

`app.use([path,] callback [, callback...])`

Mounts the specified [middleware](#) function or functions at the specified path: the middleware function is executed when the base of the requested path matches `path`.

Arguments

Argument	Description	Default
<code>path</code>	<p>The path for which the middleware function is invoked; can be any of:</p> <ul style="list-style-type: none">• A string representing a path.• A path pattern.• A regular expression pattern to match paths.• An array of combinations of any of the above. <p>For examples, see Path examples.</p>	<code>'/'</code> (root path)



React

9.7 Handling Routes

EXPRESS Js

```
const app = express();
app.use('/', (req, res, next) => {
  console.log("Hello World");
  //res.send('<p>Welcome to Submit Details Page</p>');
  next();
});

app.use('/submit-details', (req, res, next) => {
  console.log("Second Middleware", req.url, req.method);
  res.send('<p>Welcome to Submit Details Page</p>');
});

app.use('/', (req, res, next) => {
  console.log("Second Middleware", req.url, req.method);
  res.send('<p>Welcome to Complete Coding</p>');
});
```

1. Order matters
2. Can not call `next()` after `send()`
3. “/” matches everything
4. Calling `res.send` implicitly calls `res.end()`



React

9.7 Handling Routes

EXPRESS Js

```
var express = require('express');  
var app = express();
```

HTTP method for which the middleware function applies.

```
app.get('/', function(req, res, next) {  
  next();  
})
```

Path (route) for which the middleware function applies.

The middleware function.

Callback argument to the middleware function, called "next" by convention.

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

HTTP **response** argument to the middleware function, called "res" by convention.

HTTP **request** argument to the middleware function, called "req" by convention.



React

9.7 Handling Routes

EXPRESS Js

```
const app = express();
```

```
app.post('/', (req, res, next) => {  
  console.log("Hello World");  
  //res.send('<p>Welcome to Submit Details Page</p>');  
  next();  
});
```

```
app.use('/submit-details', (req, res, next) => {  
  console.log("Second Middleware", req.url, req.method);  
  res.send('<p>Welcome to Submit Details Page</p>');  
});
```

```
app.use('/', (req, res, next) => {  
  console.log("Second Middleware", req.url, req.method);  
  res.send('<p>Welcome to Complete Coding</p>');  
});
```



Practise Set

Create a new project.

1. Install **nodemon** and **express**.
2. Add two **dummy middleware** that logs request path and request method respectively.
3. Add a **third middleware** that returns a response.
4. Now add **handling** using two more middleware that handle path **/**, a request to **/contact-us** page.
5. **Contact us** should return a form with name and email as input fields that submits to **/contact-us** page also.
6. Also handle **POST** incoming request to **/contact-us** path using a separate middleware.

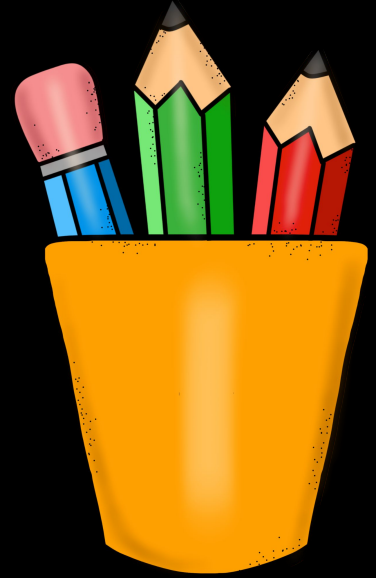




React

Revision

1. What is Express.js
2. Need of Express.js
3. Installing Express.js
4. Adding Middleware
5. Sending Response
6. Express DeepDive
7. Handling Routes







10. Express.js DeepDive

1. Parsing Requests
2. Express Router
3. Adding 404
4. Common Paths
5. Adding HTML Files
6. Serving HTML Files
7. Using File Helper





React

10.1 Parsing Requests

EXPRESS Js

```
// External Modules
```

```
const express = require('express');
```

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

```
const app = express();
```

```
app.use(bodyParser.urlencoded());
```

```
app.get('/details', (req, res, next) => {
```

```
  console.log("Hello World");
```

```
  res.send(`
```

```
    <h1>Enter Your Details:</h1>
```

```
    <form action="/submit-details" method="POST">
```

```
      <input type="text" name="username" placeholder="Enter your name"><br>
```

```
      <br><input type="submit" value="Submit">
```

```
    </form>
```

```
  `);
```

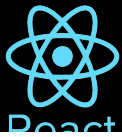
```
});
```

```
app.post('/submit-details', (req, res, next) => {
```

```
  console.log(req.url, req.method, req.body);
```

```
  res.redirect('/');
```

```
});
```



EXPRESS Js



Stays

Experiences

Airbnb your home



Where

Search destinations

Check in

Add dates

Check out

Add dates

Who

Add guests



Icons



Amazing pools



Farms



Amazing views



Surfing



Ski-in/out



Rooms



Cabins



Countryside



Treehouses



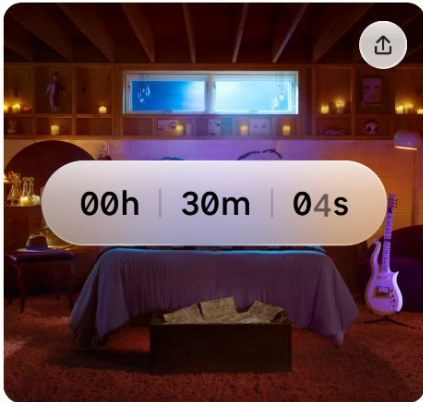
OMG!



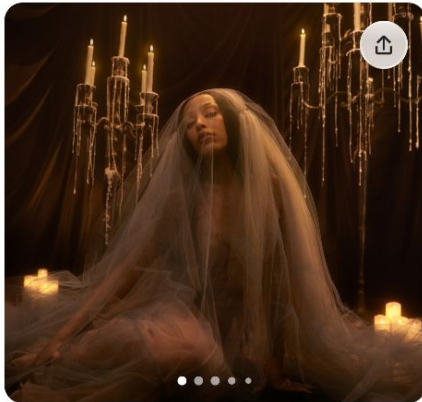
Luxe



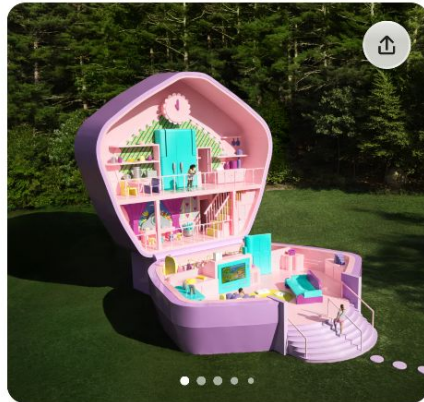
Beach



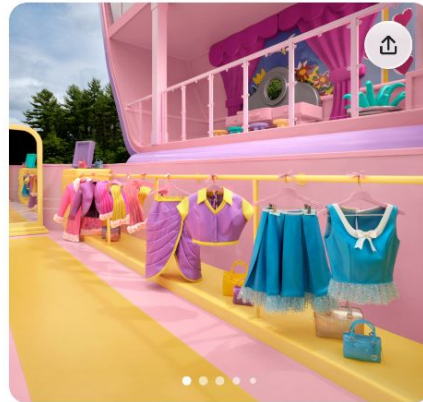
Stay in Prince's Purple Rain house
Hosted by Wendy And Lisa
₹564 per guest



Join the Living Room Session with Do...
Hosted by Doja Cat
Coming 8 October



Sleepover at Polly Pocket's Compact
Hosted by Polly Pocket
Sold out



Playdate at Polly Pocket's Compact
Hosted by Polly Pocket
Sold out



React

10.2 Express Router

EXPRESS Js

node > air-bnb >  app.js > ...

```
1  // External Modules
2  const express = require('express');
3  // Local Modules
4  const hostRouter = require('./routes/host');
5  const userRouter = require('./routes/user');
6
7  const app = express();
8
9  app.use(express.urlencoded({extended:true}));
10 app.use(hostRouter);
11 app.use(userRouter);
12
13 const PORT = 3000;
14 app.listen(PORT, () => {
15   console.log(`Server running at http://localhost:${PORT}/`);
16 });
```