

ATELIER 4 : DYNAMIC ROUTE NEXT JS

Introduction

Définir des routes à l'aide de chemins prédéfinis n'est pas toujours suffisant pour les applications complexes. Dans Next.js, vous pouvez ajouter des **crochets** à une page ([param]) pour créer une route dynamique (alias url slugs, id et autres).

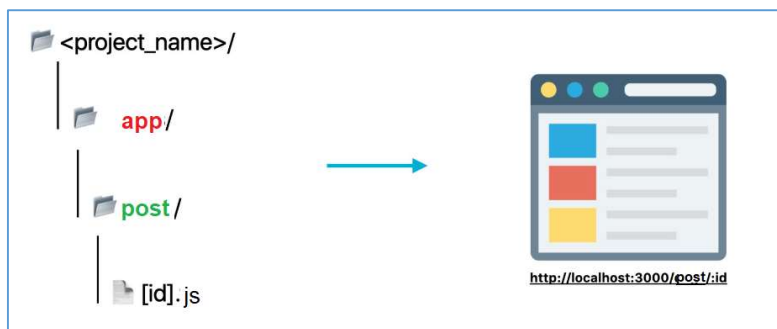
Le reste de la route est toujours défini par la hiérarchie des dossiers comme nous en avons vu dans le routage.

Exemples :

`app/post/[id].js` → `/post/:id` (`/post/123`)

`app/comments/[...all].js` → `/comments/*` (`/comments/2023/id/title`)

Si nous considérons le premier exemple, `app/post/[id].js`, toute route correspondant au modèle `/post/:id` sera rendue par le composant React exporté par la page spéciale `[id].js` dans le répertoire `app/post`, comme `/post/123`

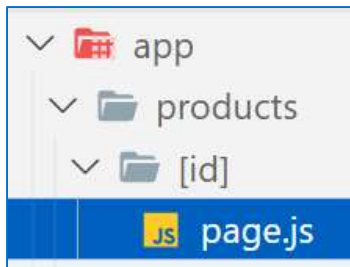


Le détail des articles

Créer le répertoire `[id]` sous le dossier `/app/products`



Créer dans ce répertoire le fichier `page.js`

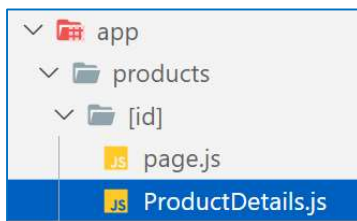


```
import React from 'react'
import ProductDetails from '../ProductDetails'

const ProductDetailsPage = ({params}) => {
  return (
    <ProductDetails params={params}/>
  )
}

export default ProductDetailsPage
```

Cr  er le fichier /app/products/[id]/ProductDetails.js



```
import React from 'react'

async function getProductDetails(id){
  try {
    const res = await fetch
    (`https://api.escuelajs.co/api/v1/products/${id}`);
    const product = await res.json();
    return product;
  }
  catch (error) {
    console.log(error);
  }
}

const ProductDetails = async ({params}) => {

  const product = await getProductDetails(params.id);

  return (
```

```

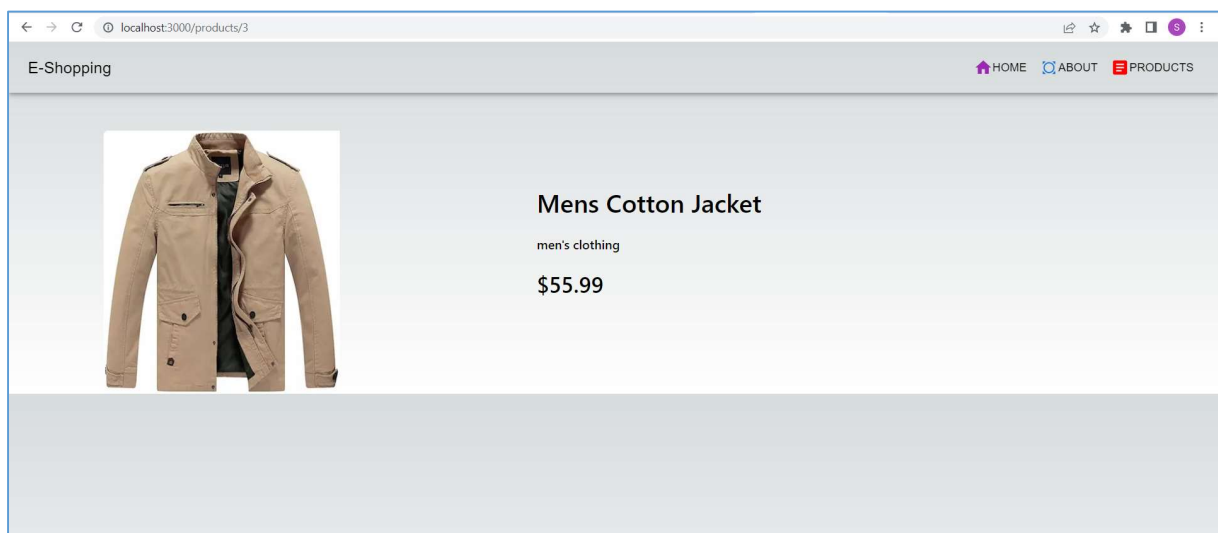
<div className="container">
  <div className="row mt-5">
    <div className="col-5">
      <Image
        src={product.images[0]}
        className="img-fluid rounded-start"
        height={500}
        width={500}
        alt={product.title}
      />
    </div>

    <div className="col-6 mt-5">
      <h2 className="my-4">{product.title}</h2>
      <h6 className="my-4">{product.category.name}</h6>
      <h3 className="my-4">${product.price}</h3>
    </div>

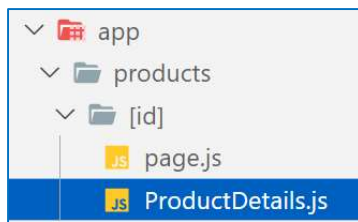
  </div>
</div>
)
}

export default ProductDetails

```



Modifier le fichier /app/products/[id]/ProductDetails.js



```
import React from 'react'

async function getProductDetails(id){
  try {
    const res = await fetch
    (`https://api.escuelajs.co/api/v1/products/${id}`);
    const product = await res.json();
    return product;
  }
  catch (error) {
    console.log(error);
  }
}

const ProductDetails = async ({params}) => {

  const product = await getProductDetails(params.id);

  return (
    <div className="container">
      <div className="row mt-5">
        <div className="col-5">
          <img
            src={product.images[0]}
            className="img-fluid rounded-start"
            height={500}
            width={500}
            alt={product.title}
          />
        </div>

        <div className="col-6 mt-5">
          <h2 className="my-4">{product.title}</h2>
          <h6 className="my-4">{product.category.name}</h6>
          <h3 className="my-4">${product.price}</h3>
        </div>

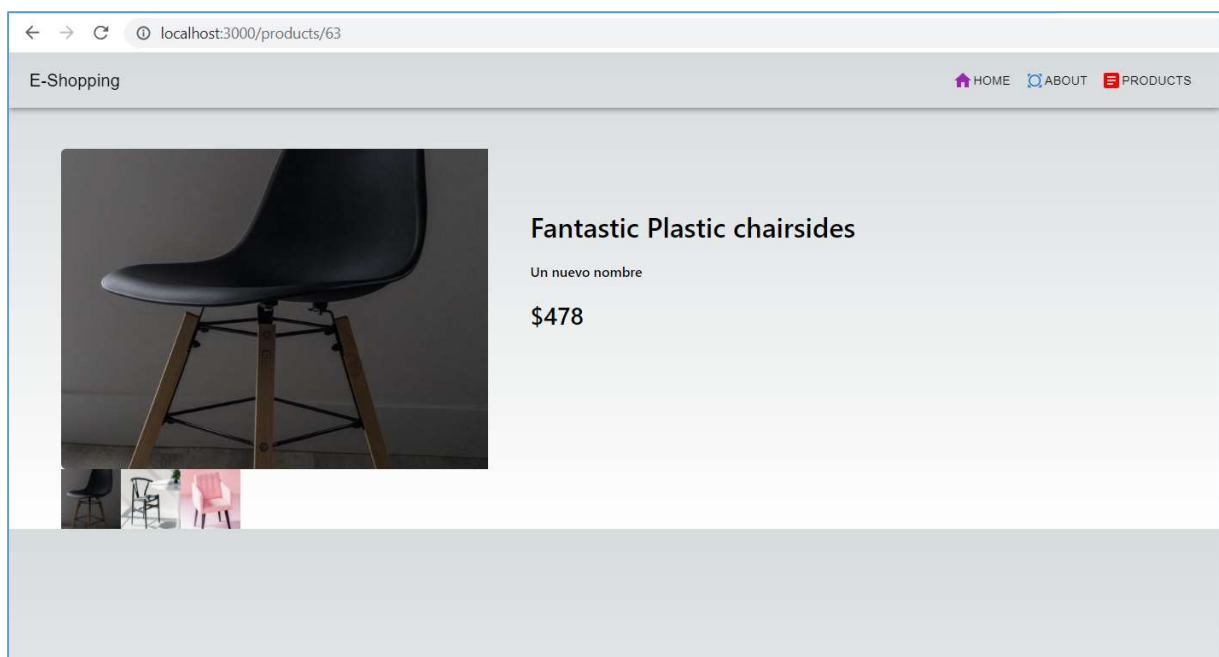
        <div >
          {product.images.map((image, index)=>(
            <img key={index} src={image}
```

```

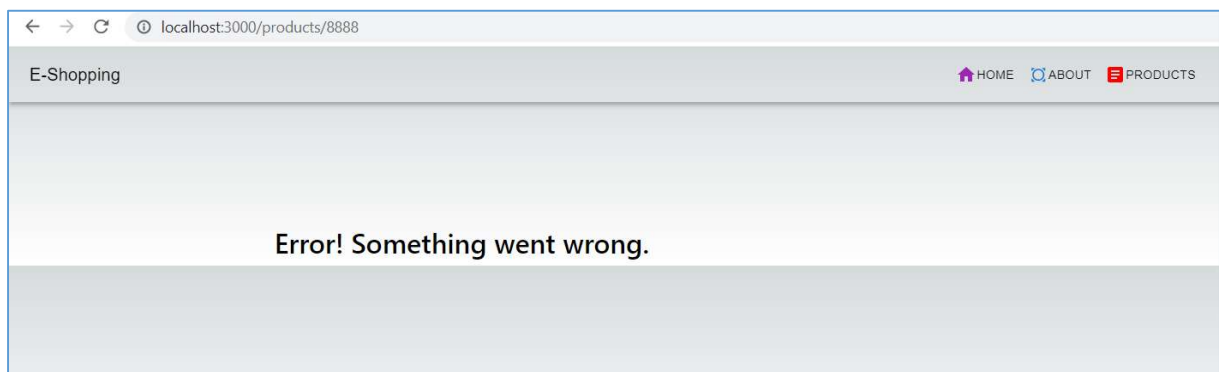
        height={70}
        width={70}
        alt={product.title}
      />
    )))
  </div>
</div>
)
}

export default ProductDetails

```

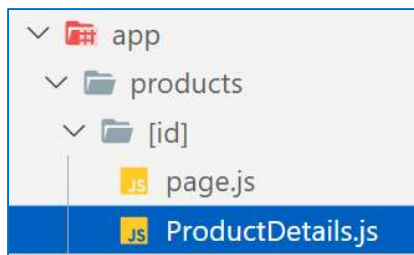


Si on met dans l'url une valeur d'id inexistante ça va générer une erreur.



Pour la corriger, on va ajouter une méthode appelée notFound générée par NextJS.

Modifier /app/products/[id]/ProductDetails.js



```
import React from 'react'

import { notFound } from "next/navigation";

async function getProductDetails(id){
  try {
    const res = await fetch
    (`https://api.escuelajs.co/api/v1/products/${id}`);
    const product = await res.json();
    return product;
  }
  catch (error) {
    console.log(error);
  }
}

const ProductDetails = async ({params}) => {

  const product = await getProductDetails(params.id);

  if (!product || !product?.title) notFound();

  return (
    <div className="container">
      <div className="row mt-5">
        <div className="col-5">
          <img
            src={product.images[0]}
            className="img-fluid rounded-start"
            height={500}
            width={500}
            alt={product.title}
          />
        </div>

        <div className="col-6 mt-5">
          <h2 className="my-4">{product.title}</h2>
          <h6 className="my-4">{product.category.name}</h6>
          <h3 className="my-4">${product.price}</h3>
        </div>
      </div>
    </div>
  )
}
```

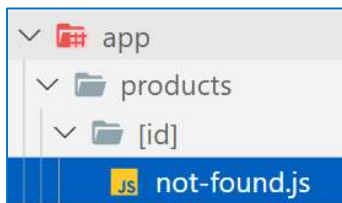
```

    <div >
      {product.images.map((image, index)=>(
        <img key={index} src={image}
          height={70}
          width={70}
          alt={product.title}
        />
      ))}
    </div>
  </div>
)
}

export default ProductDetails ;

```

Puis, créer le fichier not-found.js sous /app/products/[id]



```

import React from "react";

const NotFound = () => {
  return (
    <div className="d-flex justify-content-center mt-5">
      <h1>Product not found.</h1>
    </div>
  );
};

export default NotFound;

```

