# Request Parameters in Express

Jul 16, 2019

The first parameter to Express route handlers and middleware functions is the [Express request object](https://expressjs.com/en/api.html" \l "req). This parameter is usually called req.

**const** express = require('express');**const** app = express();

app.get('\*', **function**(req, res) {

*// `req` is an instance of Node.js' built-in HTTP request class,*

*// with some additional features from Express*

req **instanceof** require('http').IncomingMessage; *// true*

res.json({ ok: 1 });

});

**const** server = **await** app.listen(3000);

## Request Parameters

Express parses [query string](https://en.wikipedia.org/wiki/Query_string) parameters by default, and puts them into the [req.query property](https://expressjs.com/en/api.html" \l "req.query).

**const** axios = require('axios');**const** express = require('express');**const** app = express();

app.get('\*', **function**(req, res) {

**const** name = req.query.name; *// 'Jean-Luc Picard'*

**const** rank = req.query.rank; *// 'Captain'*

res.json({ name, rank });

});

**const** server = **await** app.listen(3000);

*// Send a GET request to the server with URL-encoded params in the// query string***const** querystring = 'name=Jean-Luc Picard&rank=Captain';**const** res = **await** axios.get('http://localhost:3000?' + querystring);

res.data.name; *// 'Jean-Luc Picard'*

res.data.rank; *// 'Captain'*

Express also supports [named route parameters](https://expressjs.com/en/api.html" \l "req.params) and puts them in the req.params object. Named route parameters are always strings, and Express automatically decodes them using [decodeUriComponent()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/decodeURIComponent).

**const** axios = require('axios');**const** express = require('express');**const** app = express();

app.get('/:model/:id', **function**(req, res) {

**const** model = req.params.model; *// 'user'*

**const** id = req.params.id; *// '1'*

res.json({ model, id });

});

**const** server = **await** app.listen(3000);

*// Send a GET request to the server with URL params***const** res = **await** axios.get('http://localhost:3000/user/1');

res.data.model; *// 'user'*

res.data.id; *// '1'*

Express does **not** parse the [request body](https://masteringjs.io/tutorials/express/body) for you by default. To opt in to parsing JSON request bodies, use the [express.json() middleware](https://expressjs.com/en/api.html" \l "express.json). Express will then parse the HTTP request body and put the parsed body in req.body.

**const** axios = require('axios');**const** express = require('express');**const** app = express();

*// Parse the request body as JSON. Requires Express >= 4.16.0.*

app.use(express.json());

app.put('\*', **function**(req, res) {

**const** name = req.body.name; *// 'Jean-Luc Picard'*

**const** rank = req.body.rank; *// 'Captain'*

res.json({ name, rank });

});

**const** server = **await** app.listen(3000);

*// Send a PUT request to the server with a request body***const** body = { name: 'Jean-Luc Picard', rank: 'Captain' };**const** res = **await** axios.put('http://localhost:3000', body);

res.data.name; *// 'Jean-Luc Picard'*

res.data.rank; *// 'Captain'*

## Headers

To get the value of an [HTTP request header](https://developer.mozilla.org/en-US/docs/Glossary/Request_header), you should use Express' [req.get() function](https://expressjs.com/en/api.html" \l "req.get). You may also use [Node.js' native req.headers property](https://nodejs.org/api/http.html" \l "http_message_headers).

**const** axios = require('axios');**const** express = require('express');**const** app = express();

app.get('\*', **function**(req, res) {

*// `req.get()` is case-insensitive.*

**const** authorization = req.get('authorization');

*// Or you can use `req.headers`*

req.headers.authorization;

res.json({ authorization });

});

**const** server = **await** app.listen(3000);

*// Send a GET request to the server with an 'Authorization' header***const** res = **await** axios.get('http://localhost:3000', {

headers: {

'Authorization': 'test'

}

});

res.data.authorization; *// 'test'*

Here's a [tutorial on how to set request headers in Axios](https://masteringjs.io/tutorials/axios/headers) if you're unfamiliar with Axios.

## Body Size Limit

By default, [express.json() limits the request body to 100kb](https://expressjs.com/en/api.html" \l "express.json). If the request body is any larger, Express will throw an [HTTP 413 "Payload Too Large" error](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/413). You can configure this limit using the limit option to express.json().

**const** axios = require('axios');**const** express = require('express');**const** app = express();

*// Set the body size limit to 10 bytes*

app.use(express.json({ limit: 10 }));

app.put('\*', **function**(req, res) {

**const** name = req.body.name; *// 'Jean-Luc Picard'*

**const** rank = req.body.rank; *// 'Captain'*

res.json({ name, rank });

});

**const** server = **await** app.listen(3000);

*// Send a PUT request to the server with a request body***const** body = { name: 'Jean-Luc Picard', rank: 'Captain' };**const** err = **await** axios.put('http://localhost:3000', body).

catch(err => err);

err.response.status; *// 413*