



# Setup **Environment Variables** in NextJS



Wasif Alam  
@wasif alam





## Why Environment Variables Matter?

➔ Consider environment variables as **placeholders** for confidential data in web development. They enable us to manage secret information, like passwords and tokens, separately from our code, ensuring a **secure** and flexible web application.

### We will be covering the following steps:

1. Installing latest Next app.
2. Set Up environment variable.
3. Access variables form server.
4. Access variables form server component.
5. Access variables form client component.





## Step #1: Installing Next App

Let's first install Next.js using the following command:

```
npx create-next-app@latest
```

Ensure that you use the following settings to achieve the same result as mine

```
→ Desktop npx create-next-app@latest
✓ What is your project named? ... redux-next
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
? What import alias would you like configured? > @/*
```



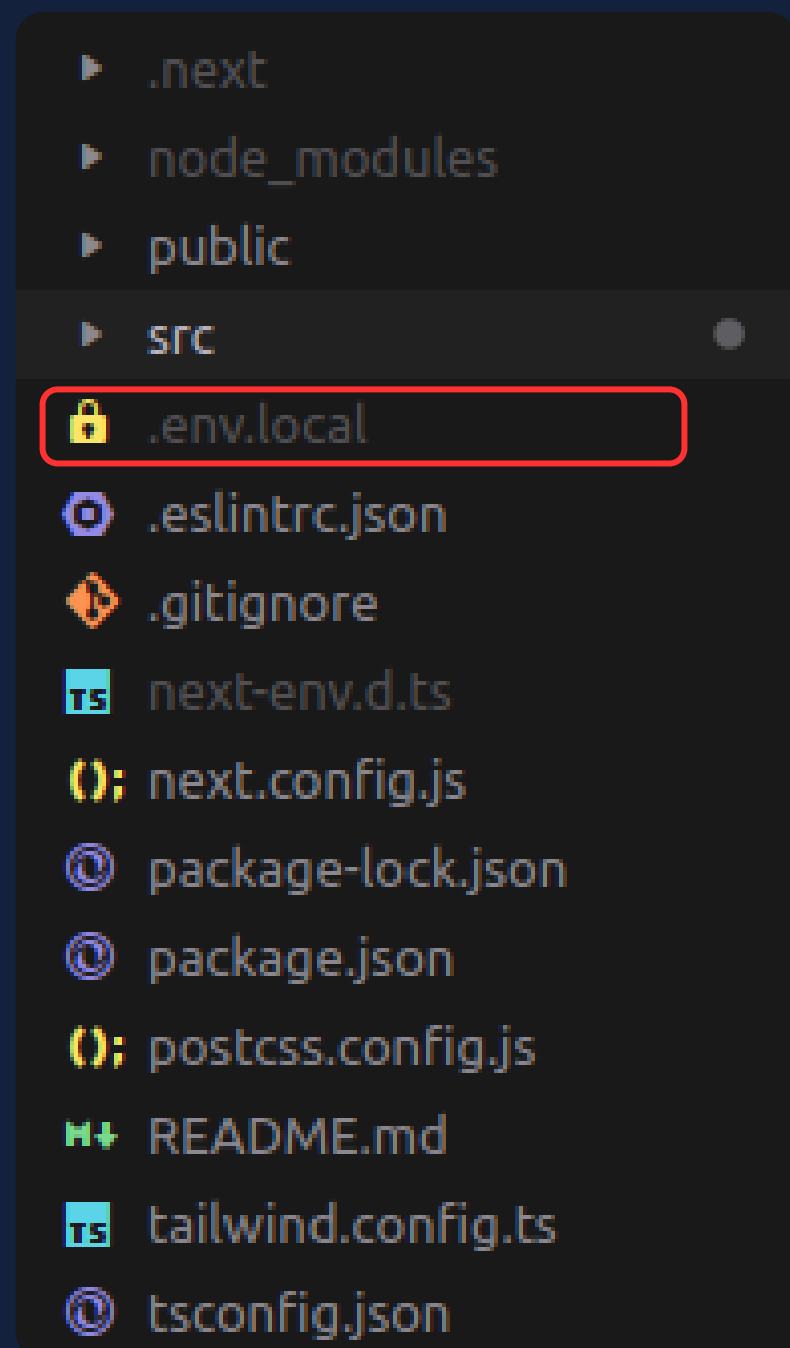
**Wasif Alam**  
@wasif alam





## Step #2: Setup Environment Variables

Let's add a file named `'.env.local'` at the root of our Next.js project:

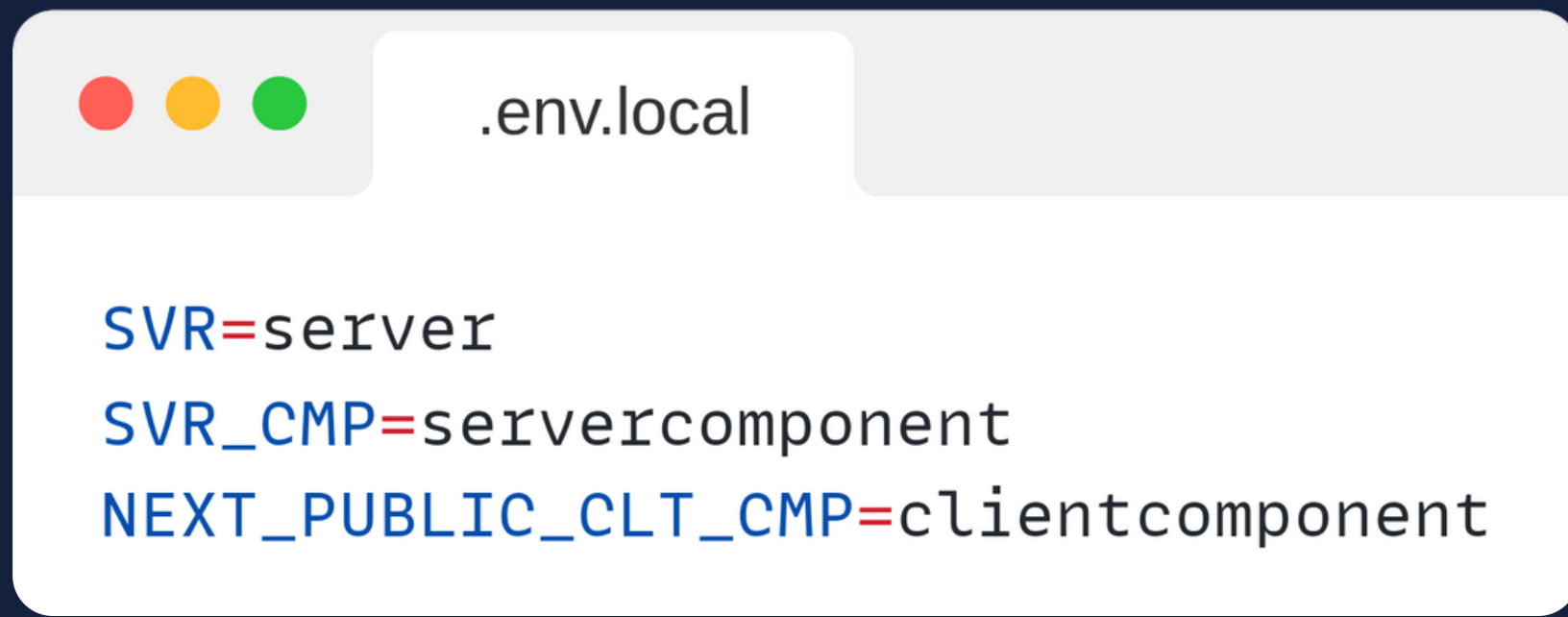


**Note:** Next.js has built-in support for loading environment variables from `.env.local`





Now, let's define the following variables to our recently created `'.env.local'` file:

A mockup of a code editor window with a title bar containing three colored circles (red, yellow, green) and a tab labeled '.env.local'. The editor area contains three lines of text: 'SVR=server', 'SVR\_CMP=servercomponent', and 'NEXT\_PUBLIC\_CLT\_CMP=clientcomponent'.

```
.env.local

SVR=server
SVR_CMP=servercomponent
NEXT_PUBLIC_CLT_CMP=clientcomponent
```

**Note:** To make the value of an environment variable accessible in a client component, we need to prefix the variable with `NEXT_PUBLIC_`





## Step #3: Access variable from server

We can access an environment variable from a **route handler** like the following, located at **src/app/api** in our case:

```
import { NextResponse } from 'next/server';

export const GET = async () => {
  console.log(process.env.SVR);

  return NextResponse.json({
    status: 'success'
  });
};
```

Accessing the endpoint: **localhost:3000/api** in the browser will display the environment variable log on our IDE terminal.





## Step #4: Access variable from server component

We can access an environment variable from a **server component** like the following, located at **src/app** in our case:

```
import Header from './components/Header';

export default function Home() {
  console.log(process.env.SVR_CMP);

  return (
    <main>
      <Header />
      Home Page
    </main>
  );
}
```

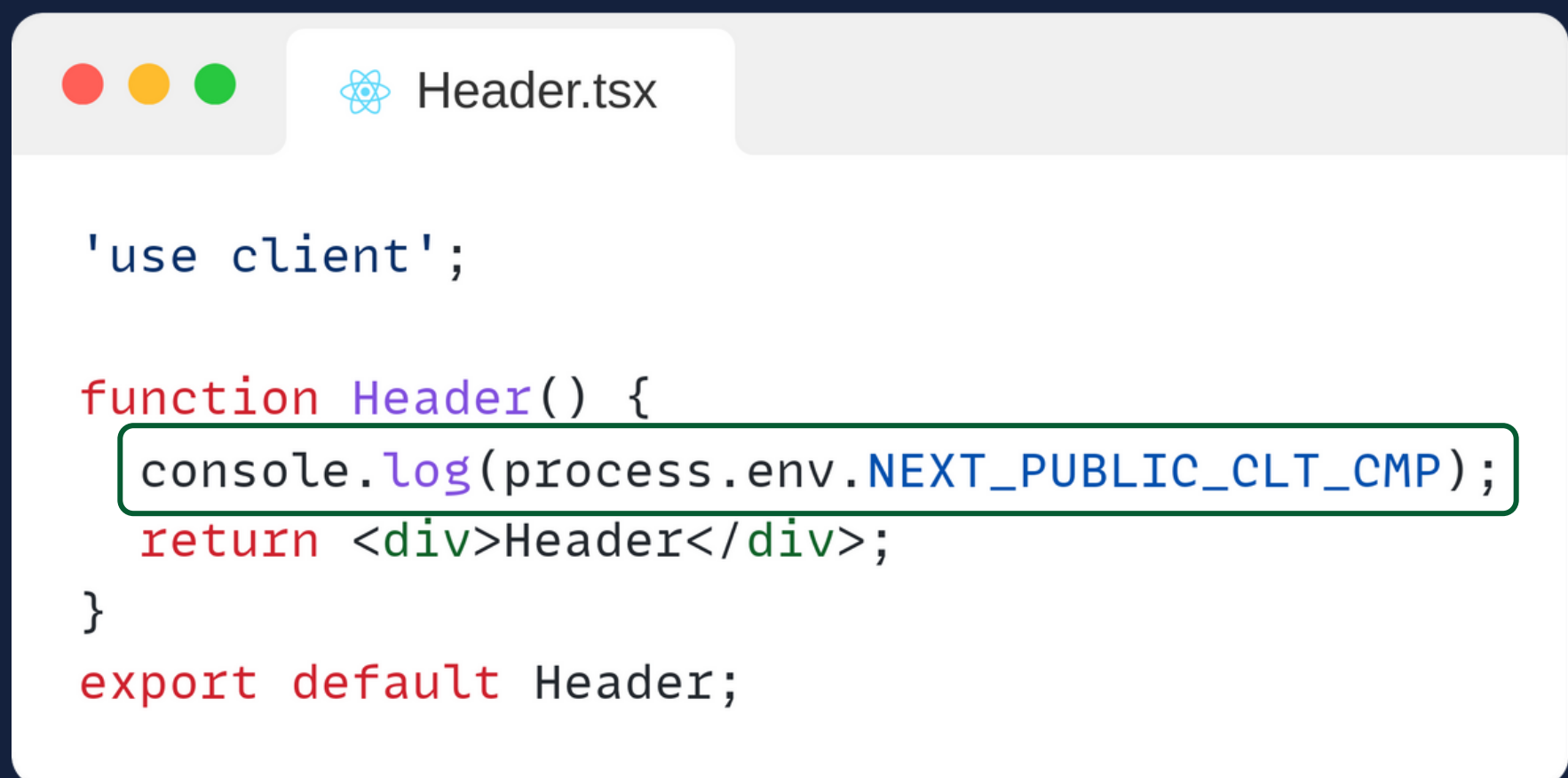
client component





## Step #5: Access variable from client component

We can access an environment variable from a **client component** like the following, located at **src/app/components** in our case:

A code editor window titled 'Header.tsx' with a React logo icon. It contains the following TypeScript code: 

```
'use client';

function Header() {
  console.log(process.env.NEXT_PUBLIC_CLT_CMP);
  return <div>Header</div>;
}

export default Header;
```

 The line `console.log(process.env.NEXT_PUBLIC_CLT_CMP);` is highlighted with a green rectangular border.

Refreshing the browser will display the environment variable log on our **browser** terminal.







**That's a wrap, everyone! 🙌**  
**Thanks for joining.**

If you're interested in delving deeper into **Environment variables** on Next.js, check out their official web page at:  
<https://nextjs.org/docs/app/building-your-application/configuring/environment-variables>



**Wasif Alam**  
@wasif alam



10/10

🚀 Implement secure environment variable management now to fortify your Next.js applications!

Enjoyed this guide? Let's connect on LinkedIn for more insightful content like this! 💡



Wasif Alam  
@wasif alam

