

【Next.js】路由機制(動態)

在 Next.js 中，**Dynamic Routes (動態路由)** 允許你根據 URL 中的參數來動態建立頁面，這對於像是部落格文章、使用者頁面、商品詳細頁等非常有用。

□ 基本概念

在 Next.js 中，**每個檔案都對應一個路由**。當你在 `app/` 或 `pages/` 目錄中使用 **方括號 ([param])** 定義檔案名稱時，就代表建立了一個動態路由。

例如：

```
pages/  
└─ blog/  
   └─ [slug].tsx
```

這樣的結構可以對應：

- `/blog/hello-world`
- `/blog/my-second-post`

在 `app/` 目錄中則會是：

```
app/  
└─ blog/  
   └─ [slug]/  
      └─ page.tsx
```

□ Dynamic Routes 實作 (`app/`)

1 □ 建立檔案：`app/post/[id]/page.tsx`

```
interface PageProps {  
  params: { id: string }  
}  
  
export default function Post({ params }: PageProps) {  
  return <h1>Post ID: {params.id}</h1>  
}  
  
// { params }: PageProps  
// { params } => 解構，取出物件的 params 屬性  
// :PageProps typescript宣告型別  
// 直接簡化成 { params }: { params: { id: string } }
```

□ 多層動態路由

支援巢狀結構，例如：

```
app/  
└─ user/  
   └─ [userId]/  
      └─ post/  
         └─ [postId]/  
            └─ page.tsx
```

訪問路徑 `/user/1/post/10` 對應的 `params` 為：

```
params: {
  userId: '1',
  postId: '10'
}
```

☐ Catch-all 路由（捕捉所有）

你可以使用 `[...slug]` 來捕捉多層級路徑：

- `app/products/[...slug].tsx` 會匹配 `/products/a`、`/products/a/b/c`

```
interface PageProps {
  params: {
    slug?: string[];
  };
}

export default function ProductPage({ params }: PageProps) {
  return (
    <div>
      <h1>Products Page</h1>
      <pre>{JSON.stringify(params.slug)}</pre>
    </div>
  );
}
```

如果要可選，可用 `[...slug]` 表示非必要。

☐ 小結

類型	說明	範例
<code>[param]</code>	基本動態路由	<code>/post/1</code> → <code>[id].tsx</code>
<code>[...param]</code>	捕捉多層級路徑 (Catch-all)	<code>/docs/a/b</code> → <code>[...slug].tsx</code>
<code>[...param]</code>	可選 catch-all (Optional)	<code>/docs</code> or <code>/docs/a</code>

若你想看實際範例或搭配 `getStaticPaths`、`generateStaticParams` 使用，也可以提出，我可以補上。需要嗎？

🕒 修訂版本 #6

★ 由 treeman 建立於 25 🕒 2025 10:16:34

✍ 由 treeman 更新於 25 🕒 2025 14:31:46