

[> Menu](#)

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`.

TS 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) {}
8
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
14 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)

TS app/dashboard/[team]/route.ts TypeScript

```
1  type Params = {
2    team: string
3  }
4
5  export async function GET(request: Request, context: { params: Params }) {
6    const team = context.params.team // '1'
7  }
8
9  // 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	<code>params</code>
<code>app/dashboard/[team]/route.js</code>	<code>/dashboard/1</code>	<code>{ team: '1' }</code>
<code>app/shop/[tag]/[item]/route.js</code>	<code>/shop/1/2</code>	<code>{ tag: '1', item: '2' }</code>





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

Version	Changes
v15.0.0	The default caching for <code>GET</code> handlers was changed from static to dynamic
v13.2.0	Route Handlers are introduced.

Was this helpful?    

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.

