## Creating API Routes with Next.js

In Next.js, API routes are created in the pages/api directory. [Each file in this directory corresponds to an API endpoint](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[1](https://nextjs.org/docs/pages/building-your-application/routing/api-routes). For example, if you create a file named hello.ts in the pages/api directory, you’ll have an API route at /api/hello.

Here’s a simple example of an API route that returns a “Hello, World!” message:

// pages/api/hello.ts

import type { NextApiRequest, NextApiResponse } from 'next'

type ResponseData = { message: string }

export default function handler(req: NextApiRequest, res: NextApiResponse<ResponseData>) {

res.status(200).json({ message: 'Hello from Next.js!' })

}

## Handling HTTP Requests and Responses

[In the handler function above, req is an object containing various HTTP incoming message properties (like req.body, req.headers, etc.), and res is an object for returning the response](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[1](https://nextjs.org/docs/pages/building-your-application/routing/api-routes).

You can handle different HTTP methods (like GET, POST, etc.) by checking req.method. Here’s an example:

// pages/api/hello.ts

import type { NextApiRequest, NextApiResponse } from 'next'

export default function handler(req: NextApiRequest, res: NextApiResponse) {

if (req.method === 'POST') {

// Process a POST request

} else {

// Handle any other HTTP method

}

}

## Integrating API Routes with Frontend Components

You can call these API routes from your frontend components using the fetch function. Here’s an example of a React component that calls the /api/hello route when a button is clicked:

import React from 'react'

export default function Home() {

const handleClick = async () => {

const res = await fetch('/api/hello')

const data = await res.json()

console.log(data)

}

return (

<div>

<button onClick={handleClick}>Call API</button>

</div>

)

}

[In this example, when the button is clicked, a GET request is made to the /api/hello route, and the response is logged to the console](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[1](https://nextjs.org/docs/pages/building-your-application/routing/api-routes).

## API Routes in Next.js 13 and Later Versions

[Since version 13.2, Next.js introduced API Route Handlers in the app folder](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[2](https://stackoverflow.com/questions/75418329/how-do-you-put-api-routes-in-the-new-app-folder-of-next-js). [They work like pages, but the file for the segment should be called route.js instead of page.js](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[2](https://stackoverflow.com/questions/75418329/how-do-you-put-api-routes-in-the-new-app-folder-of-next-js). Here’s an example:

// app/api/route.js

import { NextResponse } from 'next/server'

export async function GET(request) {

return NextResponse.json({ message: 'Hello World' }, { status: 200 })

}

export async function POST(request) {

return NextResponse.json({ message: 'Hello World' }, { status: 200 })

}

[In this example, you can access /api as any API route, for example, with fetch('/api')](https://nextjs.org/docs/pages/building-your-application/routing/api-routes)[2](https://stackoverflow.com/questions/75418329/how-do-you-put-api-routes-in-the-new-app-folder-of-next-js).