**Zustand with Next.js (app router) - Server and Client Components**

[[](https://medium.com/@mak-dev?source=post_page-----da9c191b73df--------------------------------)](https://medium.com/@mak-dev?source=post_page-----da9c191b73df--------------------------------)

[Mohammadali Karimi](https://medium.com/@mak-dev?source=post_page-----da9c191b73df--------------------------------)

·

Follow

3 min read

·

6 days ago

18

2



In this article, we try to implement Zustand with Next js app router to fetch data from server-side and store it with Zustand on client-side.

**What is Zustand?**

Zustand is a small, fast and scalable bearbones state management solution using simplified flux principles. Has a comfy API based on hooks, isn’t boilerplatey or opinionated.

*To understand how to install and use Zustand you can read the*[*Zustand document*](https://github.com/pmndrs/zustand)*.*

**What is the disadvantage of using state management systems on the server-side?**

Server-side components are meant to be stateless, and whether you use Zustand or other state managements as a hack to bring state into the server, it’s just going to maintain a global state that’s been shared between all users and they can see what the last user put in the “state” inside the server.

For example, assume your website has several pages behind authentication, try this: Put something that’s only accessible by authentication into a server-side Zustand store. Render it into a server-side component in a server-rendered page. Now try accessing this page via two different users. If you put something specific to the first user, and then load the page for the second user, you’ll see that the data from the first one has now leaked into the page shown for the second user.

**So, What is the solution?**

The workaround is to fetch data on the server-side page by adding "use server"to the top of the function body and passing the fetched data to other client-side components like the AppInitializer component to handle states with Zustand.

From this point on, Let’s get our hands dirty and do some coding.

**Implementations:**

Firstly, Create a folder in your /src path of the project called storeand then create a store to handle our states for example we createuseStore :

import { create } from 'zustand'  
  
export const useStore = create(() => ({  
 user: null  
}))

After creating the store we need a component to set our fetched data from server-side in the Zustand store on client-side, let's call it AppInitializer. Then we should pass fetched data on server-side as props to the initializer component.

'use client'  
  
import { useStore } from '@/store/useStore'  
  
export default function AppInitializer({ user, children }) {  
 useStore.setState({   
 user,  
 // ...  
 })  
  
 // ...  
 return children  
}

In the last Next js, server components can use the inline function level or module level "use server" directive. To inline a Server Action, add "use server" to the top of the function body, so in our page we follow like this:

export default async function Page() {  
 async function getUser() {  
 'use server'  
 const res = await fetch('https://api.example.com/...')  
 const data = await res.json()  
 }  
  
 const user = await getUser()  
   
 return (  
 <AppInitializer user={user}>  
 // ...   
 </AppInitializer>  
 )  
}

*Also you can move your fetching fucntions to the /actions/server path.*

By using this architecture, we can incorporate every Next js server-side feature (like cookie) and after processing on server-side then render the app in client-side in order to store data in our state management like Zustand.

Thank you for reading, I hope that helps!  
Also, you can find me on [**LinkedIn**](http://www.linkedin.com/in/mak-dev)**.**

[Nextjs](https://medium.com/tag/nextjs?source=post_page-----da9c191b73df---------------nextjs-----------------)

[Zustand](https://medium.com/tag/zustand?source=post_page-----da9c191b73df---------------zustand-----------------)

[React Hook](https://medium.com/tag/react-hook?source=post_page-----da9c191b73df---------------react_hook-----------------)

[State Management](https://medium.com/tag/state-management?source=post_page-----da9c191b73df---------------state_management-----------------)

[JavaScript](https://medium.com/tag/javascript?source=post_page-----da9c191b73df---------------javascript-----------------)