Cadriciel Web côté client

Cours 9 - Pages dynamiques

Table des matières

Routes/Segments Dynamiques - Définition

Il est souvent possible que la route soit appelée pour montrer un item dynamique provenant d'une base de données, comme un article de blog.

Il serait inconsevable que chaque article de blog soit fait à partir d'une page statique manuellement écrite, publiée sur Git et mise en production (i.e. Vercel) ...

Dans ce cas, nous ne pouvons pas prévoir le nom exact d'un segment de la route, comme le nom de l'article.

Est-ce qu'il y aurait une solution à ce cas de figure ?

Convention

Un segment dynamique peut être créé en enveloppant le nom d'un dossier entre crochets : [folderName]. Par exemple, [id] ou [slug].

Les segments dynamiques sont passés en tant que paramètres "prop" aux fonctions layout, page, route et generateMetadata.

Convention Globales Typescript

Convention des Segments, nommage des dossiers et typage TS :

Singulier : app/blog/[slug]/page.js
TS : { slug: string }

Catch-all : app/shop/[...slug]/page.js
TS - { slug: string[] }

C-all (Optional) : app/shop/[[...slug]]/page.js
TS : { slug?: string[] }

Multi : app/[category]/[item]/page.js
TS : { category: string, item: string }

Exemple

Chemin dossier et fichier : app/blog/[slug]/page.tsx

```tsx
export default function Page({ params }: { params: { slug: string } }) {
  return <div>My Post: {params.slug}</div>
}
```

Exemple

Route : app/blog/[slug]/page.js

URL : /blog/a

params : { slug: 'a' }

Variables de routage (next/navigation) : usePathname()

usePathname() est un Hook de Component client permettant de lire le chemin d'accès de l'URL actuel.

Utile pour changer un aspect graphique en fonction de l'URL actuel, comme possiblement un Header, un bouton vers un lien, etc.

Variables de routage (next/navigation) : usePathname()

```
'use client'

import { usePathname } from 'next/navigation'

export default function ExampleClientComponent() {
  const pathname = usePathname()
  return <p>Current pathname: {pathname}</p>
}
```

Exemples

const pathname = usePathname()

URL	Valeur de retour
/	'/'
/dashboard	'/dashboard'
/dashboard?v=2	'/dashboard'
/blog/hello-world	'/blog/hello-world'

Variables de routage (next/navigation) : useSearchParams()

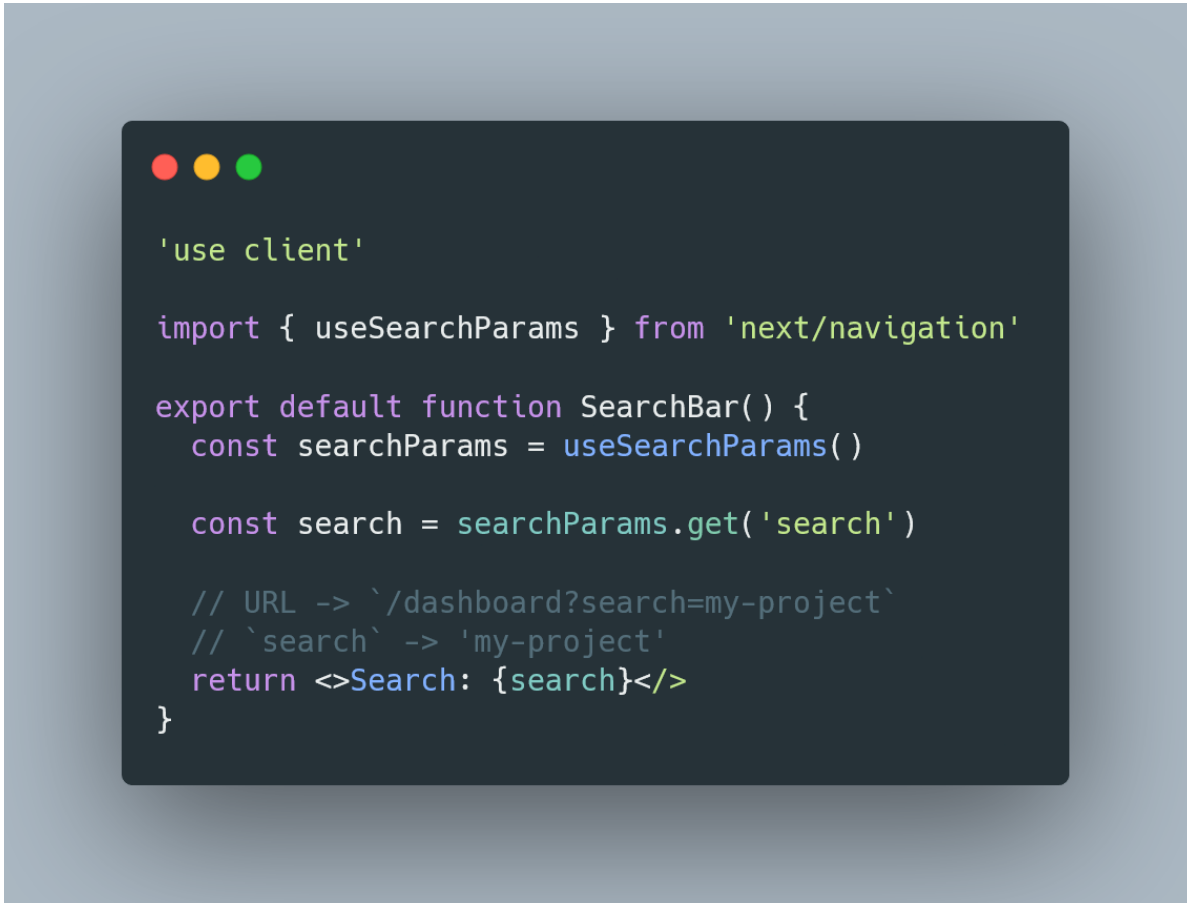
Comment lire un paramètre d'url ? (i.e. /store/clothes/shirt?size=small&color=blue)

Avec la variable useSearchParams.

useSearchParams est un hook de composant client permettant de lire la chaîne de requête de l'URL actuel.

NB : De manière professionelle, un URL plus propre n'utilise pas de paramètre. De plus, il n'est pas nécessaire d'avoir de paramètres avec la fonctionnalité de segment qui est équivalente.

Variables de routage (next/navigation) : useSearchParams()



```tsx
'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}</>
}
```

Exemples

const searchParams = useSearchParams()

URL	searchParams.get("a")
/dashboard?a=1	'1'
/dashboard?a=	''
/dashboard?b=3	null
/dashboard?a=1&a=2	'1' - getAll() pour avoir toutes les valeurs

URL	searchParams.has("a")
/dashboard?a=1	true
/dashboard?b=3	false

Bibliographie

- https://nextjs.org/docs/app/api-reference/functions/use-pathname
- https://nextjs.org/docs/app/api-reference/functions/use-search-params