window` at all when rendering the page, but that isn't practical. Let's look at some actual solutions that allow us to use `window` on the client but ignore it on the server.

useEffect to the rescue

An easy solution to resolve this issue is to rely on the `useEffect`, conveniently hooks aren't run when doing server-side rendering. Wrapping the usage of `window` inside a `useEffect` that is triggered on mount means the server will never execute it and the client will execute it after hydration.

```
useEffect(() => {
  window.navigator.geolocation.getCurrentPosition(
      (newPos) => setPosition(newPos),
      console.error
    );
}, []);
```

Checking for `window`

There is another option that is useful outside of React too. Checking if `window` exists in the current context:

```
if (typeof window === "undefined") { /* we're on the server */ }
```

We can safely put code that is only supposed to be executed on the server behind this guard, likewise we can use `typeof window !== "undefined"` for code that should only run on the client.

Interact with the CodeSandbox

Conclusion

`ReferenceError: window is not defined` is a pretty common error you may run into when using Next.js for the first time but don't let that be a blocker. If you keep in mind that your code could be run on the client and the server, you'll get used to ensuring your code works for both. I hope these solutions will help you on your journey.