

Dynamically update title

```
import { DynamicMetadataNamingForClientServer }
from "../dynamicMetadataUseClient";

export const metadata = {
  title: Counter
  ${DynamicMetadataNamingForClientServer.count}
}

export default function CounterPage(){
  return (
    <>
      <DynamicMetadataNamingForClientServer/>
    </>
  )
}
```

this provide an error:
Property 'count' does not exist on type '() => Element'.

and
"use client";

```
import {useState} from "react";
```

```
export const DynamicMetadataNamingForClientServer
= () =>{
  const [count, setCount] = useState(0);
  return(
    <>
      <p>Count: {count}</p>
      <button onClick={()=>{setCount(count+1)}}>click
for Increament</button>
    </>
  )
}
```

The error you're seeing:

ts

Property 'count' does not exist on type '() => Element'.

is caused because you're trying to access

`DynamicMetadataNamingForClientServer.count`, but

`DynamicMetadataNamingForClientServer` is a **React component**, not an object or class with properties.

What's Happening?

You have:

```
ts

export const metadata = {
  title: `Counter ${DynamicMetadataNamingForClientServer.count}` // ❌ ERROR
}
```

But `DynamicMetadataNamingForClientServer` is a function component:

```
tsx

export const DynamicMetadataNamingForClientServer = () => {
  const [count, setCount] = useState(0);
  return (
    <>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>click for Increment</button>
    </>
  );
};
```

Since it's a function, it doesn't have a `.count` property.

What's Your Goal?

If your **goal** is to dynamically update the metadata (like the title) based on the counter value, then you **can't do that statically** at the top level like:

```
ts
```

```
export const metadata = { title: ... }
```

because `metadata` in Next.js is **static or server-side evaluated**, and React state (like `useState`) only exists **client-side**.

✓ Solutions Based on Your Goals

🎯 If You Want to Dynamically Change the Document Title Based on `count` (Client-side)

Use the `useEffect` hook on the client to update the document title:

```
tsx

"use client";

import { useState, useEffect } from "react";

export const DynamicMetadataNamingForClientServer = () => {
  const [count, setCount] = useState(0);

  useEffect(() => {
    document.title = `Counter ${count}`; // Dynamically sets title in browser
  }, [count]);

  return (
    <>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Click to Increment</button>
    </>
  );
};
```

And in your page file:

```
tsx

import { DynamicMetadataNamingForClientServer } from "../dynamicMetadataUseClient";

export default function CounterPage() {
```

```
return <DynamicMetadataNamingForClientServer />;  
}
```

✗ What You Tried Won't Work Because:

- Static metadata can't access runtime component state.
 - `useState` doesn't exist at the time `metadata` is evaluated.
-

✓ Summary

If you want metadata to **change based on state**, you must do it **client-side** using `document.title`, not through `export const metadata`.

Let me know if you're trying to update metadata **server-side based on some param**, I can help with that too.