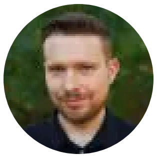
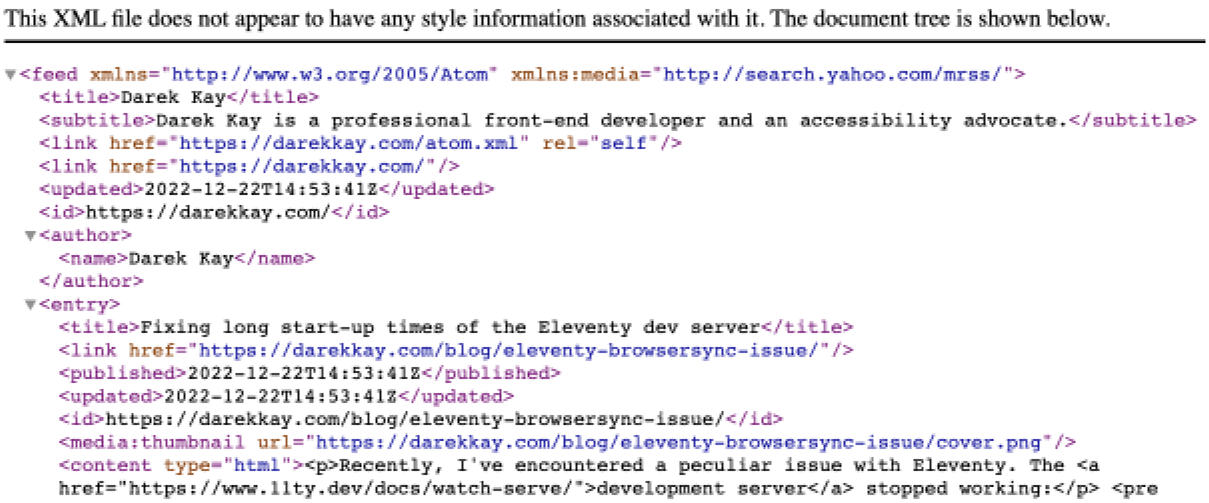


Style your RSS feed

Published on 20 Jun 2023 in rss blog

RSS is not dead. It is not *mainstream*, but it's still a thriving protocol, especially among tech users. However, many people do not know what RSS feeds are or how to use them. Most browsers render RSS as raw XML files, which doesn't help users understand what it's all about:



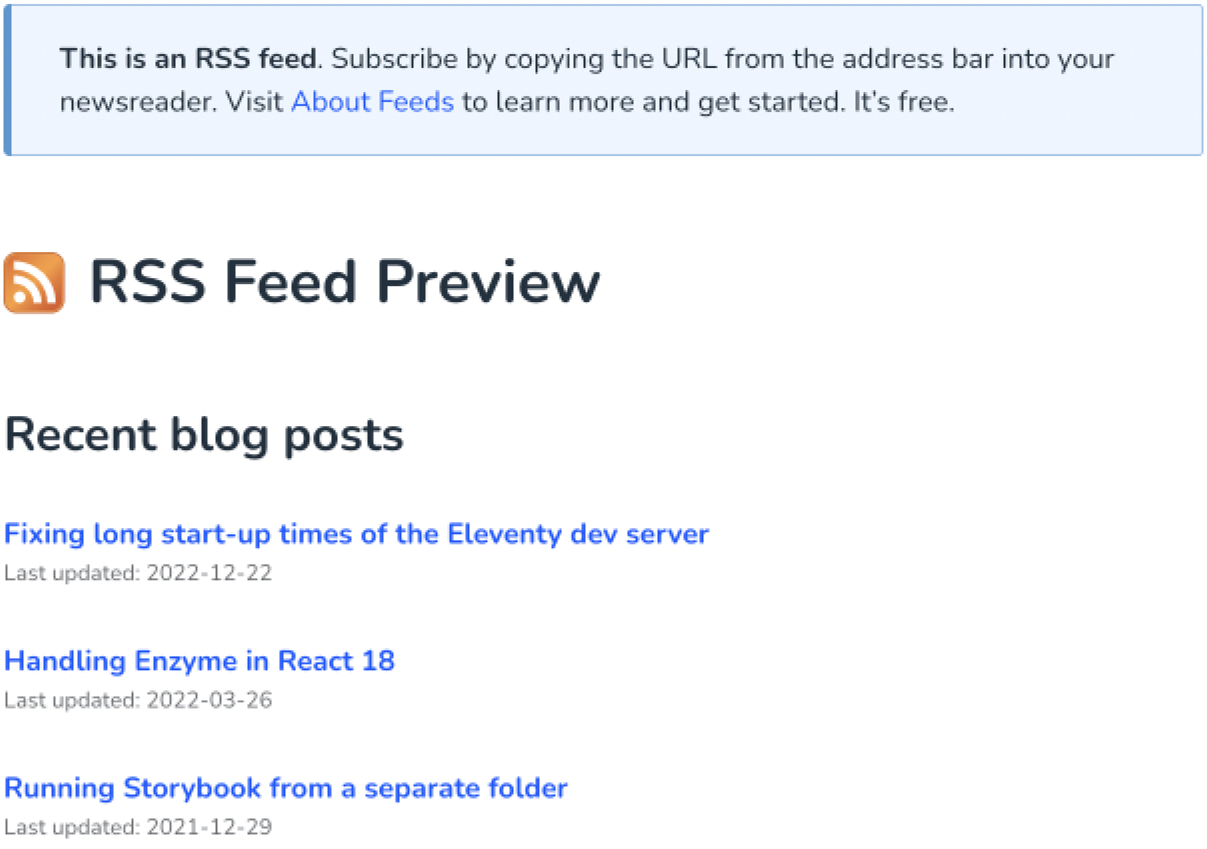
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<feed xmlns="http://www.w3.org/2005/Atom" xmlns:media="http://search.yahoo.com/mrss/">
   <title>Darek Kay</title>
   <subtitle>Darek Kay is a professional front-end developer and an accessibility advocate.</subtitle>
   <link href="https://darekkay.com/atom.xml" rel="self"/>
   <link href="https://darekkay.com/"/>
   <updated>2022-12-22T14:53:41Z</updated>
   <id>https://darekkay.com/</id>
 ▼<author>
     <name>Darek Kay</name>
   </author>
 ▼<entry>
     <title>Fixing long start-up times of the Eleventy dev server</title>
     <link href="https://darekkay.com/blog/eleventy-browsersync-issue/"/>
     <published>2022-12-22T14:53:41Z</published>
     <updated>2022-12-22T14:53:41Z</updated>
     <id>https://darekkay.com/blog/eleventy-browsersync-issue/</id>
     <media:thumbnail url="https://darekkay.com/blog/eleventy-browsersync-issue/cover.png"/>
     <content type="html"><p>Recently, I've encountered a peculiar issue with Eleventy. The <a
     href="https://www.11ty.dev/docs/watch-serve/">development server</a> stopped working:</p> <pre
```

In this post, I'll explain how to style RSS feeds and educate readers at the same time.

XSL(T) to the rescue

This is how the RSS feed for this blog looks like:



This is an RSS feed. Subscribe by copying the URL from the address bar into your newsreader. Visit About Feeds to learn more and get started. It's free.

RSS Feed Preview

Recent blog posts

Fixing long start-up times of the Eleventy dev server
Last updated: 2022-12-22

Handling Enzyme in React 18
Last updated: 2022-03-26

Running Storybook from a separate folder
Last updated: 2021-12-29

To style a raw XML file in a browser, you have to provide styling information. You can do that by attaching an `xml-stylesheet` processing instruction within the RSS feed:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/rss.xsl" type="text/xsl"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:media="http://search.yahoo.com/mrss/">
  ...
</feed>
```

The `href` attribute specifies a URL to a valid XSL file. You can check out mine for inspiration. Here's the gist:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="3.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:atom="http://www.w3.org/2005/Atom">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
  <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
    <head>
      <title>
        RSS Feed | <xsl:value-of select="/atom:feed/atom:title"/>
      </title>
      <link rel="stylesheet" href="/assets/styles.css"/>
    </head>
    <body>
      <p>
        This is an RSS feed. Visit
        <a href="https://aboutfeeds.com">About Feeds</a>
        to learn more and get started. It’s free.
      </p>
      <h1>Recent blog posts</h1>
      <xsl:for-each select="/atom:feed/atom:entry">
        <a>
          <xsl:attribute name="href">
            <xsl:value-of select="atom:link/@href"/>
          </xsl:attribute>
          <xsl:value-of select="atom:title"/>
        </a>
        Last updated:
        <xsl:value-of select="substring(atom:updated, 0, 11)" />
      </xsl:for-each>
    </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

This code is inspired by pretty-feed-v3, but I've adopted it for the Atom specification.

The XSL file transforms the XML feed into a valid HTML document that any browser can render. You can use any information included in the XML file and put it into a new markup structure. You can even include content that is not part of the XML feed. A perfect use case is to include a note about what RSS feeds are and how to use them (as shown in the example above).

You can define inline CSS styles with a regular `<style>` element, but it's also possible to import external CSS files. I've imported my blog's main CSS stylesheet, so I didn't have to write any new code to make the RSS feed match my blog's design.

It's also possible to use XSLT functions to alter values. Here's how I truncate the timestamp to display only the date:

```
<!-- Full timestamp -->
<xsl:value-of select="atom:updated" />

<!-- Date only -->
<xsl:value-of select="substring(atom:updated, 0, 11)" />
```

The browser support for XSL transformations is great; unsupported browsers will fall back to the default behavior (progressive enhancement).

Examples

Here are some styled RSS feeds you can check for inspiration:

- This blog
- My photography website
- BBC News
- Hsiaoming Yang
- Dave Rupert
- Oli Warner

The XSL files are public, so you can check how those pages have been implemented.

Sitemap

You can apply XSL files to *any* XML file. Another use case for styled XML files is a website sitemap. Although sitemaps are meant to be consumed by machines (e.g. crawlers), you can style them with little effort. See my sitemap and the respective XSL file as an example. Here's the gist:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:sitemap="http://www.sitemaps.org/schemas/sitemap/0.9">
  <xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
      <body>
        <h1>Sitemap</h1>
        <xsl:for-each select="/sitemap:urlset/sitemap:url">
          <a>
            <xsl:attribute name="href">
              <xsl:value-of select="sitemap:loc"/>
            </xsl:attribute>
            <xsl:value-of select="sitemap:loc"/>
          </a>
          Last updated:
          <xsl:value-of select="substring(sitemap:lastmod, 0, 11)" />
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Conclusion

I love RSS, and — according to my stats — many of my readers love it, too. With all the social media walled gardens, I hope for RSS to become more popular. By providing a custom XSL file, you can make your RSS feed look nice *and* include some information about what RSS actually is.

Related posts

- Estimating the number of your RSS subscribers

I am **Darek Kay** — a software developer, blogger and hobby photographer.

You can reach me via **Mastodon**, **GitHub** or **Email**.

Subscribe to my **RSS Feed**.