What does body-parser do with express?

Asked 5 years, 1 month ago Active 4 months ago Viewed 340k times

Ask Question

420

I don't understand why we need `body-parser` in an Express application, as we can get data without using `body-parser`. And what does it do actually and how?

node.js express body-parser

139

Share Improve this question Follow — edited Jul 24 '17 at 16:51 Andrew Li 48.4k ●10 ●109 ●133 — asked Jul 11 '16 at 12:08 jiten 4,245 ●3 ●10 ●7

76 in order to read HTTP POST data , we have to use "body-parser" node module. body-parser is a piece of express middleware that reads a form's input and stores it as a javascript object accessible through `req.body` – refactor Jul 11 '16 at 12:19

4 With express you can read any data inside HTTP request, such as headers `req.headers` (array), you can read the body of the http packet as `req.body` explained by @CleanCrispCode and you can read as query parameter `req.query.variable`, It helps since express automatically transforms the request in javascript objects – Fernando Zamperin Jul 11 '16 at 17:54

5 @refactor -- this might be *one* of the many reasons we *have to use* body parser, but it doesn't say what it does, i.e. that HTTP request and response objects are streams and that they're not 'readable' as single object like `res.body` without the entire stream being buffered into `res.body` first.

Join Stack Overflow to learn, share knowledge, and build your career. Sign up with email Sign up with Google Sign up with GitHub Sign up with Facebook

1 Also see You don't need body-parser in Express 4.16+ – Shivam Jha Sep 21 '20 at 20:48

Add a comment

10 Answers

Active Oldest Votes

321

To handle `HTTP POST` requests in **Express.js** version 4 and above, you need to install the middleware module called `body-parser`.

`body-parser` extracts the entire body portion of an incoming request stream and exposes it on `req.body`.

The middleware was a part of Express.js earlier but now you have to install it separately.

This `body-parser` module parses the JSON, buffer, string and URL encoded data submitted using `HTTP POST` request. Install `body-parser` using NPM as shown below.

```
npm install body-parser --save
```

edit in 2019-april-2: in express@4.16.0 the body-parser middleware bundled with express. for more details see this

Share Improve this answer Follow — edited Mar 28 at 12:36 Jasper 90 ●10 — answered Apr 26 '17 at 6:47 Malatesh Patil 3,785 ●1 ●12 ●14

188 This is quite possibly the lamest thing ever. Why would Express core devs make it incredibly difficult for newcomers to get on board by making them install additional middleware for the most common use cases in web development? – elmt Apr 5 '18 at 18:34

7 @elmt if you want something with opinions, try sails.js – George Apr 13 '18 at 17:07

2 @user1063287 yes it does. `urlencoded()` and `json()` are actually middleware factories that return a middleware function which invokes `next()` – Nick Manning Jun 14 '18 at 16:50

3 It is not lame @elmt, node is not only for web, it can be used on desktop, mobile, etc, and in these cases it is not a required module. Node can adapt to your application without any liability – fnaquira Jul 12 '18 at 19:42

39 @fnaquira - You're confused. This is about express not node. – elmt Jul 12 '18 at 20:46

Show **8** more comments

107

Yes we can work without `body-parser`. When you don't use that you get the raw request, and your body and headers are not in the root object of request parameter . You will have to individually manipulate all the fields.

Or you can use `body-parser`, as the express team is maintaining it .

What body-parser can do for you: It simplifies the request.
How to use it: Here is example:

Install `npm install body-parser --save`

This how to use body-parser in express:

```
const express = require('express'),
      app = express(),
      bodyParser = require('body-parser');

// support parsing of application/json type post data
app.use(bodyParser.json());

//support parsing of application/x-www-form-urlencoded post data
app.use(bodyParser.urlencoded({ extended: true }));
```

Link.

https://github.com/expressjs/body-parser.

And then you can get body and headers in root request object . Example

```
app.post("/posturl",function(req,res,next){
    console.log(req.body);
    res.send("response");
})
```

Share Improve this answer Follow — edited Aug 6 '19 at 11:52 — answered Jan 21 '18 at 7:53 Himanshu sharma 6,033 ●4 ●37 ●62

4 Hey thanks for the info, can you post a code example without body parser? – Ilyas karim Oct 23 '19 at 14:06

1 @Ilyas you can check some blog itnext.io/.... Here they have use http module of node.js , same way you can use in express also , inside `app.post("/posturl",function(req,res,next){` – Himanshu sharma Aug 24 '20 at 4:14

Add a comment

67

The answer here explain it very detailed and brilliantly, the answer contains:

> In short; *body-parser* extracts the entire body portion of an incoming request stream and exposes it on `req.body` as something easier to interface with. You don't need it per se, because you could do all of that yourself. However, it will most likely do what you want and save you the trouble.
>
> ---
>
> To go a little more in depth; body-parser gives you a middleware which uses nodejs/zlib to unzip the incoming request data if it's zipped and stream-utils/raw-body to await the full, raw contents of the request body before "parsing it" (this means that if you weren't going to use the request body, you just wasted some time).
>
> After having the raw contents, body-parser will parse it using one of four strategies, depending on the specific middleware you decided to use:
>
> - bodyParser.raw(): Doesn't actually parse the body, but just exposes the buffered up contents from before in a Buffer on `req.body`.
> - bodyParser.text(): Reads the buffer as plain text and exposes the resulting string on req.body.
> - bodyParser.urlencoded(): Parses the text as URL encoded data (which is how browsers tend to send form data from regular forms set to POST) and exposes the resulting object (containing the keys and values) on `req.body`. For comparison; in PHP all of this is automatically done and exposed in `$_POST`.
> - bodyParser.json(): Parses the text as JSON and exposes the resulting object on `req.body`.
>
> Only after setting the `req.body` to the desirable contents will it call the next middleware in the stack, which can then access the request data without having to think about how to unzip and parse it.

You can refer to *body-parser* github to read their documentation, it contains information regarding its working.

Share Improve this answer Follow — edited Jun 20 '20 at 9:12 Community ♦ 1 ●1 — answered Nov 25 '17 at 12:52 Suraj Jain 3,977 ●24 ●39

Add a comment

64

Let's try to keep this least technical.

Let's say you are sending a html form data to node-js server i.e. you made a request to the server. The server file would receive your request under a request object. Now by logic, if you console log this request object in your server file you should see your form data some where in it, which could be extracted then, but whoa ! you actually don't !

So, where is our data ? How will we extract it if its not only present in my request.

Simple explanation to this is http sends your form data in bits and pieces which are intended to get assembled as they reach their destination. So how would you extract your data.

But, why take this pain of every-time manually parsing your data for chunks and assembling it. Use something called "body-parser" which would do this for you.

body-parser parses your request and converts it into a format from which you can easily extract relevant information that you may need.

For example, let's say you have a sign-up form at your frontend. You are filling it, and requesting server to save the details somewhere.

Extracting username and password from your request goes as simple as below if you use body-parser.

```
var loginDetails = {
    username : request.body.username,
    password : request.body.password
};
```

So basically, body-parser parsed your incoming request, assembled the chunks containing your form data, then created this body object for you and filled it with your form data.

Share Improve this answer Follow — edited Aug 19 '18 at 22:04 Brian Tompsett - 汤莱恩 5,241 ●67 ●51 ●120 — answered Aug 19 '18 at 21:54 abdulwahid211 636 ●5 ●4

Add a comment

15

In order to get access to the post data we have to use `body-parser`. Basically what the `body-parser` is which allows express to read the body and then parse that into a `Json` object that we can understand.

Share Improve this answer Follow — edited Jul 26 '17 at 15:53 Utkarsh Dubey 689 ●9 ●29 — answered Jul 26 '17 at 12:23 Satish Kuppili 167 ●1 ●3

Add a comment

11

It parses the HTTP request body. This is usually necessary when you need to know more than just the URL you hit, particular in the context of a POST or PUT PATCH HTTP request where the information you want is contains in the body.

Basically its a middleware for parsing JSON, plain text, or just returning a raw Buffer object for you to deal with as you require.

Share Improve this answer Follow — answered Jul 26 '17 at 17:31 Deepak Patidar 374 ●1 ●8 ●23

Add a comment

9

Understanding Requests Body

> When receiving a POST or PUT request, the request body might be important to your application. Getting at the body data is a little more involved than accessing request headers. The request object that's passed in to a handler implements the ReadableStream interface. This stream can be listened to or piped elsewhere just like any other stream. We can grab the data right out of the stream by listening to the stream's 'data' and 'end' events.
>
> The chunk emitted in each 'data' event is a Buffer. If you know it's going to be string data, the best thing to do is collect the data in an array, then at the 'end', concatenate and stringify it.
>
> ```
> let body = [];
> request.on('data', (chunk) => {
>   body.push(chunk);
> }).on('end', () => {
>   body = Buffer.concat(body).toString();
>   // at this point, `body` has the entire request body stored in it as a stri
> });
> ```

Understanding body-parser

As per its documentation

> Parse incoming request bodies in a middleware before your handlers, available under the req.body property.

As you saw in the first example, we had to parse the incoming request stream manually to extract the body. This becomes a tad tedious when there are multiple form data of different types. So we use the body-parser package which does all this task under the hood.

It provides four modules to parse different types of data

- JSON body parser
- Raw body parser
- Text body parser
- URL-encoded form body parser

After having the raw content body-parser will use one of the above strategies(depending on middleware you decided to use) to parse the data. You can read more about them by reading their documentation.

After setting the `req.body` to the parsed body, body-parser will invoke `next()` to call the next middleware down the stack, which can then access the request data without having to think about how to unzip and parse it.

Share Improve this answer Follow — answered Jul 31 '19 at 9:25 Himansh 599 ●6 ●11

Add a comment

9

If you don't want to use seperate npm package body-parser, latest express (4.16+) has built-in body-parser middleware and can be used like this,

```
const app = express();
app.use(express.json({ limit: '100mb' }));
```

p.s. Not all functionalities of body parse are present in the express. Refer documentation for full usage here

Share Improve this answer Follow — answered Oct 19 '20 at 19:02 AnandShiva 471 ●6 ●12

1 Note: Internally the Expess 4.16+ still uses library 'body-parser'. – Martin Mar 31 at 8:25

Add a comment

8

These are all a matter of convenience.

Basically, if the question were 'Do we *need* to use `body-parser`?' The answer is 'No'. We can come up with the same information from the client-post-request using a more circuitous route that will generally be less flexible and will increase the amount of code we have to write to get the same information.

This is kind of the same as asking 'Do we *need* to use `express` to begin with?' Again, the answer there is no, and again, really it all comes down to saving us the hassle of writing more code to do the basic things that express comes with 'built-in'.

On the surface - `body-parser` makes it easier to get at the information contained in client requests in a variety of formats instead of making you capture the raw data streams and figuring out what format the information is in, much less manually parsing that information into useable data.

Share Improve this answer Follow — answered Nov 1 '17 at 9:31 Rich Werden 735 ●9 ●11

Add a comment

4

Keep it simple :

- if you used `post` request so you will need the `body` of the request, so you will need `body-parser`.
- No need to install *body-parser* with `express`, but you have to `use` it if you will receive post request.

`app.use(bodyParser.urlencoded({ extended: false }));`

```
{ extended: false }
```

false meaning, you do not have nested data inside your *body* object. Note that: the request data embedded within the request as a body Object.

Share Improve this answer Follow — edited Apr 28 at 19:51 — answered Dec 12 '20 at 0:36 Hossam Maher 1,487 ●2 ●14 ●25

Add a comment

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.

Not the answer you're looking for? Browse other questions tagged node.js express body-parser or ask your own question.

The Overflow Blog
- Podcast 371: Exploring the magic of instant Python refactoring with Sourcery
- Pandemic lockdowns accelerated cloud migration by three to four years

Featured on Meta
- Review queue workflows - Final release
- Please welcome Valued Associates: #958 - V2Blast & #959 - SpencerG
- Outdated Answers: unpinning the accepted answer A/B test

Linked
- 841 How to access POST form fields
- 12 node.js, express, how to get data from body form-data in post request
- 3 Apollo Server 2 + Express: req.body missing on post handler
- 3 Accessing raw POST data in Express
- 2 Why is req.body undefined in Express?
- 3 Why is req.body returning undefined in POSTMAN
- 4 Accept POST request from other website to Angular Application
- 2 Why do we need express body-parser?

See more linked questions

Related
- 2194 How to decide when to use Node.js?
- 1841 How can I update NodeJS and NPM to the next versions?
- 1478 How do I completely uninstall Node.js, and reinstall from beginning (Mac OS X)
- 482 What is Express.js?
- 1043 How to install a previous exact version of a NPM package?
- 2535 What is the --save option for npm install?
- 515 Proper way to return JSON using node or Express
- 593 bodyParser is deprecated express 4
- 101 What does 'extended' mean in express 4.0?

Hot Network Questions
- Resource linking fails on lStar
- Are there (small, pod) hotels for single people in Western countries?
- Connect 3/8-16 bolts at 90 degrees
- Should I gave a make up lecture if some students found what I taught is a bit unclear?
- Numbers without prime factors in a set of positive relative density
- Montes nec non vorticum timores... - is this correct Latin?
- Am I breaking any design rules for the tile layout of this walls?
- Generating calendars for the next 10 years
- What purpose do chords serve in music?
- Can a double bar line begin a repeated section?
- Does everything have a infinitesimal amount of uncertainty?
- How would I create a clean version of this logo?
- Getting bounding box polygons from xmin, ymin, xmax, ymax in attribute table
- What is the most useful information to recover from an apocalypse?
- Why did only Steve Rogers have a physical change after injecting the super soldier serum?
- US: Is there any subject that it is unlawful to teach in schools?
- Leaving company, should I write personalized farewell emails?
- Big Bang and the Pauli Exclusion Principle
- Does the Taliban government of Afghanistan have a written legal code?
- Rollerskating or rollerblading and falling into water, drowning dangers
- Kollmorgen TBM frameless motor material
- Is my investment safe if the broker/bank I'm using goes into insolvency?
- How to get the embedded filename of a GPG encrypted file
- How does CA have standing in CA vs Activision Blizzard?

Question feed