



**DEV ACADEMY**  
TE KURA HANGARAU  
O AOTEAROA

# Creating a Web Server with Express

Week 4, Day 1





# 1: Setup

```
1  import express from 'express'
2
3  const server = express()
4
5  // configure server
6  server.get('...')
7  server.post('...')
8
9  // run the server
10 server.listen('port', () => { /* ... */ })
```



## 2: Request/Response

req / res (request, response)

```
1 // http://localhost:3000/path/to/resource
2 server.get('/path/to/resource', (request, response) => {})
```

### 3: Responding with text/HTML with `res.send()`



We can send plain text:

```
1 // http://localhost:3000/path/to/resource
2 server.get('/path/to/resource', (req, res) => {
3   res.send('Hello World')
4 })
```

Browsers will also interpret HTML sent as a response:

```
1 // http://localhost:3000/path/to/resource
2 server.get('/path/to/resource', (req, res) => {
3   res.send('<h1 style="color:red;">Hello World</h1>')
4 })
```



## 4: Responding with JSON with `res.json()`

```
1 // http://localhost:3000/pets/dogData
2 server.get('/pets/dogData', (req, res) => {
3   // .json forms an object in JSON format
4   res.json({ name: 'Pups', number: 'many', cuteness: 'overload'})
5 })
```



## 5: Responding with a File with `res.sendFile()`

```
1  import * as Path from 'node:path'
2
3  ...
4
5  // http://localhost:3000/pets/cat
6  server.get('/pets/cat', (req, res) => {
7    // .sendFile requires an _absolute_ file path
8    // so we can use Path.join() or Path.resolve()
9    res.sendFile(Path.resolve('./public/cat.html'))
10 }
```



## 6: Using Query Parameters with `req.query`

```
1 // http://localhost:3000/path/to/resource?size=large
2 server.get('/path/to/resource', (req, res) => {
3   // req.query = { size: 'large' }
4   const size = res.query.size
5   // ^? 'large'
6
7   // ...
8 })
```



## 7: Using Route Parameters with `req.params`

```
1 // http://localhost:3000/pets/cat
2 server.get('/pets/:type', (req, res) => {
3   // req.params = { type: 'cat' }
4   const type = res.params.type
5   // ^? 'cat'
6
7   // ...
8 })
```





## 8: Posting data with HTTP POST and `req.body`

We send the data in JSON format:

```
1  {  
2    "colour": "ginger",  
3    "ears": "long"  
4  }
```

To access the posted data (`req.body`) in our route:

```
1  // http://localhost:3000/create-pet  
2  server.post('/create-pet', (req, res) => {  
3    console.log(req.body.colour req.body.ears)  
4  
5    // ...  
6  })
```