

Express.js json Middleware

The `express.json()` middleware is a built-in middleware function in Express.js that is used to **parse incoming JSON request bodies**.

1. Basic Usage

Use `express.json()` before handling requests that contain JSON data.

```
const express = require('express');
const app = express();
// Middleware to parse JSON data
app.use(express.json());
app.post('/data', (req, res) => {
  res.json({ receivedData: req.body });
});
app.listen(3000, () => console.log('Server running on port 3000'));
```

➤ Example Request:

POST /data
Content-Type: application/json

```
{
  "name": "John Doe",
  "email": "john@example.com"
}
```

➤ Parsed Output (`req.body`):

```
{
  "name": "John Doe",
  "email": "john@example.com"
}
```

2. Limiting Payload Size

By default, Express allows requests **up to 100KB**. To prevent large payloads (DoS attacks), set a custom limit:

```
app.use(express.json({ limit: '10kb' })); // Limits JSON payload to 10KB
```

➤ If the payload exceeds the limit, Express will return:

413 Payload Too Large

3. Handling Malformed JSON

If a client sends **invalid JSON**, Express will throw an error. You can catch this using an error-handling middleware:

```
app.use(express.json());
app.use((err, req, res, next) => {
  if (err instanceof SyntaxError) {
    return res.status(400).json({ message: 'Invalid JSON format' });
  }
  next();
});
```

➤ **Example Request (Malformed JSON):**

POST /data

Content-Type: application/json

```
{
  "name": "John Doe",
  "email": "john@example.com" // Missing closing brace
```

➤ **Response:**

```
{
  "message": "Invalid JSON format"
}
```

4. Combining with `express.urlencoded()`

If your app handles both **JSON and form data**, use:

```
app.use(express.json()); // For JSON requests
app.use(express.urlencoded({ extended: true })); // For URL-encoded form data
```

➤ This allows Express to handle:

- **JSON requests** (Content-Type: application/json)
 - **Form submissions** (Content-Type: application/x-www-form-urlencoded)
-

5. Summary

Feature	Example
Parse JSON data	<code>app.use(express.json())</code>
Set payload size limit	<code>app.use(express.json({ limit: '10kb' }))</code>
Handle invalid JSON	Custom error-handling middleware
Support both JSON & forms	<code>app.use(express.json());</code> <code>app.use(express.urlencoded({ extended: true }));</code>

Would you like more examples? ☐