

gatsby-remark-code-repls

This plug-in adds support for generating links to popular REPLs, using code in local files to populate the contents of the REPL. This enables example code to be stored along side of, and revisioned with, your website content.

It currently supports:

- [Babel](#)
- [Codepen](#)
- [CodeSandbox](#)
- [Ramda](#)

This plug-in was created to solve a couple of problems the React team has faced with [reactjs.org](#):

- Examples were stored separately from documentation (e.g. in Codepen) which made it more difficult to coordinate updates. (It was easy to forget to update an example when an API changes.)
- Examples (e.g. Codepens) were owned by a single author, so the community couldn't contribute PRs to update them without forking and fragmenting ownership.
- It was easy to create invalid links (e.g. Babel REPL links that _don't quite work).

Overview

For example, given the following project directory structure:

```
./examples/  
├── components-and-props  
│   ├── composing-components.js  
│   ├── extracting-components.js  
│   └── rendering-a-component.js  
├── hello-world.js  
└── introducing-jsx.js
```

These example files can be referenced via links in markdown that get transformed to HTML links that open the embedded code examples in a REPL. For example:

```
<!-- before -->  
[See it in Babel] (babel://hello-world.js)  
  
<!-- after -->  
<a href="https://babeljs.io/repl/#?presets=react&code_lz=...">  
  See it in Babel  
</a>  
  
<!-- before -->  
[Try it on CodePen] (codepen://components-and-props/rendering-a-component.js)  
  
<!-- after -->  
<a href="/redirect-to-codepen/components-and-props/rendering-a-component">  
  Try it on CodePen  
</a>  
  
<!-- before -->
```

```
[Try it on
CodeSandbox] (codesandbox://components-and-props/rendering-a-component.js)

<!-- after -->
<a href="https://codesandbox.io/api/v1/sandboxes/define?parameters=...">
  Try it on CodeSandbox
</a>
```

Creating CodeSandbox Example With Multiple Files

Sometimes a larger code example would require more than a single file, with various types. For example, you might have an example folder like this:

```
├─ my-example
|   ├─ index.js
|   ├─ util.js
|   └─ index.css
```

CodeSandbox supports code example with multiple files. With this plugin, you can do:

```
[Try it on
CodeSandbox] (codesandbox://my-example/index.js,my-example/util.js,my-
example/index.css)
```

Caveat

The first file path you passed to `codesandbox://` will be the entry of your example, that is, the `main` field specified in your `package.json`.

And in `index.js`, you could import other files using the ES6 modules syntax:

```
import { foo } from "./utils"

import "./index.css"
```

How does it work?

Codepen links point to Gatsby pages (also created by this plug-in) that redirect using the [Codepen prefill API](#) to create a working, runnable demo with the linked example code.

Babel and CodeSandbox links use the [same URL compression schema used by the Babel REPL](#) to embed the local code example in a URL that enables it to be viewed directly within the target REPL.

Ramda links use basic URL encoding to embed the local code example in a URL that enables it to be viewed directly within Ramda's REPL.

All example links are also verified to ensure that they reference valid example files. For example, if there is a link to `codepen://components-and-props/rendering-a-component`, this plug-in will verify that a file `components-and-props/rendering-a-component.js` exists within the specified examples directory. (This will avoid broken links at runtime.)

Installation

```
npm install gatsby-remark-code-repls
```

Usage

```
// In your gatsby-config.js
{
  resolve: 'gatsby-remark-code-repls',
  options: {
    // Optional default link text.
    // Defaults to "REPL".
    // e.g. <a href="...">Click here</a>
    defaultText: 'Click here',

    // Example code links are relative to this dir.
    // e.g. examples/path/to/file.js
    directory: `${__dirname}/examples/`,

    // Optional link target.
    // Note that if a target is specified, "noreferrer" will also be added.
    // e.g. <a href="..." target="_blank" rel="noreferrer">...</a>
    target: '_blank',

    // Provider specific options
    codepen: {
      // Optional path to a custom redirect template.
      // The redirect page is only shown briefly,
      // But you can use this setting to override its CSS styling.
      redirectTemplate: `${__dirname}/src/redirect-template.js`,

      // Optional HTML contents to inject into REPL.
      // Defaults to `<div id="root"></div>`.
      // e.g. '<div id="root"></div>'
      html: '',

      // Optional externals to load from a CDN.
      // e.g. '//unpkg.com/react/umd/react.development.js'
      externals: [],

      // Include CSS with matching name.
      // If set to `true`, when specifying `file1.js` as example file,
      // it will try to inject the CSS in `file1.css` if the file exists,
      // otherwise the default behaviour is preserved
      includeMatchingCSS: false,
    },
  },

  codesandbox: {
```

```
// Optional HTML contents to inject into REPL.
// Defaults to `
```