

Controller

Receive request as body

Receive data from form data

```
const express = require("express");
const app = express();
const port = 3000;

// Middleware to parse JSON and URL-encoded form data
app.use(express.json());
app.use(express.urlencoded({ extended: true })); //

// Serve a basic HTML form for testing
app.get("/", (req, res) => {
  res.send(`
    <form method="POST" action="/submit">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name">
      <br>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email">
      <br>
      <input type="submit" value="Submit">
    </form>
  `);
});

// POST route to handle form submission
app.post("/submit", (req, res) => {
  // Access form data from req.body
  const name = req.body.name;
  const email = req.body.email;

  res.send(`Received data: Name - ${name}, Email - ${email}`);
});

// Start the server
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

Receive data from json

```
const express = require("express");  
const app = express();  
const port = 3000;
```

```
// Middleware to parse JSON and URL-encoded form data  
app.use(express.json());  
app.use(express.urlencoded({ extended: true })); //
```

```
// POST route to handle form submission  
app.post("/submit", (req, res) => {  
  // Access form data from req.body  
  const name = req.body.name;  
  const email = req.body.email;  
  
  res.status(200).json({ name, email });  
});  
  
// Start the server  
app.listen(port, () => {  
  console.log(`Server is running on port ${port}`);  
});
```

express.json() Middleware

Purpose: This middleware is used to parse JSON data sent in the request body.

Use Case: It's commonly used when clients (such as web browsers or other applications) send JSON data to your server, often in the context of APIs or when data is submitted via AJAX requests with a JSON payload.

How It Works: When this middleware is applied using `app.use(express.json())`, it parses the JSON data from the request body and makes it available in `req.body` as a JavaScript object for easy access and manipulation by your route handlers.

`express.urlencoded()` Middleware:

Purpose: This middleware is used to parse URL-encoded form data sent in the request body.

Use Case: It's used when clients submit HTML forms with data in a format similar to query strings (e.g., `name=value&name2=value2`). This middleware is essential for handling standard HTML form submissions.

How It Works: When applied with `app.use(express.urlencoded())`, it parses the URL-encoded data from the request body and makes it available in `req.body` as a JavaScript object for easy access and manipulation by your route handlers.