

Adding Animals Routes

18 min

Congratulations, you have made our glorious Express Yourself Machine [fully operational!](#) Not only that, you now have all the tools you need to create a basic CRUD API! Time to practice your skills with a new set of routes.

You are going to add an additional set of functionality to our machine: Animal Mode! This will involve creating similar GET, POST, PUT, and DELETE routes.

Instructions

1. Checkpoint 1 Passed

1.

In your **app.js** file, Create a GET /animals route to return an array of all animals.

2. Checkpoint 2 Passed

2.

Create a GET /animals/:id route to respond with a single animal.

3. Checkpoint 3 Passed

3.

Create a PUT /animals/:id route to update an animal in animals and send back the updated animal.

4. Checkpoint 4 Passed

4.

Create a POST /animals route to add new animals to the animals and respond with the new animal.

5. Checkpoint 5 Passed

5.

Create a DELETE /animals/:id route to delete animals by ID.

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 animals = [];
seedElements(animals, 'animals');

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

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

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

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.get('/animals/:id', (req, res, next) => {  
  const foundAnimal = getElementById(req.params.id, animals);  
  if (foundAnimal) {  
    res.send(foundAnimal);  
  } 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.put('/animals/:id', (req, res, next) => {  
  const animalIndex = getIndexById(req.params.id, animals);  
  if (animalIndex !== -1) {  
    updateElement(req.params.id, req.query, animals);  
    res.send(animals[animalIndex]);  
  }  
});
```

```
    } 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();  
  }  
});
```

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

```
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.delete('/animals/:id', (req, res, next) => {  
  const animalIndex = getIndexById(req.params.id, animals);  
  if (animalIndex !== -1) {  
    animals.splice(animalIndex, 1);  
    res.status(204).send();  
  } else {  
    res.status(404).send();  
  }  
});
```

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