1. Markup language for content distribution: RSS

1.1 Introduction

Learning objectives, concepts, and skills:

- Describe the syndication systems
- Analyze the most used formats for web syndication
- Use tools to create and consume content

1.2 Content syndication

Content Syndication means republishing the same piece of content (an article, a video, an infographic, etc.) on one or more different websites. Syndicate content helps publishers give fresh information to their readers. The third-party website gets free, relevant content. The content creator gets free exposure and publicity, and backlinks to their own website, boosting their organic traffic.

Syndication in the **media industry** is not new. Before the Internet, newspapers and magazines with large circulation and readership sometimes printed syndicated content provided by smaller publications and freelance writers. Both benefitted. The small publication got to reach a much wider audience. The large publication got more content without having to invest resources in creating it.

1.3 RSS

Really Simple Syndication (RSS) is the term used to refer to the **collection** of **Web feed formats** that provide updated or shared information in a standardised way. The information could be a website, blog entries, news headlines, audio, or video files. RSS documents can contain complete or summarised text, metadata, author, and publishing information.

RSS feeds benefit both **publishers** and **subscribers.** Using feeds, publishers syndicate work automatically in a format that different applications can easily access and view.

A Really Simple Syndication reader (**RSS reader**) is the tool used to read Really Simple Syndication feeds.

Readers collect the website URLs that subscribers would like to follow. Subscribers store feeds manually or by clicking the RSS feed button found on most browsers or websites. The reader can check frequently for updates and download them for the subscriber.

There are some distinct **advantages** to using **RSS**. Instead of visiting the individual websites, RSS feeds can help provide users with updates and information from different sites in one convenient place. Another advantage is in ensuring the user's **privacy** because, unlike a website's email subscription, RSS does not require the user to provide contact information.

RSS is not as popular as it was, but many people still use RSS to stay informed about news, podcasts, and other information online. Recent browsers can directly read RSS files, but you can use a special **RSS reader** or **aggregator** too.

Now, many people follow a social media channel online rather than use and subscribe to a feed.

RSS is available in different versions which are not compatible with each other: RSS 0.9, RSS 1.0 and RSS 2.0. **Atom** is also an XML-based feed language designed to fit the needs of webloggers and news sites. Atom attempts to replace RSS feeds and remove the uncertainty with incompatibilities in the different RSS versions.

1.3.1 RSS feeds

An **RSS feed** is simply an **XML text file**. It's created by a website publisher and contains a running list of articles or other content published by the site, with the newest entry always at the top of the list. Each entry contains details like the article's title, description, and link to the content.

RSS feeds are published and updated in real time. Subscripts to a site's RSS feed will always have access to the newest published content. That can be handy for news sites and podcasts that are frequently updated.

An RSS feed displays content that matters to the user.

Subscriptions trigger an automatic transmission of updates to RSS feed readers. These updates summarise new information. RSS readers receive the content from feeds in real-time, placing the most recent items at the top.

In the **podcast** world, RSS feeds are especially important, helping to ensure that new podcast episodes are easy to find for relevant audiences.

RSS feeds are how new information and published podcast episodes can be transferred to popular podcast directories, like Apple and Spotify. Therefore, most podcast hosting platforms integrate RSS technology.

Most **web content managers** auto-generate the XML file for RSS feeds. For example, in WordPress it is enough to add '/feed' after a web URL to access it.

You can create an RSS feed with a file written in **XML**.

An RSS file usually ends in either **.rss** or .**xml**. A common name for the feed file is **feed.rss**.

RSS files have the **rss root element**:

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
    ...
</rss>
```

The distribution **channel** must be located inside the **rss** global container. The channel is how people refer to the RSS service. If an HTML website contains the same information as an RSS file, the title of the channel should be the same as the title of the website.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
    <channel>
    ...
    </channel>
</rss>
```

An RSS file can have multiple channels.

The channel element defines the **channel**, along with some other children elements:

- **title** - Required. The title of the channel.
- **link** - Required. A URL to the website who provides the web feed.
- **description** - Required. A description of the channel.
- **language** - Optional. The language code of the language used by the channel. Example: "en-US".

There are other optional elements.

Each element to update (article, podcast, etc.) is called an **item**.

The **item** element defines an item, which should be children of a channel element. Children elements are:

- **title** - Required. The title of the item.
- **link** - Required. A URL link to a website. If the item contains a summary of an article on a website, the link will be a link to the full article.
- **description** - Required. A description of the item.
- **author** - Optional. E-mail address of the author of the item.

- **pubDate** – Optional. Publishing date.

There are other optional elements.

The following code is an example of a simple RSS feed:

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
    <channel>
        <title>Example Feed</title>
        <description>Insert witty or insightful remark here</description>
        <link>http://example.org/</link>
        <pubDate>Sat, 12 Nov 2022 18:30:02 GMT</pubDate>
        <managingEditor>johndoe@example.com (John Doe)</managingEditor>
        <item>
            <title>Test page</title>
            <link>http://example.org/test.html</link>
            <pubDate>Sat, 12 Nov 2022 18:30:02 GMT</pubDate>
            <description>Some text.</description>
        </item>
    </channel>
</rss>
```

Validate using a feed validator https://validator.w3.org/feed/ feeds once created. Upload the feed if is valid. The feed directory inside a project is used to locate the feed.

It is important to link the feed file to a web page in its header.

```
<link rel="alternate" type="application/rss+xml" title="RSS"
href="feed/feed.rss">
```

To inform their website visitors that an RSS feed is available containing content related to the website, most webmasters include a colourful graphic on the website. It has become a standard that nearly all websites having RSS feeds available will use colourful graphics, such as flags informing that RSS feeds are available for specific content. The flags were initially standardised as bright orange rectangles, but as the popularity has grown, webmasters have bent the rules a bit and have become more creative with their flags and graphics. Use a colourful graphic that works within your website design to signify that you have an RSS feed available.

Example of graphic:

It is important to link the RSS feed button to the XML feed file. Below is an example using a graphic file named feedicon.png:

```
<a href="https://example.gal/feed/feed.rss"><img style="border: 0px solid;
width: 28px; height: 28px;" alt="feed icon" src="images/feedicon.png"></a>
```

1.4 Atom feeds

Atom is also an XML format, so it uses **.xml** or **.atom** extension.

RSS does not support extension to namespaces, while Atom supports it.

A Feed includes some metadata, followed by several entries.

The following code shows an example of an Atom feed:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
    <title>Articles</title>
    <subtitle>...</subtitle>
    <updated>2022-06-21T06:08:53Z</updated>
        ...
    <entry>
        <author>
            <name>Author's name</name>
        </author>
        <title>...</title>
        ...
        <updated>2022-06-21T06:08:53Z</updated>
        <published>2022-06-21T06:07:57Z</published>
        <category../>
        <summary>...</summary>
    </entry>
    <entry>
        ...
    </entry>
    ...
</feed>
```

The name used for the items of information that make up an Atom feed is Atom entries.

The Atom "feed" element contains the following information:

- **Atom namespace** - REQUIRED. We must declare the Atom namespace as http://www.w3.org/2005/Atom.
  For example: <feed xmlns="http://www.w3.org/2005/Atom">.
- **id** - REQUIRED. Specifies an identification that uniquely identifies this Atom feed.
  For example: <id>http://dev.example.com/</id>.
- **title** - REQUIRED. Specifies the name of this Atom feed. For example: <title> Software Developers</title>.
- **updated** - REQUIRED. Specifies the publication date and time of this Atom feed in the ISO.8601 standard format.
  For example: <updated>2022-11-13T18:30:02Z</updated>.
- **author** - REQUIRED, unless all "feed/entry" sub elements have an "author" sub element. Specifies the profile of the author of this Atom feed. We can specify multiple "author" sub elements.
  For example: <author><name>Author's name</name></author>

- **contributor** - OPTIONAL. Specifies the profile of a contributor of this Atom feed. We can specify multiple "contributor" sub elements.
  For example: <contributor><name>John Doe</name></author>
- **generator** - OPTIONAL. Specifies a string indicating the program used to generate this Atom feed.
  For example: <generator>My Site Builder</generator>.
- **category** - OPTIONAL. Specifies a category that this Atom feed belongs to. We can specify multiple "category" sub elements.
  For example: <category term="Programming"/>.
- **icon** - OPTIONAL. Specifies a URI referring to an icon for this Atom feed.
  For example: <icon>http://dev.example.com/favicon.ico</icon>.
- **logo** - OPTIONAL. Specifies a URI referring to a logo for this Atom feed.
  For example: <logo>http://dev.example.com/logo.png</logo>.
- **link** - OPTIONAL. Specifies URLs to retrieve this Atom feed and other related resources. At least one link sub element with rel="self" is recommended. We can specify multiple "link" sub elements.
  For example: <link rel="self">http://dev.example.com/atom_xml.php</link>.
- **rights** - OPTIONAL. Specifies copyright notice for this Atom feed.
  For example: <rights>Copyright (c) 2022 example.com</rights>.
- **subtitle** - OPTIONAL. Specifies a short description of this Atom feed.
  For example: <subtitle>A large collection of FAQs, tutorials, tips and code examples for application and Web developers</subtitle>.
- **entry** - OPTIONAL. Provides XML sub element representing an Atom feed entry.

1.5 JSON feeds

We can use JSON feeds as an alternative for using XML for asynchronously transmitting structured information between client and server. It is a lightweight text-based open standard designed for human-readable data interchange.

The first thing we must do to use JSON feeds is to create a feed.json file. We can use the name we want but feed.json is a very common name.

Following code is an example of a very simple JSON feed:

```
{
    "version": "https://jsonfeed.org/version/1",
    "title": "My Example Feed",
    "home_page_url": "https://example.org/",
    "feed_url": "https://example.org/feed/feed.json",
    "items": [
        {
            "id": "2",
            "content_text": "This is a second item.",
            "url": "https://example.org/second-item"
        },
        {
            "id": "1",
            "content_html": "<p>Hello, world!</p>",
            "url": "https://example.org/initial-post"
        }
    ]
}
```

We must link the JSON feed file to our web page. To do that, we must write the address of our RSS file on the header of our web page.

```
<link rel="alternate" href="https://example.org/feed/feed.json"
type="application/json" />
```

1.6 RSS readers

We can use an RSS feed reader to access important data from a variety of selected feeds or sources. A reader can display posts from a list of feeds, usually based on sites user subscribes to or shows an interest in.

There are many kinds of readers and new ones crop up all the time. Some are installed on desktop, others are web-based, some relate to e-mail software, and some are connected to browsers. The operating system of your computer may influence the type of reader you can use. Browser-based newsreaders let you read RSS feeds from any computer, whereas other readers will only have feeds available on the computer where it is installed on.