Building a website to host codelabs

Introduction

I use Codelabs to document processes and step-by-step instructions for given processes or topics.

However, unlike tools for authoring codelabs, there is no real way to author a codelabs website that is not heavily geared towards Google Codelabs and its website which uses Google infrastructure and resources that may not be needed for codelabs hosted elsewhere.

Rather than try to work around Google's Codelab website tools, I decided to build my own website to host Codelabs.

The Idea

The idea for the website is to:

- Create a JSON file with information about the codelabs
- Build a website
- Insert the data from the JSON file into the template

I first thought about using a templating engine like Nunjucks or Handlebars to build the website, but I couldn't get it to work so I will use a simple HTML file instead and use a minimal handlebar template. This change forced a few changes to the plan

- Build the page to display the Codelabs
- Create an array of Codelab information on the script used to fill out the template
- Insert the data from the JSON array into the template

The template

The template is a combination of HTML tags and Handlebar templates. The root of

the template is the template element rather than a script with a specific type attribute. The `template` element was specifically designed to create inert templates that are not read as part of the document, even if they are loaded with it.

Since we expect more than one codelab to be hosted, we use Handlebar's #each helper to iterate over the list of codelabs.

For each codelab we instantiate it and fil it out with the necessary information from the data on the script.

The information inside double curly braces (llike `{{url}}` or `{{title}}` is extracted from the script data section

```
<template id="codelab-list-template">
  {{#each codelabs }}
  <div class="codelab-item">
    <h3><a href="codelabs/{{url}}">{{title}}</a></h3>
    <p>{{summary}}</p>
    <p class="small">Last Updated: {{updated}}</p>
  </div>
  {{/each}}
</template>
```

The script

The script has two sections: The first section is the data section and the second section is the template instantiation.

The data section is a constant holding an array of objects corresponding to the codelabs.

```
const codelabs = [
  {
    'updated': '2022-03-15T04:17:23-07:00',
    'title': 'How to build an API using Express.js and MongoDB',
    'summary': 'How to build an API using Express.js and MongoDB',
```

```
    'source': 'building-database-backed-app-node.md',
    'url': 'codelab-api-backend',
  }
];
```

The template instantiation does the following:

- Grabs a reference to the content of the template element
- Grabs a reference to the Handlebars compiler with the template as a paramenter
- Inserts the rendered template into the DOM using the data from the `codelabs` array

```
const template = document.getElementById('codelab-list-template').innerHTM
const renderCats = Handlebars.compile(template);
document.getElementById('codelab-list').innerHTML = renderCats({
  codelabs: codelabs,
});
```

Since this is the only template that we need to render, that's it. The rest of the HTML code loads fonts, provides a proper structure for the content and loadss the Handlebars library and the script we just discussed.

Remaining issues

There are a couple issues remaining. I'll document them here as I try to get them working.

I would like to be able to fetch the JSON file rather than hardcoding it on the script. I can't figure out if this is possible or how to do it if it is.

The last updated date is not displayed in a human friendly format. There are Handlebars helpers to do it, I just don't know if I can use them in the browser.