|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | const http = require('http');  const server = http.createServer((req, res) => {  console.log('URL:', req.url);  console.log('Method:', req.method);  console.log('Headers:', req.headers);  res.writeHead(200, { 'Content-Type': 'text/plain' });  res.write('Check your console for req details');  res.end();  });  server.listen(3000, () => {  console.log('Server is running on http://localhost:3000');  }); | | Plain text  res.write('Hello, plain text!');  **JSON**:  res.writeHead(200, { 'Content-Type': 'application/json' });  res.end(JSON.stringify({ message: 'Hello, JSON!' }));  **HTML**:  res.writeHead(200, { 'Content-Type': 'text/html' });  res.end('<h1>Hello, HTML!</h1>'); | | const http = require('http');  const server = http.createServer((req, res) => {  if (req.url === '/' && req.method === 'GET') {  res.writeHead(200, { 'Content-Type': 'text/plain' });  res.end('Welcome to the Home Page!');  } else if (req.url === '/about' && req.method === 'GET') {  res.writeHead(200, { 'Content-Type': 'text/plain' });  res.end('This is the About Page.');  } else if (req.url === '/api' && req.method === 'GET') {  res.writeHead(200, { 'Content-Type': 'application/json' });  res.end(JSON.stringify({ message: 'Welcome to the API' }));  } else {  res.writeHead(404, { 'Content-Type': 'text/plain' });  res.end('404 Not Found');  }  });  server.listen(3000, () => {  console.log('Server is running on http://localhost:3000');  }); | | console.log(req.url); // Example: "/about"  console.log(req.method); // Example: "GET"  console.log(req.headers); req.body The body of the request, often used with POST or PUT requests. To access req.body, you'll need to parse it using middleware or native code.  **Example (Using body-parser middleware with Express):**  const express = require('express');  const bodyParser = require('body-parser');  const app = express();  app.use(bodyParser.json()); // For parsing application/json  app.post('/submit', (req, res) => {  console.log(req.body); // Logs the body of the POST request  res.send('Data received');  });  app.listen(3000, () => {  console.log('Server running on http://localhost:3000');  }); | | req.query The query parameters in the URL (after ?).  **Example:** If the URL is /search?keyword=nodejs:  const express = require('express');  const app = express();  app.get('/search', (req, res) => {  console.log(req.query.keyword); // Output: "nodejs"  res.send(`Searching for: ${req.query.keyword}`);  });  app.listen(3000, () => {  console.log('Server running on http://localhost:3000');  }); | |
| req.params Route parameters, often used in dynamic routes.  **Example:** For the route /user/:id:  app.get('/user/:id', (req, res) => {  console.log(req.params.id); // Output: The value of :id  res.send(`User ID: ${req.params.id}`);  }); |
|  |
| console.log(req.ip); // Example: "::1" (localhost)  console.log(req.protocol); // Example: "http"  console.log(req.hostname); // Example: "localhost"  // req.originalUrl  // The full URL path of the request, including query string.  console.log(req.originalUrl); // Example: "/search?keyword=nodejs"  // req.baseUrl  // The base URL when the route is accessed via a router.  // Example: For a sub-route:  const express = require('express');  const router = express.Router();  router.get('/profile', (req, res) => {  console.log(req.baseUrl); // Example: "/user"  res.send('User Profile');  });  app.use('/user', router);  // req.secure  // Returns true if the request was made over HTTPS.  console.log(req.secure); // Example: false (if HTTP)  // req.cookies  // To access cookies sent by the client, use middleware like cookie-parser.  const cookieParser = require('cookie-parser');  app.use(cookieParser());  app.get('/', (req, res) => {  console.log(req.cookies); // Logs the cookies sent by the client  res.send('Cookies received');  }); |
| // req.get(headerName)  // Retrieve a specific header.  console.log(req.get('User-Agent')); // Example: "Mozilla/5.0"  // req.is(type)  // Check if the incoming request has a specific content type.  app.post('/upload', (req, res) => {  if (req.is('application/json')) {  res.send('JSON data received');  } else {  res.status(400).send('Invalid content type');  }  }); |
|  |
|  |
|  |