> Menu



Q

# route.js

Route Handlers allow you to create custom request handlers for a given route using the Web Request and Response APIs.

#### **HTTP Methods**

A **route** file allows you to create custom request handlers for a given route. The following HTTP methods are supported: GET, POST, PUT, PATCH, DELETE, HEAD, and OPTIONS.

```
route.ts
                                                                      TypeScript ∨
                                                                                   1
     export async function GET(request: Request) {}
 2
 3
     export async function HEAD(request: Request) {}
 4
 5
     export async function POST(request: Request) {}
 6
 7
     export async function PUT(request: Request) {}
 9
     export async function DELETE(request: Request) {}
10
11
     export async function PATCH(request: Request) {}
12
13
     // If `OPTIONS` is not defined, Next.js will automatically implement `OPTIONS` and
     export async function OPTIONS(request: Request) {}
```

**Good to know**: Route Handlers are only available inside the App Router. You **do not** need to use API Routes (pages) and Route Handlers (app) together, as Route Handlers should be able to handle all use cases.

#### **Parameters**

#### request (optional)

The request object is a NextRequest object, which is an extension of the Web Request API.

NextRequest gives you further control over the incoming request, including easily accessing cookies and an extended, parsed, URL object nextUrl.

### context (optional)

```
app/dashboard/[team]/route.ts
                                                                      TypeScript ∨
                                                                                    type Params = {
 1
 2
       team: string
 3
 4
 5
     export async function GET(request: Request, context: { params: Params }) {
 6
       const team = context.params.team // '1'
 7
     }
 8
    // Define params type according to your route parameters (see table below)
```

Currently, the only value of context is params, which is an object containing the dynamic route parameters for the current route.

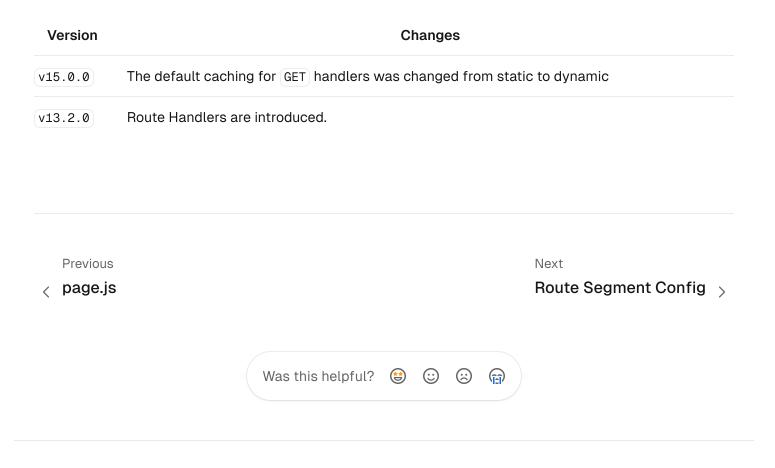
Example	URL	params
app/dashboard/[team]/route.js	/dashboard/1	{ team: '1' }
app/shop/[tag]/[item]/route.js	/shop/1/2	{ tag: '1', item: '2' }

Example	URL	params
app/blog/[slug]/route.js	/blog/1/2	{ slug: ['1', '2'] }

# **NextResponse**

Route Handlers can extend the Web Response API by returning a NextResponse object. This allows you to easily set cookies, headers, redirect, and rewrite. View the API reference.

# **Version History**



**About Vercel** 

Resources

More

**▲** Vercel

Legal

Docs Next.js Commerce Next.js + Vercel Privacy Policy

Learn Contact Sales Open Source Software

Showcase GitHub GitHub

Blog Releases X

Analytics Telemetry

Next.js Conf Governance

Previews

#### Subscribe to our newsletter

Stay updated on new releases and features, guides, and case studies.

you@domain.com

Subscribe

© 2024 Vercel, Inc.





