https://jsmastery.pro

Let's Talk About URL State Management

JavaScript Mastery

What is State Management?

State management in web development refers to how **data is stored and shared across components** of a web application.

It ensures that the application's state (data and UI) remains **consistent** and **synchronized.**

```
import React, { useState } from 'react';

function Counter() {
 const [count, setCount] = useState(0);

 return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
 );
}

export default Counter;
```

State Management Methods

There are **various ways** to manage the state of your application.

Depending on whether the state needs to be managed on a **local** or **global** level, you may reach for state management tools such as:

- **The useState hook**
- **Context API, or**
- **Redux/Redux Toolkit**

But have you tried storing the state in the **URL?**

Enter URL State Management

Have you ever visited an **e-commerce** store? Say it's an online store that **sells t-shirts.**

You choose a t-shirt and select a **color.** If you now look at the URL, you might notice **the selected color** there.

In this way, **the state is being stored in the URL -** which we can then retrieve later where needed! 🔥

useSearchParams()

In Next.js, we can use hooks such as **useSearchParams()** to read the data from the URL.

useSearchParams

`useSearchParams` is a **Client Component** hook that lets you read the current URL's **query string**.

`useSearchParams` returns a **read-only** version of the `URLSearchParams` ↗ interface.

app/dashboard/search-bar.js — JavaScript

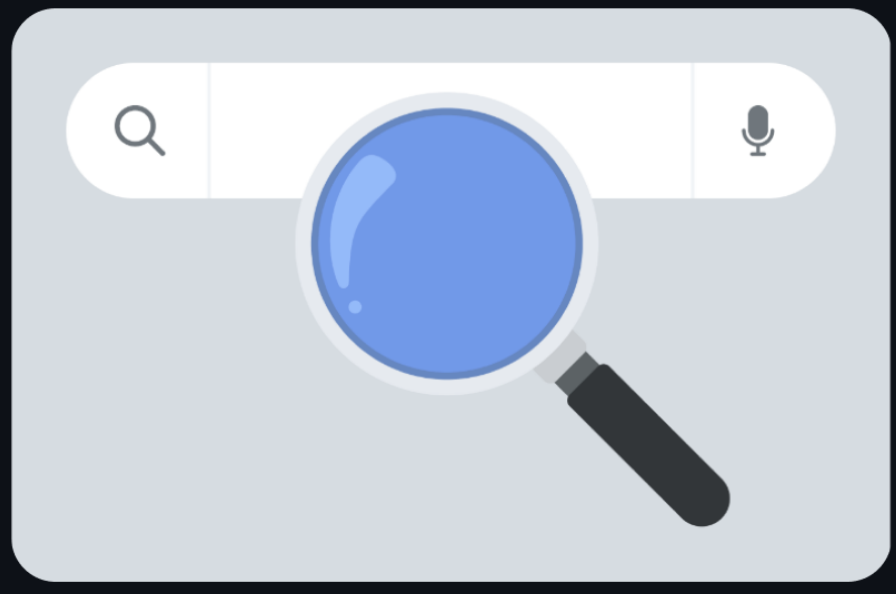
```
'use client'

import { useSearchParams } from 'next/navigation'

export default function SearchBar() {
  const searchParams = useSearchParams()

  const search = searchParams.get('search')

  // URL -> `/dashboard?search=my-project`
  // `search` -> 'my-project'
  return <>Search: {search}</>
}
```

When can it be used?

URL state management is great for things like:

- **pagination**
- **search boxes**
- **filtering**
- **modal visibility, and**
- **data fetching.**

But it should **NOT** be used when the state has **sensitive information.**



What are some advantages?

- **Bookmarkability & Easy Sharing** - If you bookmark a page, it will **look the same** when you return to it later. In addition, anyone who clicks on the shared URL **will also see exactly** what the original user was viewing.

- **Native Browser Navigation Support**- Users can navigate through different states of the application using the **back and forward buttons**.

- **Flexibility and Integration** - It **can be combined with other state management solutions** like Redux or Context API in React, allowing developers to leverage the strengths of both approaches.

Have you tried URL state management before?

Would you like to use it in your next project?

https://jsmastery.pro/next14

FOR DEVELOPERS WHO WANT TO STAND OUT AND BECOME IRREPLACEABLE

Become a top 1% Next.js 14 developer in only one course

Next.js 14 Unleashed: Elevate Your Developer Status

Gain real-world experience and land that dev job you've always imagined