
Express JS-Form data

Forms are an integral part of the web.

To get started with forms, we will first install the *body-parser*(for parsing JSON and url-encoded data) and *multer*(for parsing multipart/form data) middleware.

To install the *body-parser* and *multer*, go to your terminal and use –

npm install body-parser multer

After importing the body parser and multer, we will use the body-parser for parsing json and x-www-form-urlencoded header requests, while we will use multer for parsing multipart/form-data.

Replace your **index.js** file contents with the following code –

```
var express = require('express');
var bodyParser = require('body-parser');
var multer = require('multer');
var upload = multer();
var app = express();

//Read form saved as form.pug
app.get('/', function(req, res){
  res.render('form');
});

//Creating a view
app.set('view engine', 'pug');
app.set('views', './views');

// for parsing application/json
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
```

```
//form-urlencoded for parsing multipart/form-data
```

```
app.use(upload.array());
```

```
app.use(express.static('public'));
```

```
app.post('/', function(req, res){
```

```
  console.log(req.body);
```

```
  res.send("Request is received!");
```

```
});
```

```
app.listen(3000);
```

Let us create an html form to test this out. Create a new view called form.pug with the following code –

```
html
```

```
  head
```

```
    title Form Tester
```

```
  body
```

```
    form(action = "/", method = "POST")
```

```
      div
```

```
        label(for = "say") Say:
```

```
        input(name = "say" value = "Hi")
```

```
      br
```

```
      div
```

```
        label(for = "to") To:
```

```
        input(name = "to" value = "Express forms")
```

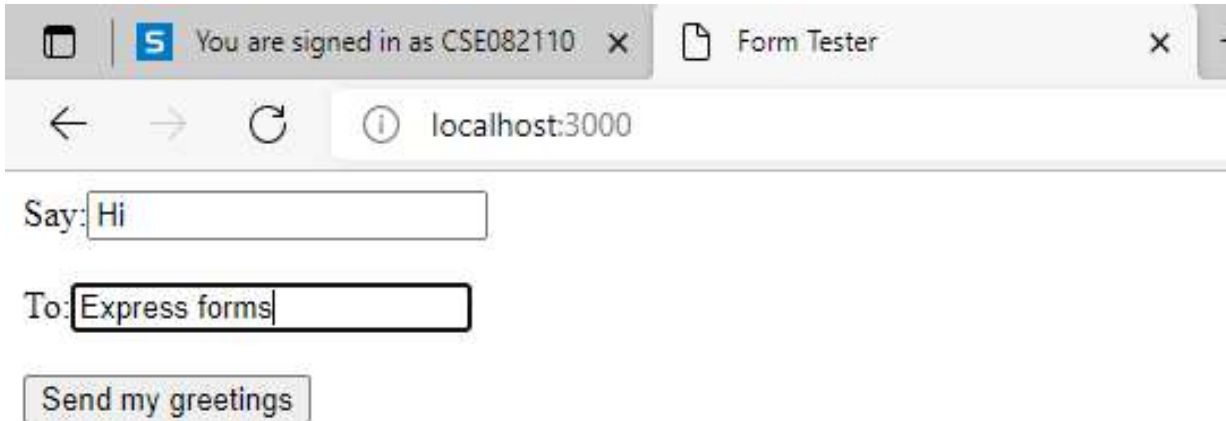
```
      br
```

```
      button(type = "submit") Send my greetings
```

Run your server using the following.

nodemon index.js

Now go to localhost:3000/ following is displayed in browser---



Browser tabs: You are signed in as CSE082110, Form Tester

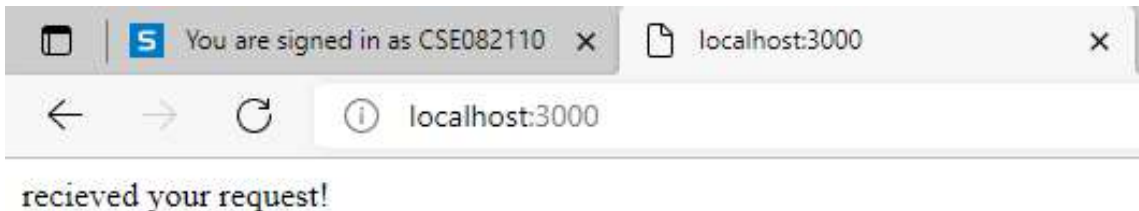
Address bar: localhost:3000

Form fields:

Say:

To:

Now, fill the form and submit the following response is generated.



Browser tabs: You are signed in as CSE082110, localhost:3000

Address bar: localhost:3000

Response: recieved your request!

In console; it will show you the body of your request as a JavaScript object as in the following screenshot –



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\shrug\express\express_eg> node index.js
{ say: 'hello', to: 'java' }
{ say: 'Hi', to: 'Express forms' }
PS C:\Users\shrug\express\express_eg> node index.js
{ say: 'Hi', to: 'Express forms' }
```

Why pug?

Pug in node.js is a template engine that uses case sensitive syntax to generate html, in other words it returns a string of html rendered as per data specified in a pug file.

We can say that pug is the middleman who plays a role to convert the injected data and translate it into html syntax.

Pug is a template engine for Node and for the browser. It compiles to HTML and has a simplified syntax, which can make you more productive and your code more readable.

Pug makes it easy both to write reusable HTML, as well as to render data pulled from a database or API
