

API Routes

Examples

Basic API Routes

API Routes with middleware

API Routes with GraphQL

API Routes with REST

API Routes with CORS

API routes provide a solution to build your **API** with Next.js.

Any file inside the folder `pages/api` is mapped to `/api/*` and will be treated as an API endpoint instead of a `page`. They are server-side only bundles and won't increase your client-side bundle size.

For example, the following API route `pages/api/user.js` returns a json response with a status code of 200:

```
export default function handler(req, res) {
  res.status(200).json({ name: 'John Doe' })
}
```

For an API route to work, you need to export a function as default (a.k.a **request handler**), which then receives the following parameters:

- **req**: An instance of `http.IncomingMessage`, plus some pre-built middlewares
- **res**: An instance of `http.ServerResponse`, plus some helper functions

To handle different HTTP methods in an API route, you can use `req.method` in your request handler, like so:

```
export default function handler(req, res) {
  if (req.method === 'POST') {
    // Process a POST request
  } else {
    // Handle any other HTTP method
  }
}
```

To fetch API endpoints, take a look into any of the examples at the start of this section.

Use Cases

For new projects, you can build your entire API with API Routes. If you have an existing API, you do not need to forward calls to the API through an API Route. Some other use cases for API Routes are:

- Masking the URL of an external service (e.g. `/api/secret` instead of `https://company.com/secret-url`)
- Using Environment Variables on the server to securely access external services.

Caveats

- API Routes do not specify CORS headers, meaning they are **same-origin only** by default. You can customize such behavior by wrapping the request handler with the CORS middleware.
- API Routes can't be used with `next export`

Related

For more information on what to do next, we recommend the following sections:

API Middlewares: learn about the built-in middlewares for the request.

Response Helpers: learn about the built-in methods for the response.

TypeScript: Add TypeScript to your API Routes.