

Other HTTP Methods

3 min

[HTTP Protocol](#) defines a number of different method verbs with many use cases. So far, we have been using the GET request which is probably the most common of all. Every time your browser loads an image, it is making a GET request for that file!

This course will cover three other important HTTP methods: PUT, POST, and DELETE. Express provides

Preview: Docs Loading link description

[methods](#)

for each one: `app.put()`, `app.post()`, and `app.delete()`.

PUT

Preview: Docs XMLHttpRequest is a built-in browser object that allows HTTP requests to be made in JavaScript. It is basically used to fetch data from APIs.

[requests](#)

are used for updating existing resources. In our Express Yourself machine, a PUT request will be used to update the name or emoji of an expression already saved in our database. For this reason, we will need to include a unique identifier as a route parameter to determine which specific resource to update.

Instructions

1. Checkpoint 1 Passed

1.

For now, open a PUT `/expressions/:id` route with an empty `(req, res, next)` callback function. We will fully implement its functionality in the next exercise.

app.js

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

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

```
// Serves Express Yourself website
```

```
app.use(express.static('public'));
```

```
const { getElementById, seedElements } = require('./utils');
```

```
const expressions = [];
seedElements(expressions, 'expressions');

const PORT = process.env.PORT || 4001;

app.get('/expressions', (req, res, next) => {
  res.send(expressions);
});

app.get('/expressions/:id', (req, res, next) => {
  const foundExpression = getElementById(req.params.id, expressions);
  if (foundExpression) {
    res.send(foundExpression);
  } else {
    res.status(404).send();
  }
});

// Add your PUT route handler below:
app.put('/expressions/:id', (req, res, next) => {

})

app.listen(PORT, () => {
  console.log(`Listening on port ${PORT}`);
});
```