

Using Fetch

The `cors` module allows us to configure express to accept connections from certain domains.

```
npm install cors --save
```

The following code configures express to allow requests from anywhere. This code should be placed before the calls to `app.get()`.

```
const cors = require('cors');
app.use(cors())
app.use(function (req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
  next();
});
```

To configure express to allow JSON objects in the body of POST requests, we include the following code before the calls to `app.get()`.

```
app.use(express.json())
```

Examples

Client-side	Server-side
<pre>async function fetchData() { const url = "http://localhost:3000" const options = { method: "GET", } let response = await fetch(url, options) console.log(response) if (response.ok) { const data = await response.text() console.log(data) const mssg1 = document.querySelector("#message1") mssg1.innerHTML = data } }</pre>	<pre>app.get('/', (req, res) => { res.send('GET text data') })</pre>

Client-side	Server-side
<pre>async function fetchDataWithStatusCodes() { const url = "http://localhost:3000/fetch" const options = { method: "GET", } let response = await fetch(url, options) if (response.status == 201) { const text = await response.text() const mssg2 = document.querySelector("#message2") mssg2.innerHTML = text } }</pre>	<pre>app.get('/fetch', (req, res) => { res.status(201).send('GET text data with status code') })</pre>

Client-side	Server-side
<pre>async function fetchObjects() { const url = "http://localhost:3000/fetch/object" const options = { method: "GET", } let response = await fetch(url, options) if (response.status == 200) { const obj = await response.json() const mssg3 = document.querySelector("#message3") mssg3.innerHTML = obj.message } }</pre>	<pre>app.get('/fetch/object', (req, res) => { res.send({ message: "GET object" }) })</pre>

Client-side	Server-side
<pre>async function fetchToEndpointWithParameter() { const mssg5 = document.querySelector("#message5") const code = 7 const url = `http://localhost:3000/fetch/param/\${code}` const options = { method: "GET" } let response = await fetch(url, options) const obj = await response.json() if (response.status == 200) { mssg5.innerHTML = obj.message } else if (response.status == 401) { mssg5.innerHTML = "Error: " + obj.message } }</pre>	<pre>app.get('/fetch/param/:code', (req, res) => { const code = req.params.code if (code == 7) { res.send({ message: "GET using parameter" }) } else { res.status(401).send({ message: "Invalid request: " + code }) } })</pre>

Client-side	Server-side
<pre>async function fetchToEndpointWithQueryString() { const mssg6 = document.querySelector("#message6") const code = 7 const url = `http://localhost:3000/fetch/query?code=\${code}` const options = { method: "GET" } let response = await fetch(url, options) const obj = await response.json() if (response.status == 200) { mssg6.innerHTML = obj.message } else if (response.status == 401) { mssg6.innerHTML = "Error: " + obj.message } }</pre>	<pre>app.get('/fetch/query', (req, res) => { const code = req.query.code if (code == 7) { res.send({ message: "GET using query string" }) } else { res.status(401).send({ message: "Invalid request: " + code }) } })</pre>

Client-side

```
async function fetchToEndpointUsingPostMethod() {  
  
  const mssg7 = document.querySelector("#message7")  
  
  const url = "http://localhost:3000/fetch"  
  
  const data = {  
    code: 7  
  }  
  
  const options = {  
    method: "POST",  
    headers: { "Content-Type": "application/json" },  
    body: JSON.stringify(data)  
  }  
  
  let response = await fetch(url, options)  
  const obj = await response.json()  
  
  if (response.status == 200) {  
    mssg7.innerHTML = obj.message  
  }  
  else if (response.status == 401) {  
    mssg7.innerHTML = "Error: " + obj.message  
  }  
}
```

Server-side

```
// add app.use(express.json())  
  
app.post('/fetch', async (req, res) => {  
  const code = req.body.code  
  
  if (code == 7) {  
    res.send({  
      message: "POST with body object"  
    })  
  }  
  else {  
    res.status(401).send({  
      message: "Invalid request: " + code  
    })  
  }  
})
```

Client-side

```
async function fetchDataFromForm() {

  const mssg8 = document.querySelector("#message8")
  const input = document.querySelector("#input8")

  const url = "http://localhost:3000/fetch"

  const data = {
    code: input.value
  }

  const options = {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(data)
  }

  let response = await fetch(url, options)

  if (response.status == 200) {
    const obj = await response.json()
    mssg8.innerHTML = obj.message
  }
  else if (response.status == 401) {
    const obj = await response.json()
    mssg8.innerHTML = "Error: " + obj.message
  }
  else if (response.status == 400) {
    mssg8.innerHTML = "Server error"
  }
}

document.querySelector("#button8").addEventListener("click",
fetchUsingDataFromForm)
```

Server-side

```
// add app.use(express.json())

app.post('/fetch', async (req, res) => {
  const code = req.body.code
  console.log(code)

  if (code == 7) {
    res.send({
      message: "POST with body object"
    })
  }
  else if (code == 5) {
    res.status(401).send({
      message: "unauthorized: " + code
    })
  }
  else {
    res.status(400).send()
  }
})
```