Springboard

Node Files Solution

« Back to Homepage

Step 1

Step 2

Step 3

Node Files Solution

Download solution

Step 1

step1.js

```
const fs = require('fs');
const process = require('process');

/** read file at path and print it out. */

function cat(path) {
  fs.readFile(path, 'utf8', function(err, data) {
    if (err) {
      console.error(`Error reading ${path}: ${err}`);
      process.exit(1);
    } else {
      console.log(data);
    }
  });
}

cat(process.argv[2]);
```

Step 2

step2.js

```
const fs = require('fs');
const process = require('process');
const axios = require('axios');

/** read file at path and print it out. */

function cat(path) {
  fs.readFile(path, 'utf8', function(err, data) {
    if (err) {
      console.error(`Error reading ${path}: ${err}`);
      process.exit(1);
    } else {
      console.log(data);
    }
  });
}

/** read page at URL and print it out. */

async function webCat(url) {
  try {
    let resp = await axios.get(url);
    console.log(resp.data);
  } catch (err) {
    console.error(`Error fetching ${url}: ${err}`);
    process.exit(1);
  }
}

let path = process.argv[2];

if (path.slice(0, 4) === 'http') {
  webCat(path);
} else {
  cat(path);
}
```

Step 3

step3.js

```
const fs = require('fs');
const process = require('process');
const axios = require('axios');

/** handle output: write to file if out given, else print */

function handleOutput(text, out) {
  if (out) {
    fs.writeFile(out, text, 'utf8', function(err) {
      if (err) {
        console.error(`Couldn't write ${out}: ${err}`);
        process.exit(1);
      }
    });
  } else {
    console.log(text);
  }
}

/** read file at path and print it out. */

function cat(path, out) {
  fs.readFile(path, 'utf8', function(err, data) {
    if (err) {
      console.error(`Error reading ${path}: ${err}`);
      process.exit(1);
    } else {
      handleOutput(data, out);
    }
  });
}

/** read page at URL and print it out. */

async function webCat(url, out) {
  try {
    let resp = await axios.get(url);
    handleOutput(resp.data, out);
  } catch (err) {
    console.error(`Error fetching ${url}: ${err}`);
    process.exit(1);
  }
}

let path;
let out;

if (process.argv[2] === '--out') {
  out = process.argv[3];
  path = process.argv[4];
} else {
  path = process.argv[2];
}

if (path.slice(0, 4) === 'http') {
  webCat(path, out);
} else {
  cat(path, out);
}
```