

Lecture 5.HTTPS Methods in Nodejs Using Express Js

09 April 2025 19:00

Summary of HTTP Methods in Node.js

HTTP methods define the type of operation the client wants to perform on the server. In Node.js (with or without frameworks like Express), these are commonly used to create APIs.

1. GET

- **Used to retrieve data** from the server.
- **Example:** Fetching user details.

```
js
CopyEdit
const express = require('express');
const app = express();
app.get('/users', (req, res) => {
  res.send('Fetching all users...');
});
```

2. POST

- **Used to send data** to the server to create a new resource.
- **Example:** Creating a new user.

```
js
CopyEdit
app.use(express.json()); // to parse JSON body
app.post('/users', (req, res) => {
  const user = req.body;
  res.send(`User created: ${JSON.stringify(user)}`);
});
```

3. PUT

- **Used to update** an existing resource completely.
- **Example:** Updating all info of a user.

```
js
CopyEdit
app.put('/users/:id', (req, res) => {
  const id = req.params.id;
  const updatedUser = req.body;
  res.send(`User with ID ${id} updated to: ${JSON.stringify(updatedUser)}`);
});
```

4. PATCH

- **Used to partially update** a resource.
- **Example:** Updating only the email of a user.

```
js
CopyEdit
app.patch('/users/:id', (req, res) => {
  const id = req.params.id;
```

```
res.send(`User with ID ${id} partially updated`);
});
```



5. DELETE

- **Used to delete** a resource from the server.
- **Example:** Deleting a user by ID.

```
js
CopyEdit
app.delete('/users/:id', (req, res) => {
  const id = req.params.id;
  res.send(`User with ID ${id} deleted`);
});
```



Summary Table

Method	Description	Used For
GET	Retrieve data	Reading resources
POST	Create a new resource	Submitting forms, user signup
PUT	Update full resource	Replace data completely
PATCH	Update part of a resource	Change specific fields
DELETE	Remove a resource	Deleting records/data