Using CodeMirror to show formatted code in Wagtail

Isaac Bythewood · 2022-06-11 · 3 min read

coding webdev

Using CodeMirror to show formatted code in Wagtail

Going through all the steps to use CodeMirror with Wagtail to show formatted code on the frontend of your site.

There are many ways to get code syntax highlighting on a website not the least of which is CodeMirror. CodeMirror is a fairly full featured browser text editor so it may not make sense to use it for every project but I find it a very nice user experience. A few of the features I enjoy include:

- The ability to use hotkeys, like `cmd + a` or `ctrl + a`, to highlight everything in the CodeMirror text area without grabbing the entire page
- Being able to set read only mode with a single option
- Line numbers with a single option
- And lots of language support for syntax highlighting

Note that at the time of writing this CodeMirror 6 has released but I'm only focusing on CodeMirror 5. I found CodeMirror 6 to be buggy and lacking in themes and language options as right now. I'm sure that will change in the future.

To get started install CodeMirror. I'm using `yarn` with `webpack` but this can easily be converted to almost any bundler you want to use:

```
yarn add codemirror@5.65.5
```

After that's done make a script called `code.js` and make sure it gets included in your bundle.

```
import CodeMirror from "codemirror/lib/codemirror.js";

import "codemirror/mode/python/python.js";
import "codemirror/mode/javascript/javascript.js";
import "codemirror/mode/htmlmixed/htmlmixed.js";
import "codemirror/mode/css/css.js";
import "codemirror/mode/shell/shell.js";

import "codemirror/lib/codemirror.css";
import "codemirror/theme/material.css";


document.addEventListener("DOMContentLoaded", () => {
  const blockCode = document.querySelectorAll(".block-code");

  if (blockCode) {
    Array.prototype.forEach.call(blockCode, (block) => {
      const textarea = block.querySelector("textarea");
      CodeMirror.fromTextArea(textarea, {
        theme: "material",
        lineNumbers: true,
        lineWrapping: true,
        readOnly: true,
        viewportMargin: Infinity,
        mode: textarea.dataset.language,
      });
    });
  }
});
```

A couple of notes:

- You can import as many or as few modes as you want, I'm only importing what my blog generally uses
- You can also use any theme you want in the `theme` folder, I'm currently using `material` but that may change

In conjunction with this I've added some custom CSS mostly to allow line wrapping and automatic height adjustments. I've aptly called this file `code.css`.

```
.CodeMirror {
  margin: 2rem auto;
  height: auto;
  max-width: 1000px;
}

.CodeMirror-line {
  margin: 0;
}

.CodeMirror-wrap pre {
  word-break: break-word;
}
```

After these two files are bundled and loaded into your website they will spawn a CodeMirror instance anywhere there is the following HTML block.

```
{% if block.block_type == 'code' %}
  <div class="block-code">
    <textarea data-language="{{ block.value.language }}">{{ block.value.text }}</
textarea>
  </div>
{% endif %}
```

Note the Django template tags here, you can use the above code anywhere up to this point and just remove the Django template tags and insert whatever tags you want. Now for the Wagtail portion. If you can't tell from the above Wagtail block code I'm using a `StreamField` for this. You can add a new `code` block to any `StreamField`.

```
body = StreamField([
    ('code', StructBlock([
        ('language', ChoiceBlock(choices=[
            ('python', 'Python'),
            ('javascript', 'Javascript'),
            ('htmlmixed', 'HTML'),
            ('css', 'CSS'),
            ('shell', 'Shell'),
        ])),
        ('text', TextBlock()),
    ])),
])
```

And you're done! We have a language choice block which is easily expanded upon for more languages in the future. Also make sure to include the new language mode in our `code.js` file. A feature improvement I've considered is having CodeMirror load into the `TextBlock` on my Wagtail admin panel for an even better admin experience, but for now I'll leave it at this.