

## Deleting Old Expressions

9 min

DELETE is the HTTP method verb used to delete resources. Because DELETE routes delete currently existing data, their paths should usually end with a route parameter to indicate which resource to delete.

First, we get the index of the resource using `getIndexById()` from **utils.js**:

```
app.delete('/monsters/:id', (req, res, next) => {  
  const expressionIndex = getIndexById(req.params.id, monsters);
```

If the index is valid, we use `splice()` to remove the element at that index:

```
monsters.splice(expressionIndex, 1);
```

Express uses `.delete()` as its method for DELETE

Preview: Docs Loading link description

[requests](#)

.

Servers often send a 204 No Content status code if deletion occurs without error.

### Instructions

1. Checkpoint 1 Passed

#### 1.

Create a DELETE `/expressions/:id` route. It should send back a 404 response for a request with an invalid id, and it should delete the proper element from the expressions array and send a 204 status with a valid id.

To test your functionality, use the DELETE tab in the upper left. Select the ID to delete and send the request.

#### Hint

You can use `getIndexById` to find the index of the element to delete. `getIndexById` will return `-1` for a non-existent ID, and the proper index if it exists. Then you can use the [splice](#) method to remove the element.

**app.js**

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

// Serves Express Yourself website
app.use(express.static('public'));

const { getElementById, getIndexById, updateElement,
        seedElements, createElement } = 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();
    }
});

app.put('/expressions/:id', (req, res, next) => {
    const expressionIndex = getIndexById(req.params.id, expressions);
```

```

    if (expressionIndex !== -1) {
      updateElement(req.params.id, req.query, expressions);
      res.send(expressions[expressionIndex]);
    } else {
      res.status(404).send();
    }
  });

app.post('/expressions', (req, res, next) => {
  const receivedExpression = createElement('expressions', req.query);
  if (receivedExpression) {
    expressions.push(receivedExpression);
    res.status(201).send(receivedExpression);
  } else {
    res.status(400).send();
  }
});

// Add your DELETE handler below:
app.delete('/expressions/:id', (req, res, next) => {
  const expressionIndex = getIndexById(req.params.id, expressions);
  if (expressionIndex !== -1) {
    expressions.splice(expressionIndex, 1);
    res.status(204).send();
  } else {
    res.status(404).send();
  }
});

app.listen(PORT, () => {

```

```
    console.log(`Listening on port ${PORT}`);  
  });
```