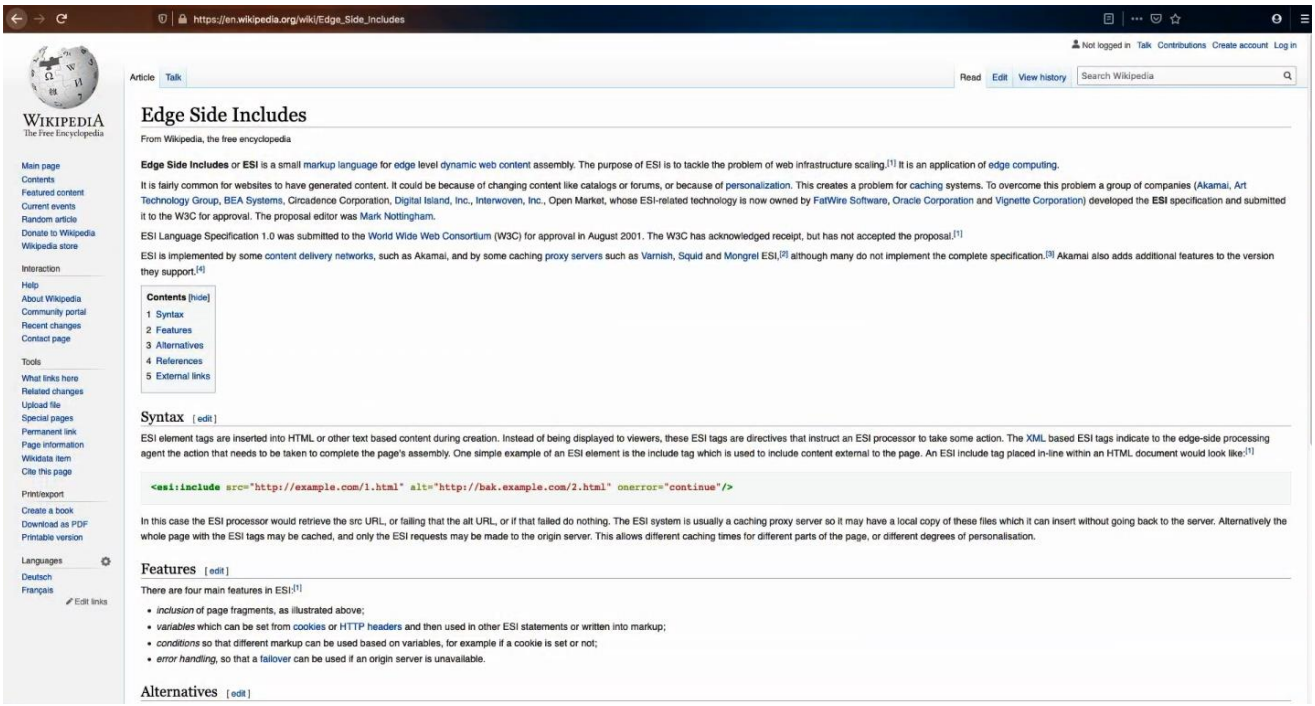
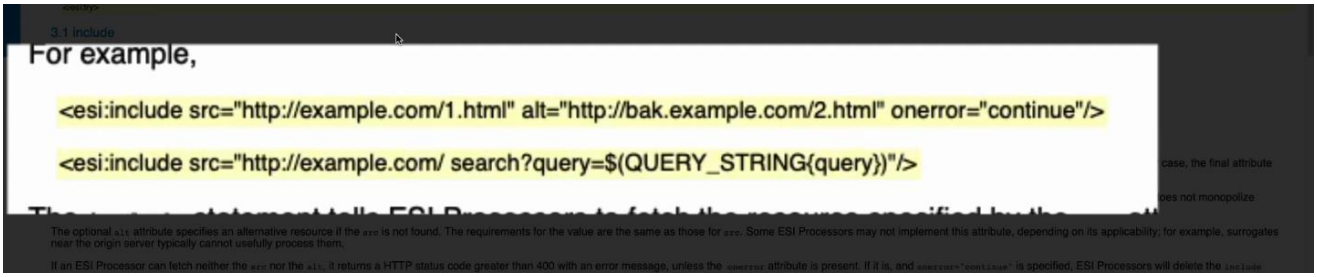
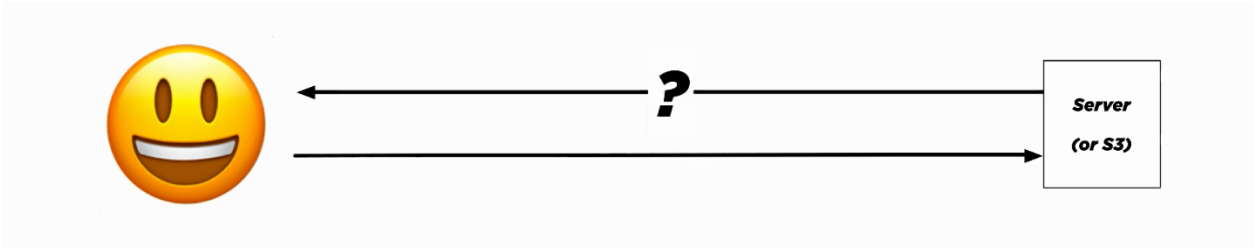
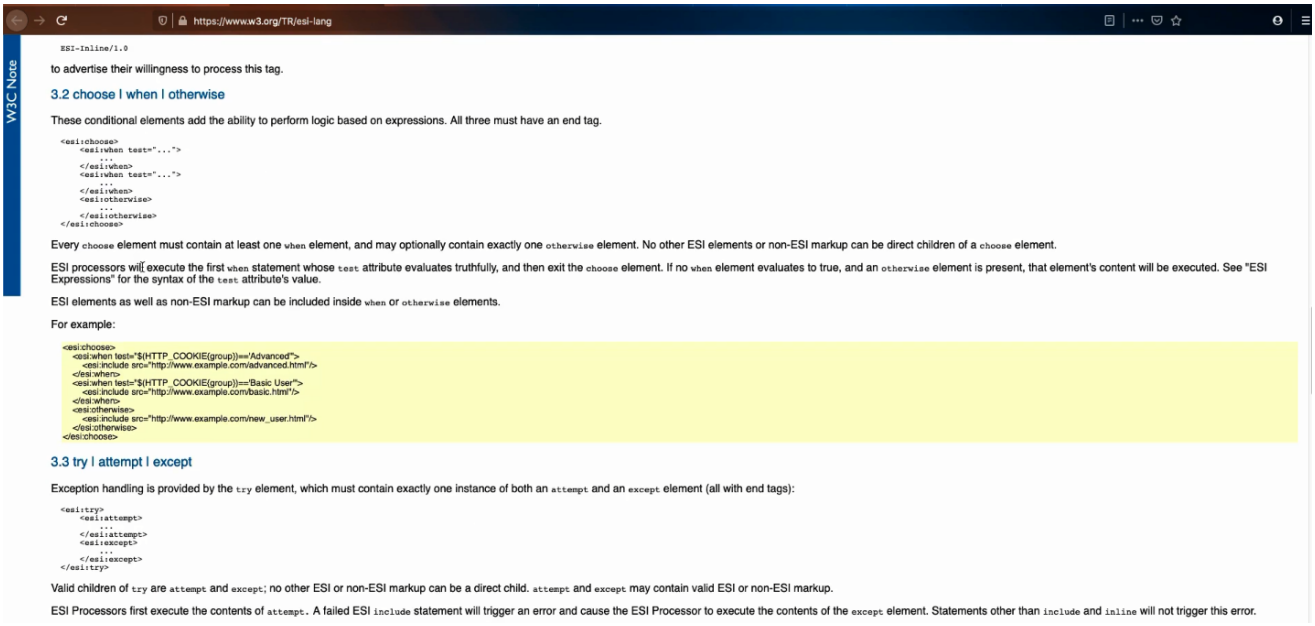
Learn how to use Edge Side Includes (ESI) in your Micro-Frontend solutions.

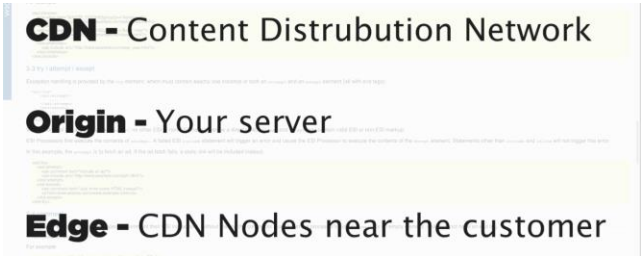




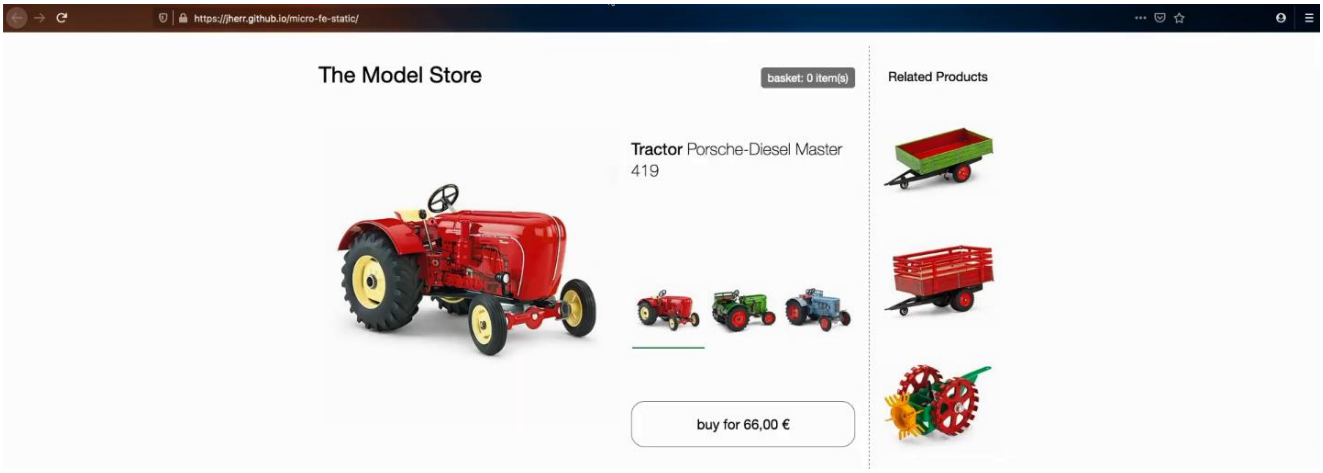
See https://www.w3.org/TR/esi-lang for details of the ESI Includes specification

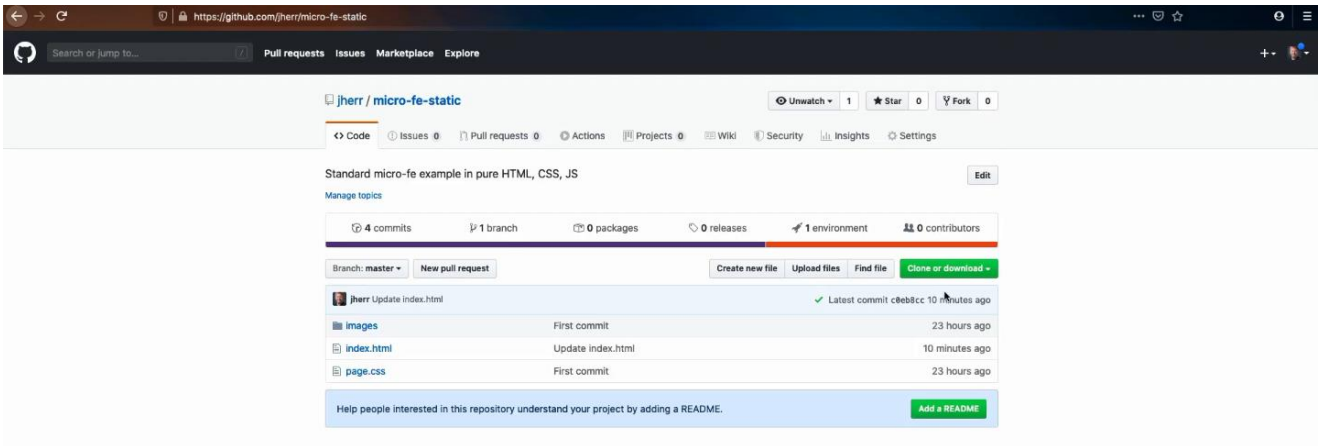


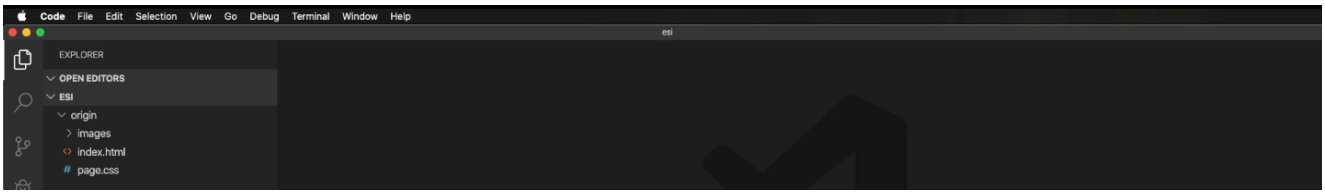


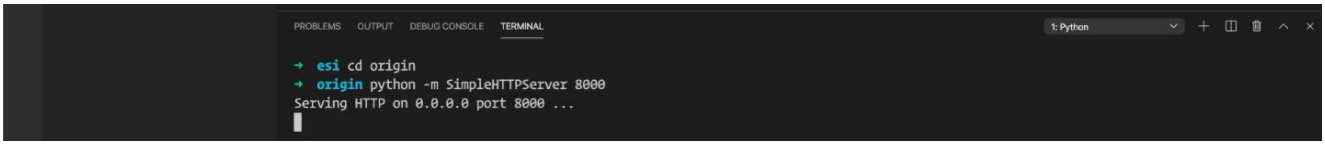


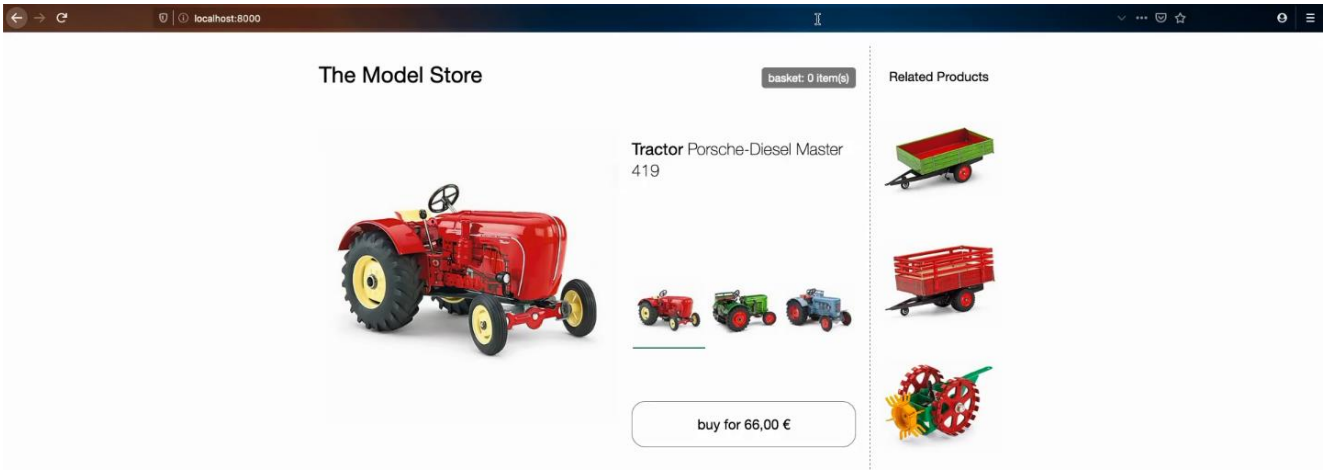
CDN provides a caching layer for your **Origin** servers called the **Edge**. ESI's are handled by the CDN

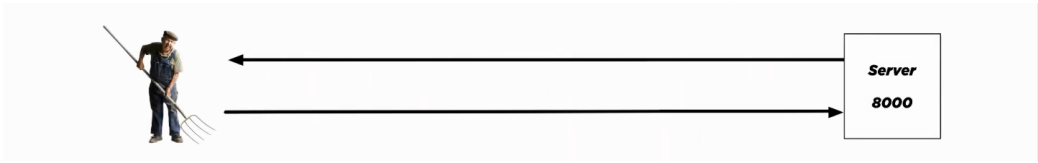


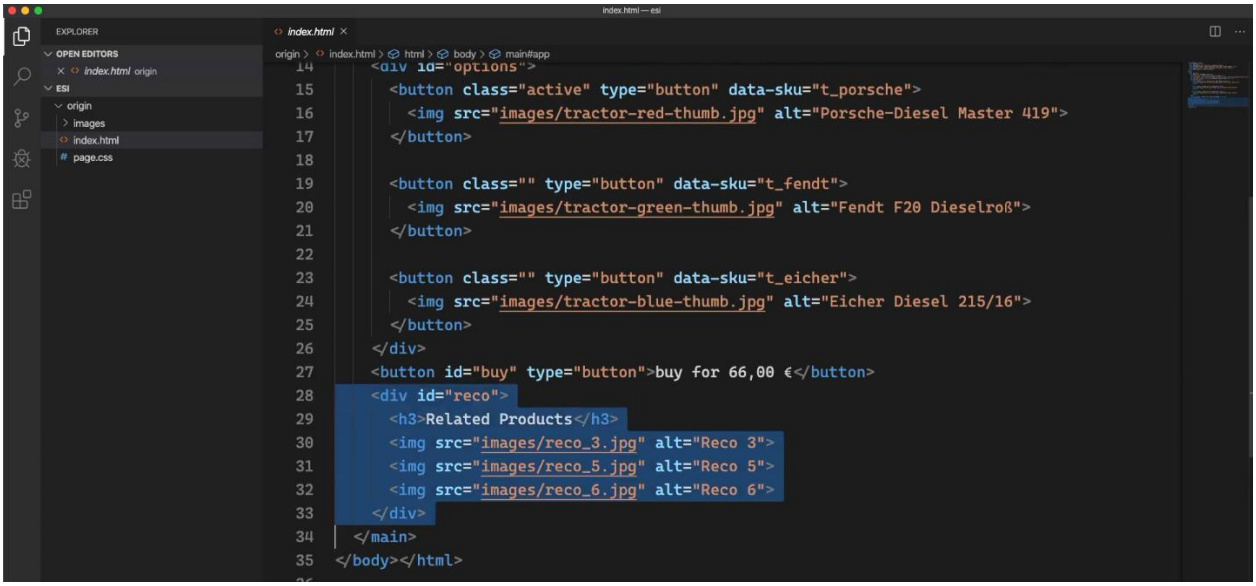










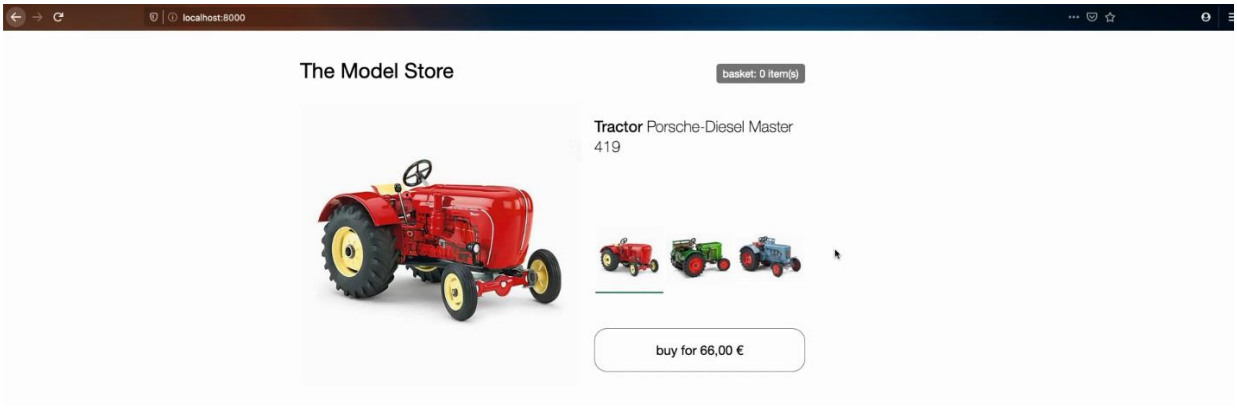


What we want to do is to take the **recommendations HTML section** out, put it in a new HTML pages within the components folder and put it back in the original page using an ESI tag.

```
<div id="reco">
  <h3>Related Products</h3>
  <img src="images/reco_3.jpg" alt="Reco 3">
  <img src="images/reco_5.jpg" alt="Reco 5">
  <img src="images/reco_6.jpg" alt="Reco 6">
</div>
```

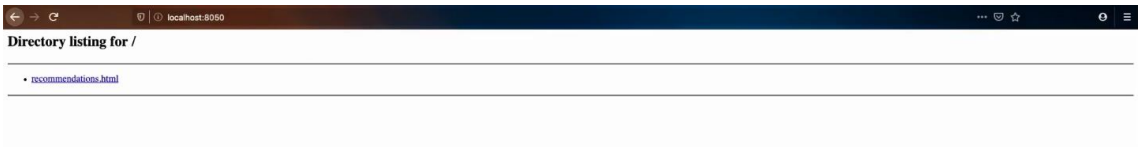
```
      <div id="options">
        <button class="active" type="button" data-sku="t_porsche">
          <img src="images/tractor-red-thumb.jpg" alt="Porsche-Diesel Master 419">
        </button>

        <button class="" type="button" data-sku="t_fendt">
          <img src="images/tractor-green-thumb.jpg" alt="Fendt F20 Dieselroß">
        </button>

        <button class="" type="button" data-sku="t_eicher">
          <img src="images/tractor-blue-thumb.jpg" alt="Eicher Diesel 215/16">
        </button>
      </div>
      <button id="buy" type="button">buy for 66,00 €</button>
      <esi:include src="http://localhost:8050/recommendations.html"></esi:include>
    </main>
  </body></html>
```

We are going to run the recommendations page out on port **8050** and then use the **ESI include** to get it from there

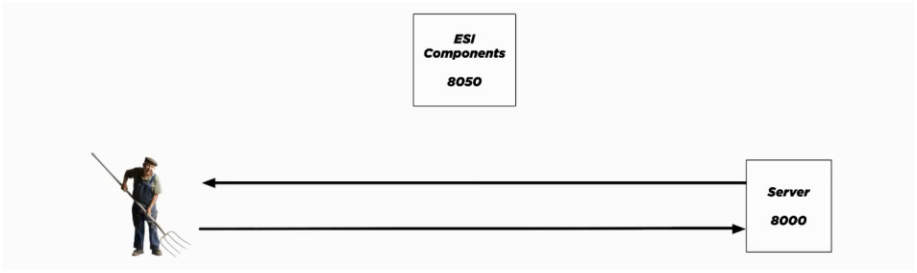



```
      <div id="options">
        <button class="active" type="button" data-sku="t_porsche">
          <img src="images/tractor-red-thumb.jpg" alt="Porsche-Diesel Master 419">
        </button>

        <button class="" type="button" data-sku="t_fendt">
          <img src="images/tractor-green-thumb.jpg" alt="Fendt F20 Dieselroß">
        </button>

        <button class="" type="button" data-sku="t_eicher">
          <img src="images/tractor-blue-thumb.jpg" alt="Eicher Diesel 215/16">
        </button>
      </div>
      <button id="buy" type="button">buy for 66,00 €</button>
      <esi:include src="http://localhost:8050/recommendations.html"></esi:include>
    </main>
  </body></html>
```

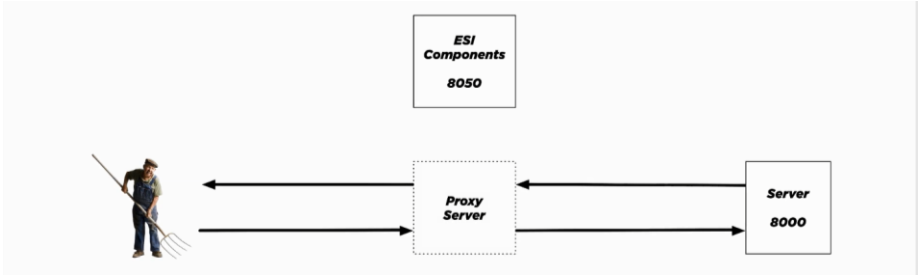
```
→ esi cd components
→ components python -m SimpleHTTPServer 8050
Serving HTTP on 0.0.0.0 port 8050 ...
```



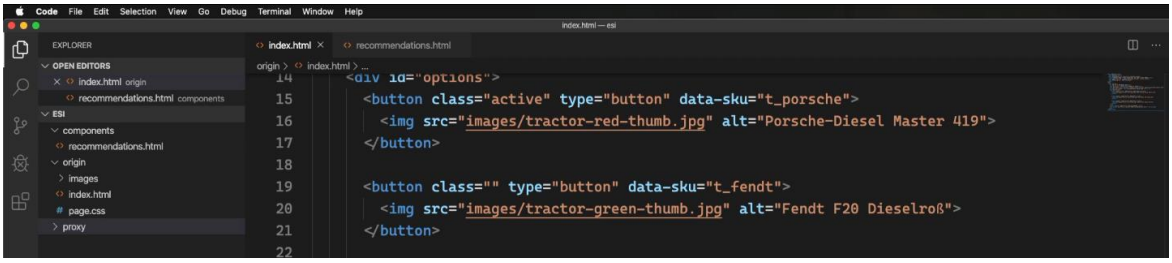
Note that there is no custom styling here, the host page is the one providing the styling



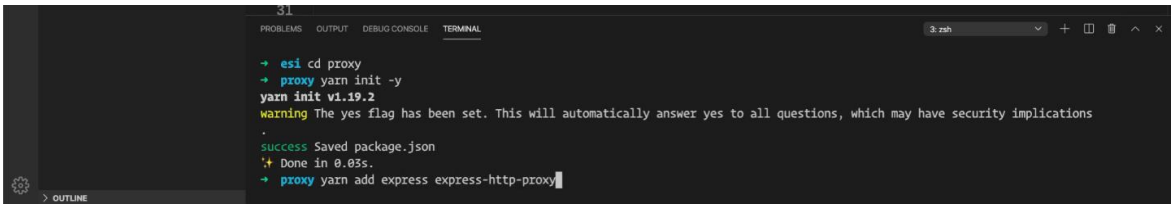
We need something between the 2 servers to resolve the ESI Include tags for us

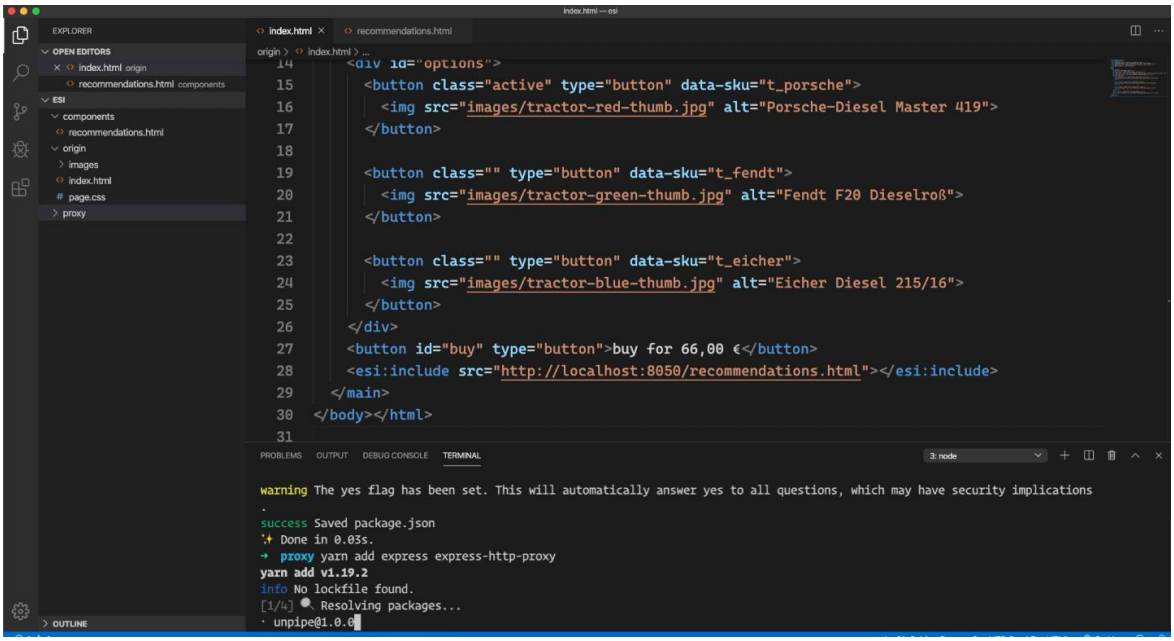


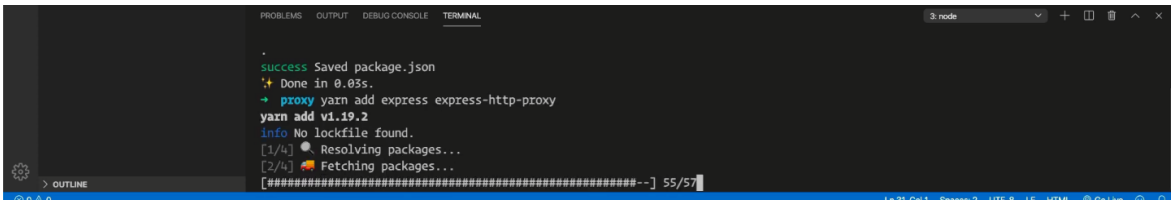
The CDN will be acting as our proxy server that helps to resolve the ESI include tags before serving the host page content

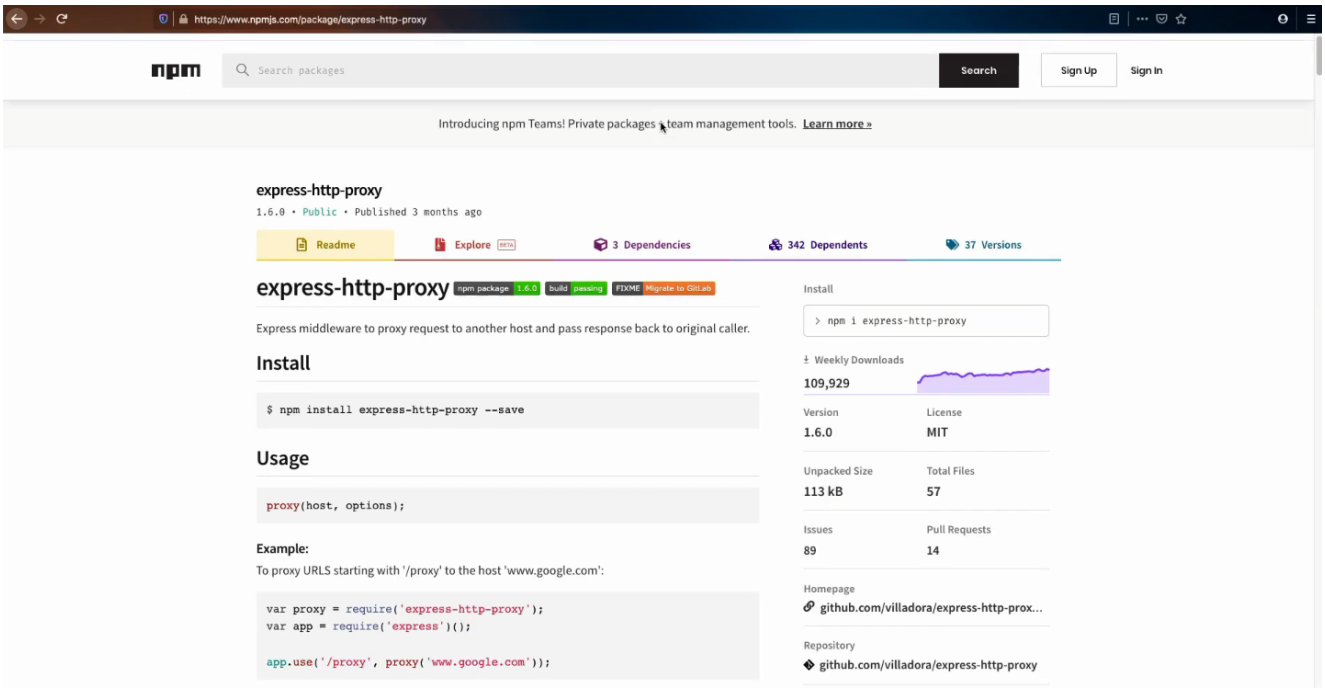


We will create a proxy folder where we will create a NodeJS Express Proxy Server

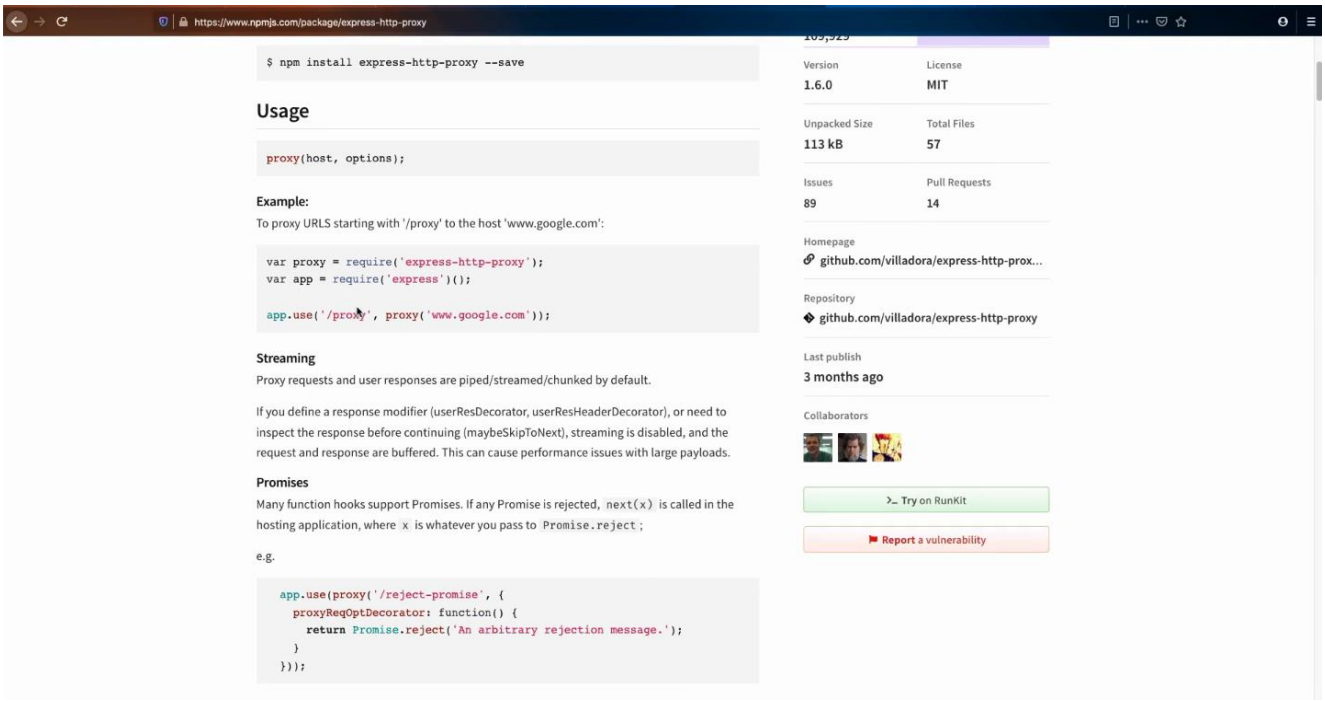


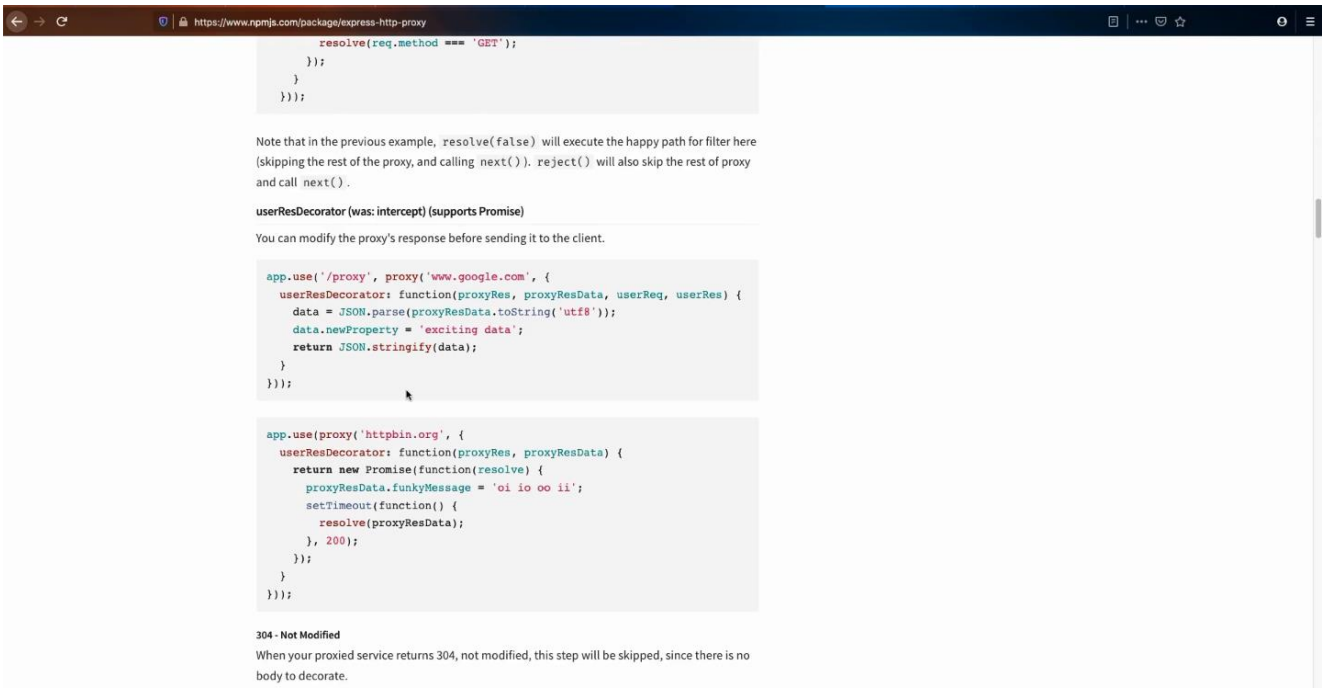




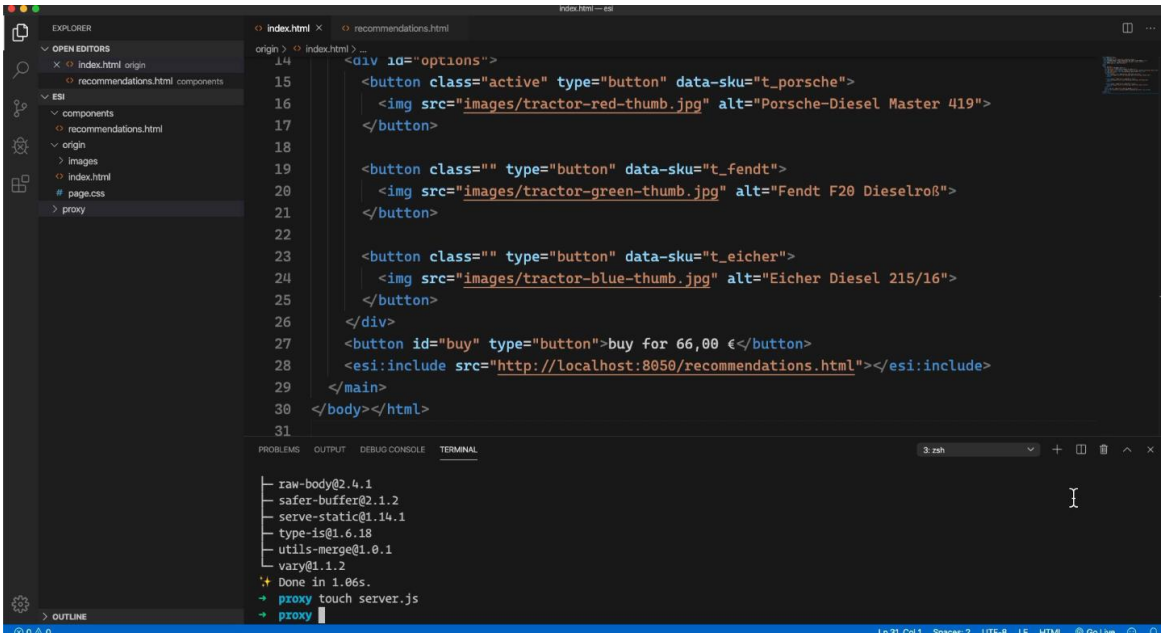


We can use the **express-http-proxy** module to run a proxy server that can help resolve the ESI Include tags in our HTML

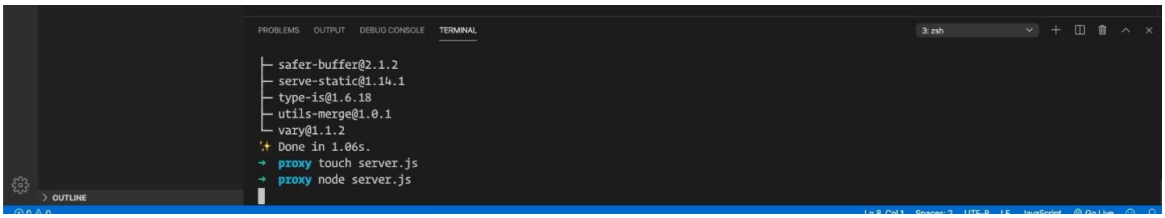


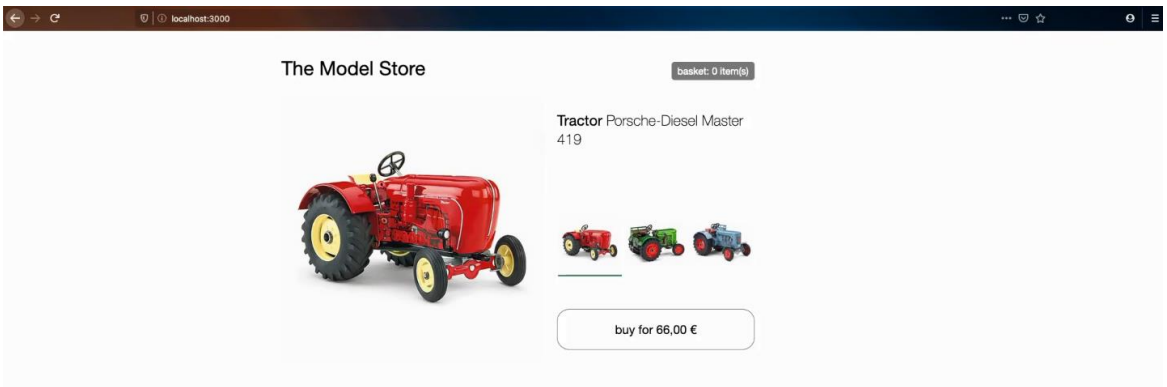


We can use the use decorator function to resolve the ESI Include tags within our proxy server before passing it on

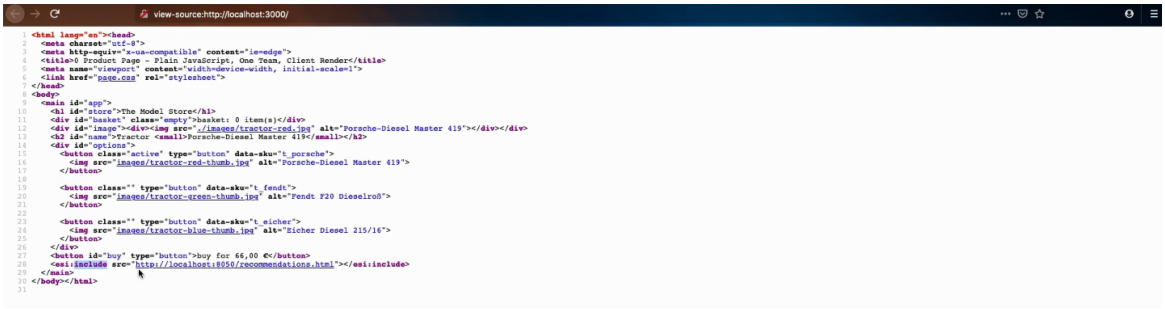


We then create our server.js file and run the proxy server out of port 8000. We will initially let everything pass through.

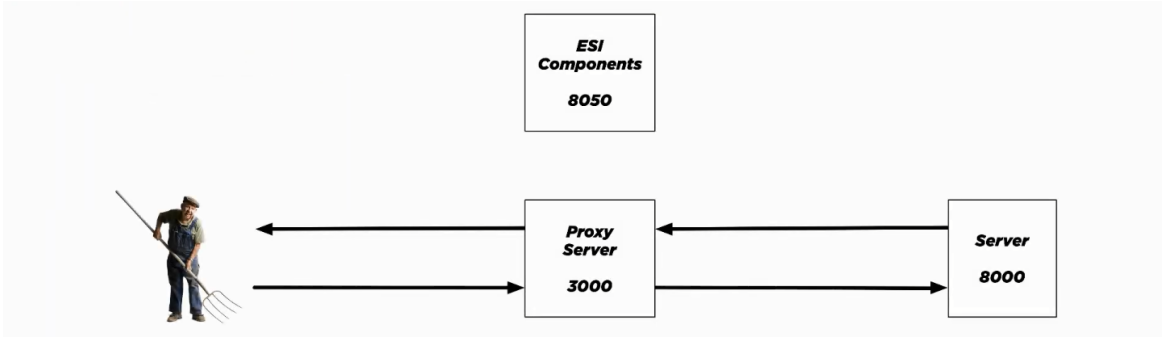




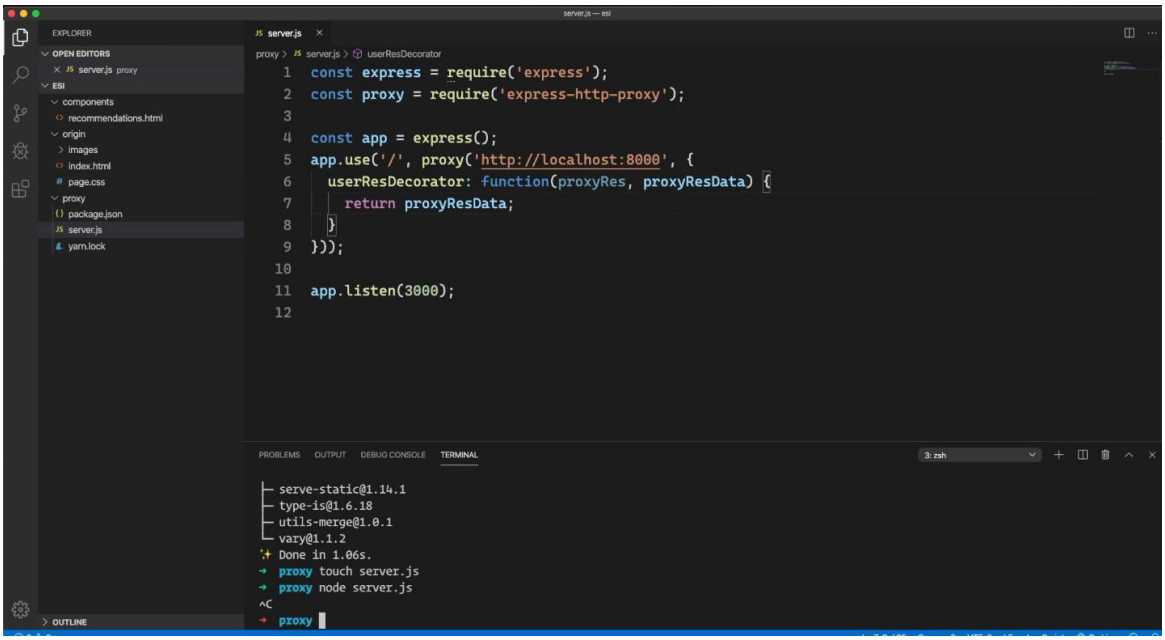
The host page looks the same as before, the recommendations HTML code is not being resolved or displayed



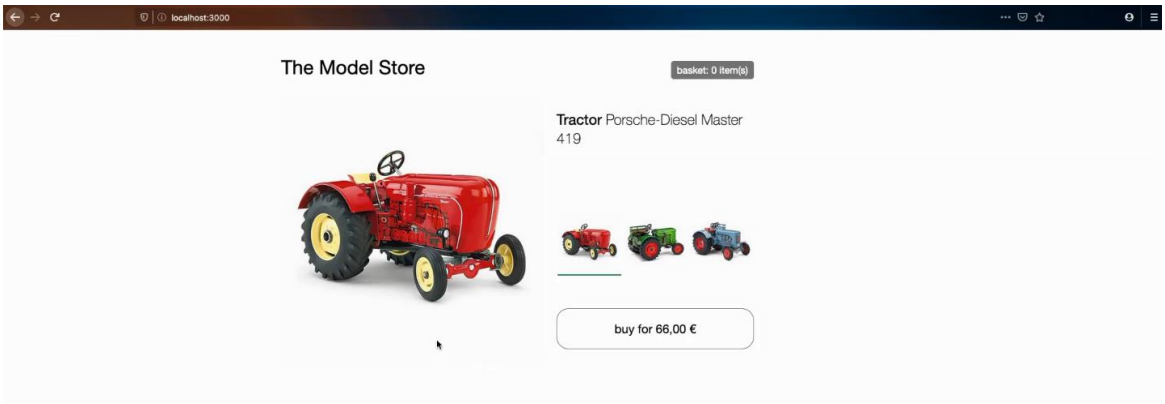
We now have the ESI Include tag passing through unresolved



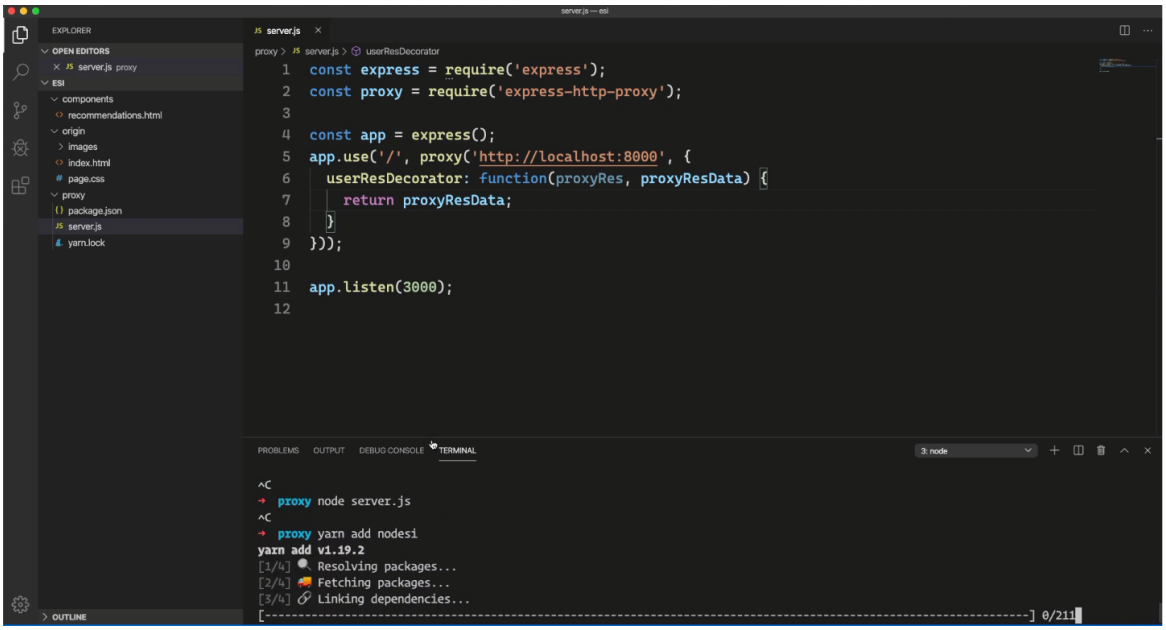
We need to put the code to resolve the ESI Include tags for the recommendations HTML code



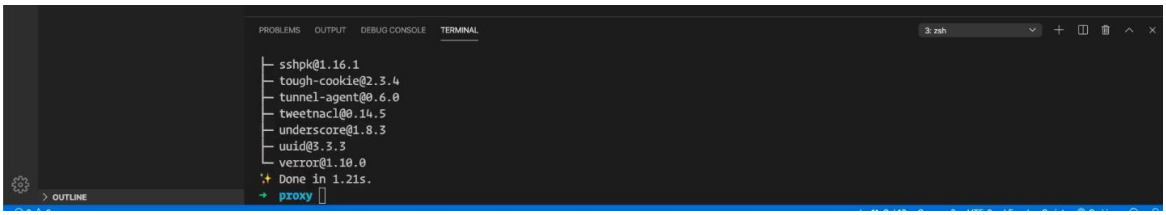
We can now set up the *user response decorator function* **userResDecorator()** discussed in the NPM page for the **express-http-proxy** and just pass things through for now.

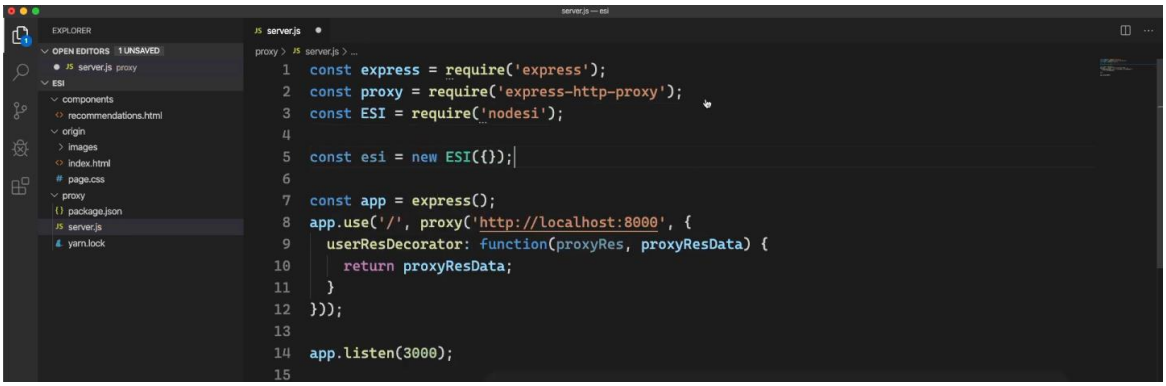


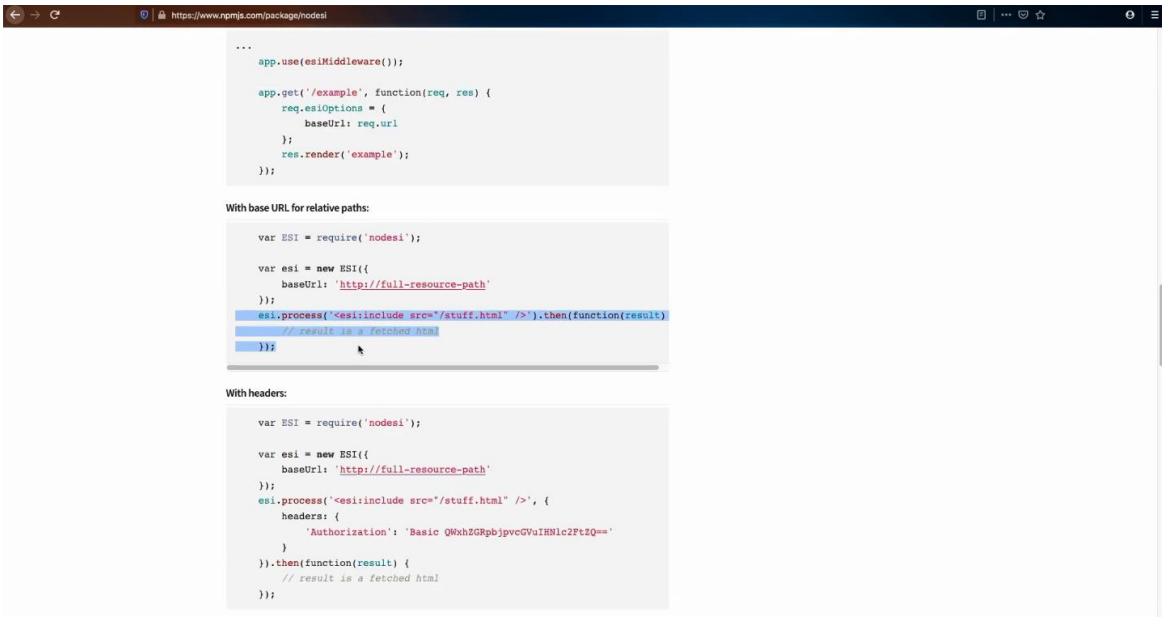
Things still work as before.

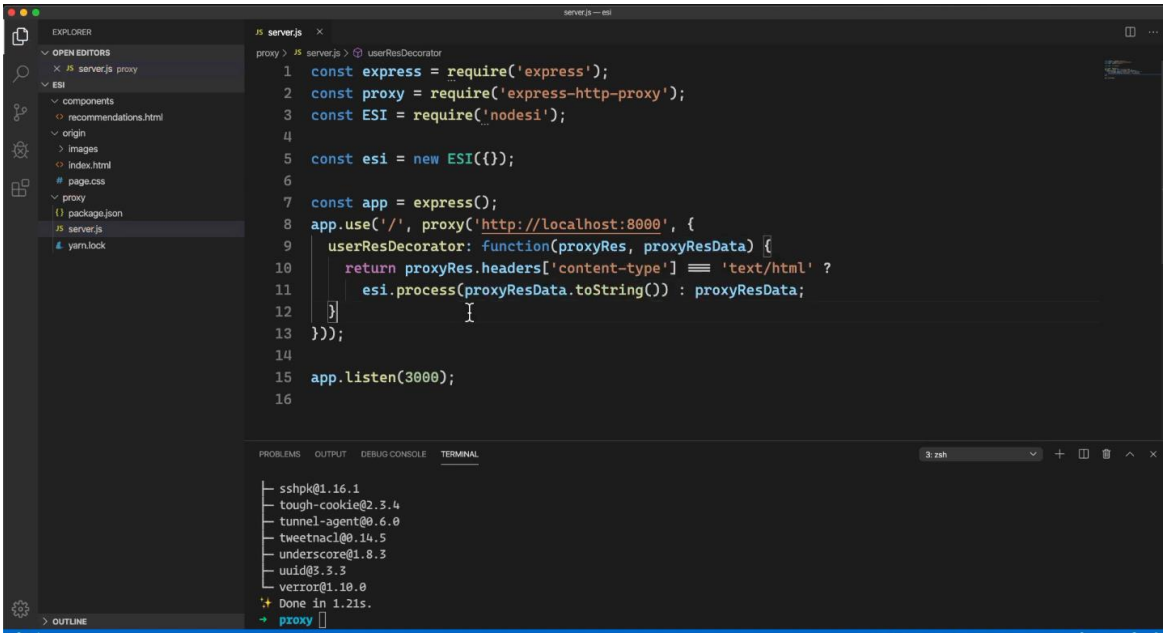


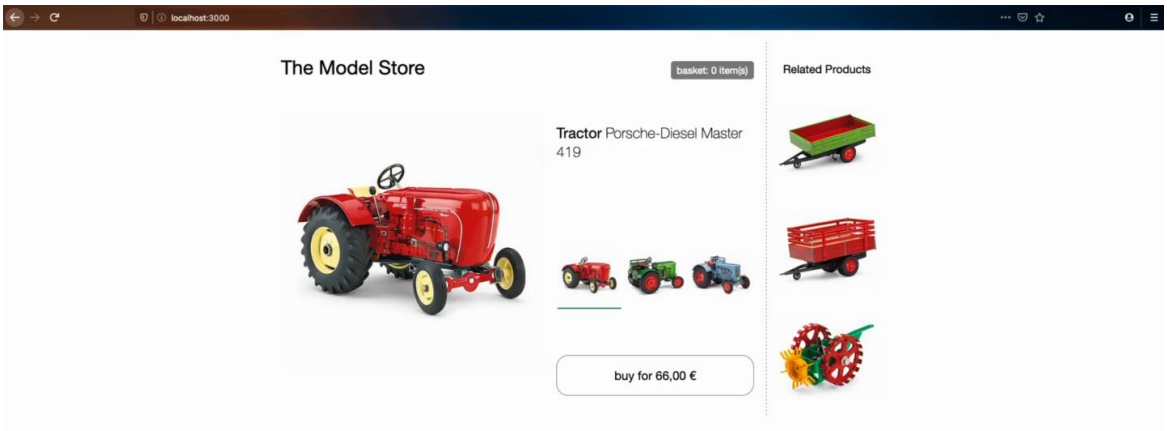
We then add the **nodesi** package using **yarn add nodesi**, the module that does ESI parsing and also handles those requests for us.











Now we have the ESI Include tag for the recommendations HTML resolved correctly and we see the recommendations

view-source:http://localhost:3000/

```
<html lang="en"><head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <title>0 Product Page - Plain JavaScript, One Team, Client Render</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="page.css" rel="stylesheet">
</head>
<body>
  <main id="app">
    <h1 id="store">The Model Store</h1>
    <div id="basket" class="empty">basket: 0 item(s)</div>
    <div id="image"><div><img src="./images/tractor-red.jpg" alt="Porsche-Diesel Master 419"></div></div>
    <h2 id="name">Tractor <small>Porsche-Diesel Master 419</small></h2>
    <div id="options">
      <button class="active" type="button" data-sku="t_porsche">
        <img src="images/tractor-red-thumb.jpg" alt="Porsche-Diesel Master 419">
      </button>

      <button class="" type="button" data-sku="t_fendt">
        <img src="images/tractor-green-thumb.jpg" alt="Fendt F20 Dieselroß">
      </button>

      <button class="" type="button" data-sku="t_eicher">
        <img src="images/tractor-blue-thumb.jpg" alt="Eicher Diesel 215/16">
      </button>
    </div>
    <button id="buy" type="button">buy for 66,00 €</button>
    <div id="reco">
  <h3>Related Products</h3>
  <img src="images/reco_3.jpg" alt="Reco 3">
  <img src="images/reco_5.jpg" alt="Reco 5">
  <img src="images/reco_6.jpg" alt="Reco 6">
</div>

  </main>
</body></html>
```

Code File Edit Selection View Go Debug Terminal Window Help

recommendations.html — esi

EXPLORER

OPEN EDITORS 1 UNSAVED
- server.js proxy
- recommendations.html components

ESI
- components
  - recommendations.html
- origin
  - images
  - index.html
  - page.css
- proxy
  - package.json
  - server.js
  - yarn.lock

server.js | recommendations.html

components > recommendations.html > div#reco > h3

```
<div id="reco">
  <h3>Related Products Hey there</h3>
  <img src="images/reco_3.jpg" alt="Reco 3">
  <img src="images/reco_5.jpg" alt="Reco 5">
  <img src="images/reco_6.jpg" alt="Reco 6">
</div>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 3: node

```
→ proxy node server.js
```

localhost:3000

The Model Store

basket: 0 item(s)

Related Products Hey there

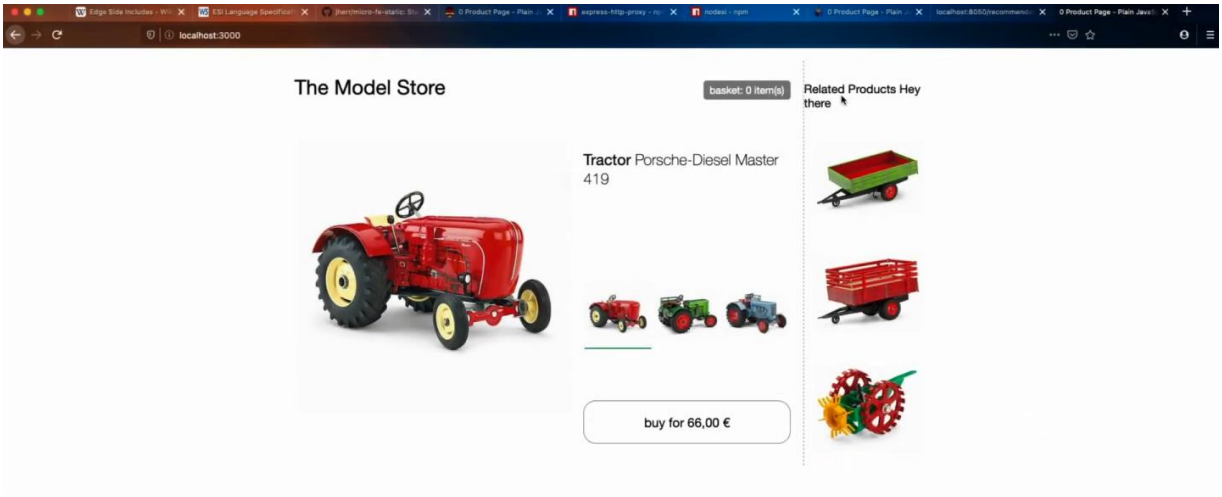
Tractor Porsche-Diesel Master 419

buy for 66,00 €

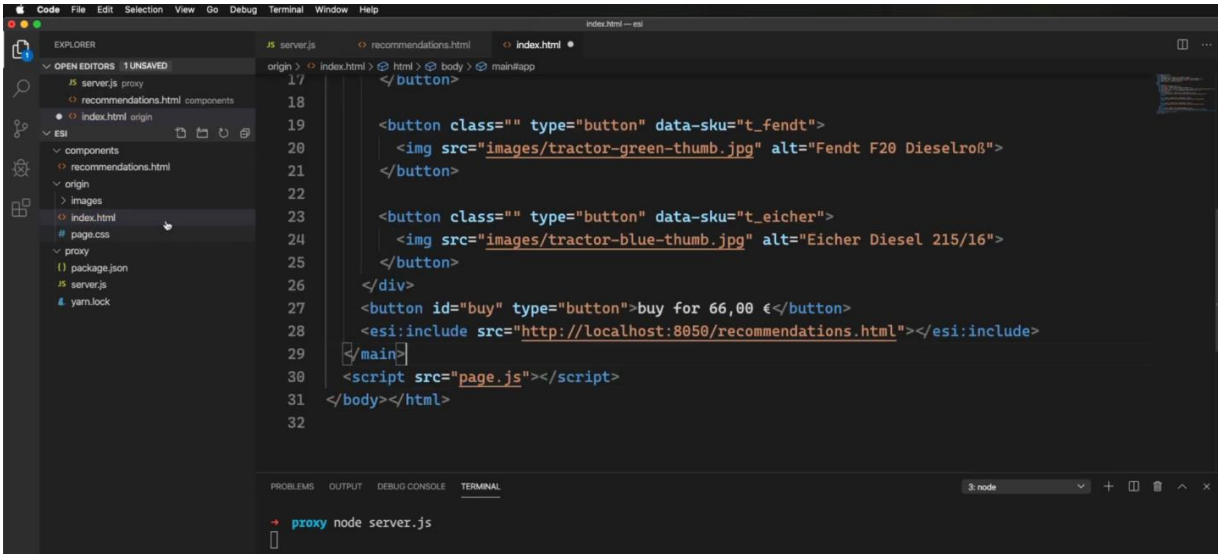
ESI Components 8050

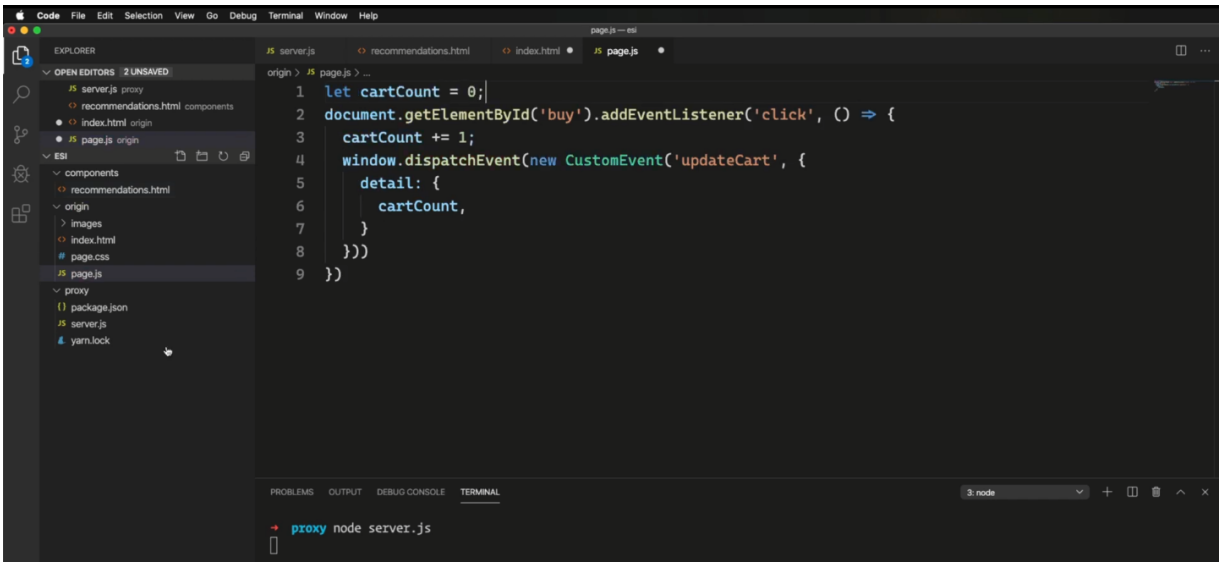
Proxy Server 3000

Server 8000

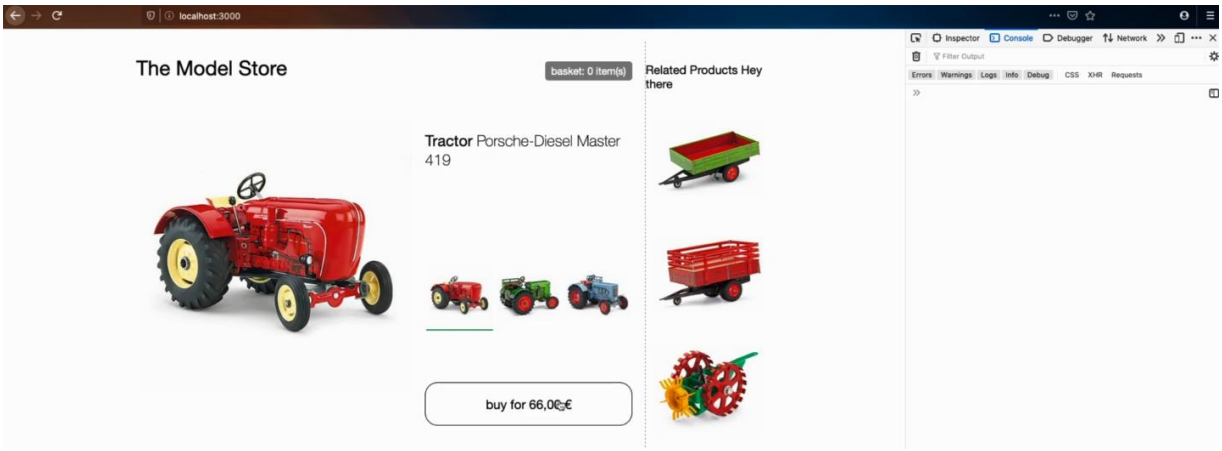


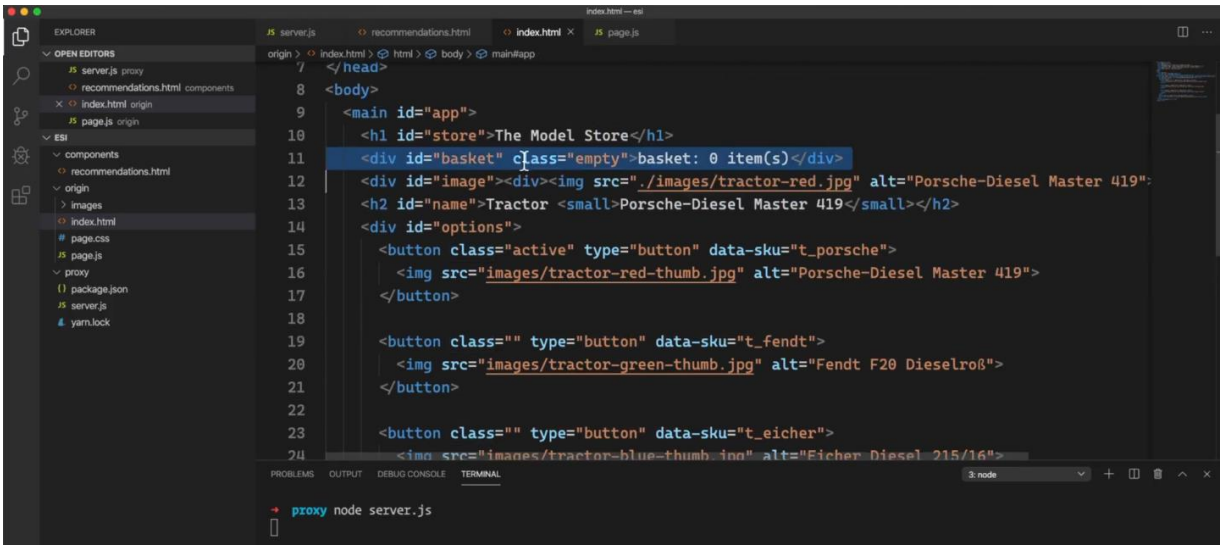
Let us now show how to make the code in the original page interact with those injected ESI components



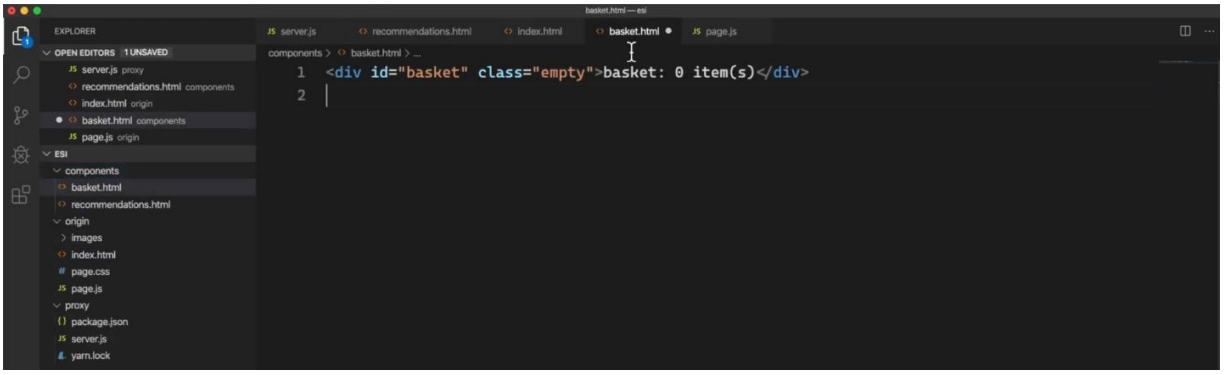


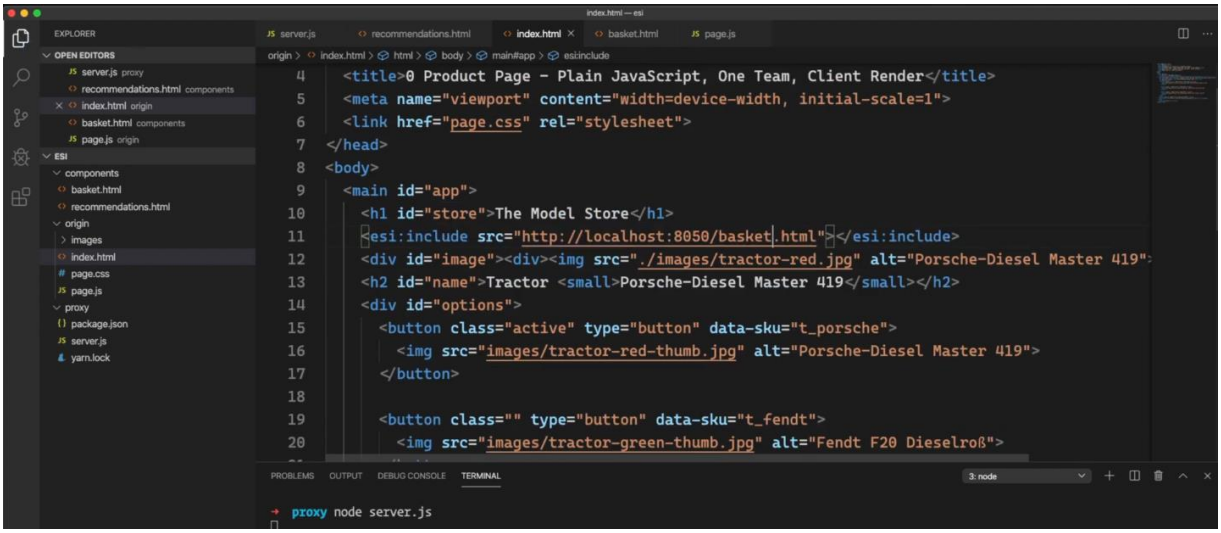
We start by creating some code for the buy button interactivity on the page in the **page.js** file. This code will add 1 to the **cartCount** value and put it into a **custom event on the window** saying that there is a new cart value.

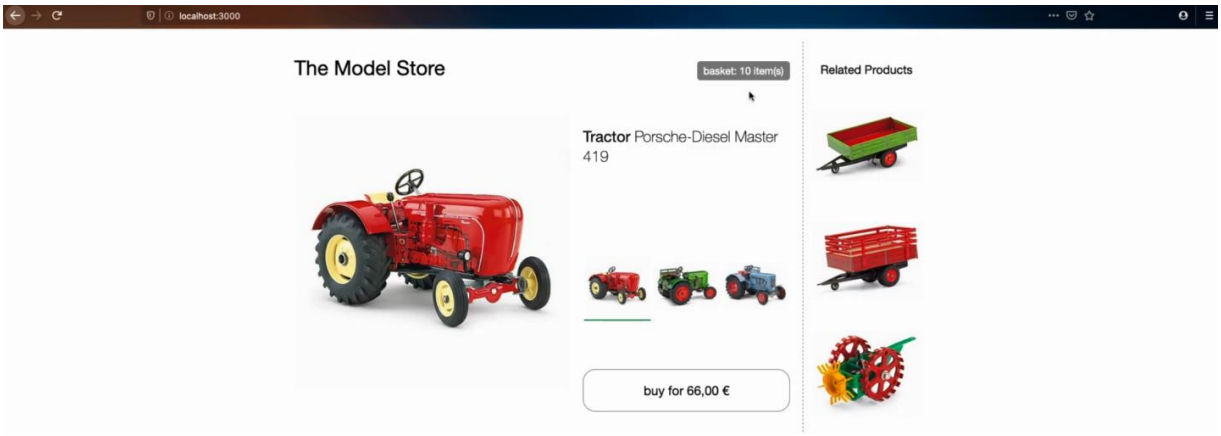


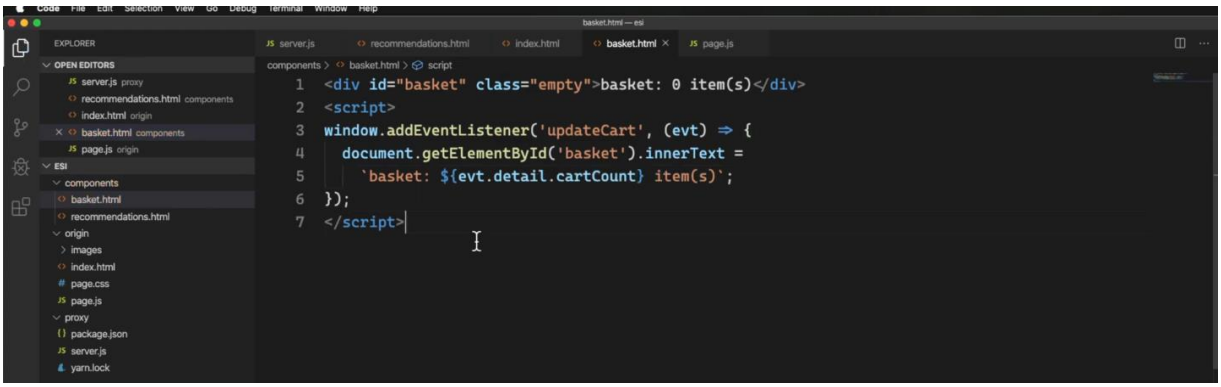


We then extract the basket HTML code into a separate basket.html file within the component folder as below

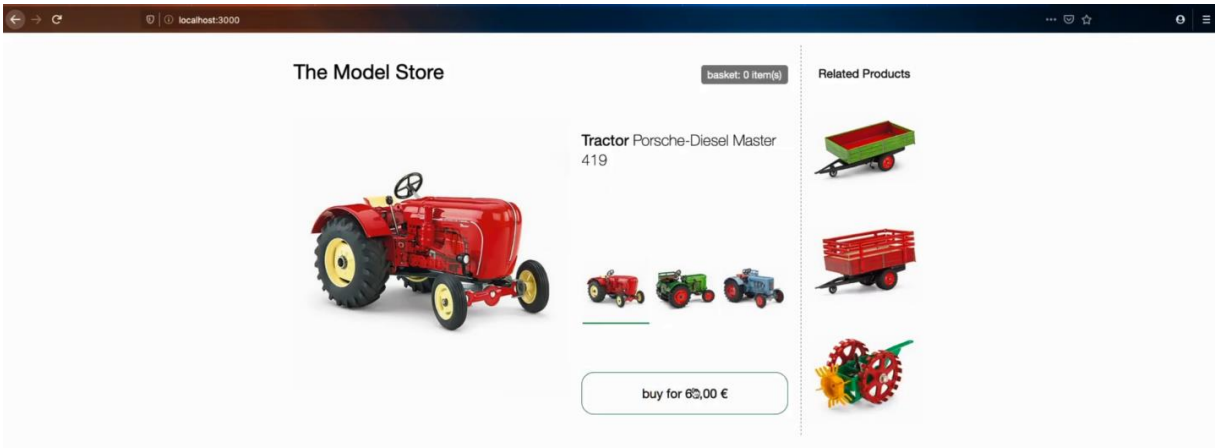


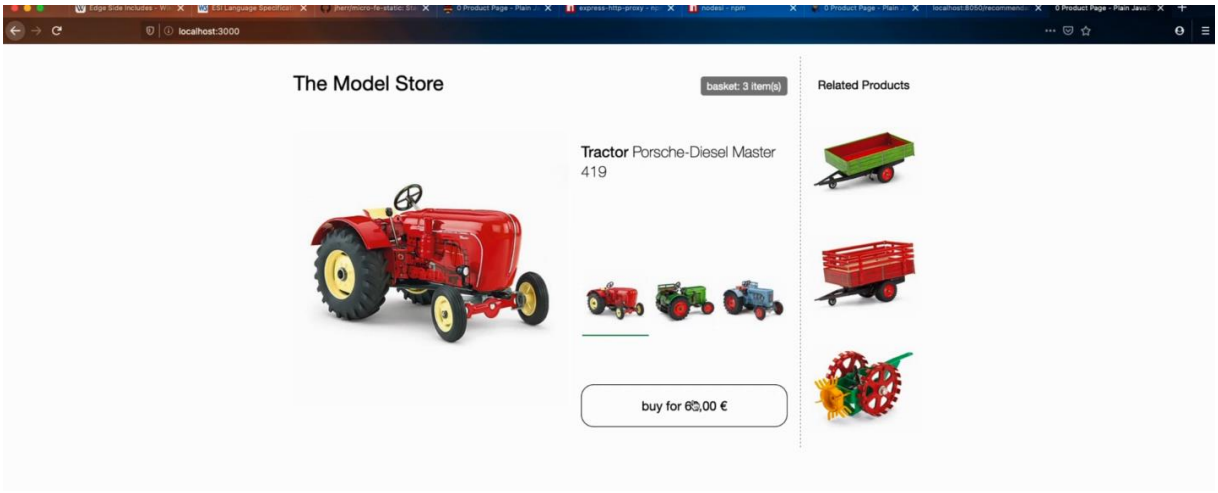






We then add code for the basket to look for that event and then update the cart count to the new value





We now can use edge side includes for injected HTML code that can interact by firing and listening to custom events on the window, we can also use Redux or RxJS instead of the window object.