

Express.js urlencoded Middleware

The `express.urlencoded()` middleware is a built-in middleware function in Express.js that is used to **parse incoming URL-encoded form data** from `application/x-www-form-urlencoded` requests.

1. Basic Usage

The middleware must be used before handling requests that contain URL-encoded data.

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

❑ Example Request:

A form submission using `application/x-www-form-urlencoded`:

POST /submit

Content-Type: `application/x-www-form-urlencoded`

`name=John+Doe&email=john%40example.com`

❑ Parsed Output (`req.body`):

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

2. `extended: true` vs `extended: false`

The `extended` option determines how URL-encoded data is parsed:

- **`extended: false`** → Uses **querystring module** (Only supports simple key-value pairs)
- **`extended: true`** → Uses **qs module** (Supports nested objects & arrays)

`app.use(express.urlencoded({ extended: true }));` // Recommended

❑ Example with Nested Data (`extended: true`)

POST /submit
Content-Type: application/x-www-form-urlencoded

user[name]=John&user[email]=john@example.com

➤ **Parsed Output (req.body with extended: true):**

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

➤ **Parsed Output (req.body with extended: false):**

```
{
  "user[name]": "John",
  "user[email]": "john@example.com"
}
```

➤ **Recommendation:**

✓ Use extended: true to allow more complex objects.

3. Handling Form Data in Express

You can use `express.urlencoded()` to process form submissions.

Example: Handling an HTML Form

```
<form action="/submit" method="POST">
  <input type="text" name="username" placeholder="Enter username">
  <input type="password" name="password" placeholder="Enter password">
  <button type="submit">Submit</button>
</form>
```

Express Handling Form Data

```
app.use(express.urlencoded({ extended: true }));
app.post('/submit', (req, res) => {
  console.log(req.body); // { username: 'John', password: '123456' }
  res.send(`Received: ${req.body.username}`);
});
```

4. Limiting Payload Size

To prevent **DoS attacks** from large payloads, set a size limit:

```
app.use(express.urlencoded({ extended: true, limit: '10kb' })); // Limit to 10KB
```

5. Combining with express.json()

For handling **both JSON and form data**, use:

```
app.use(express.json());  
app.use(express.urlencoded({ extended: true }));
```

Summary

Feature	Example
Parse form data	<code>app.use(express.urlencoded({ extended: true }));</code>
Support nested objects	<code>extended: true</code>
Simple key-value pairs only	<code>extended: false</code>
Limit request size	<code>app.use(express.urlencoded({ extended: true, limit: '10kb' }));</code>
Handle both JSON & forms	<code>app.use(express.json());</code> <code>app.use(express.urlencoded({ extended: true }));</code>

Would you like any more details?