XML

vs. JSON, YAML, TOML, etc.

Yegor Bugayenko

Lecture #7 out of 16
80 minutes

The slidedeck was presented by the author in this YouTube Video

Extensible Markup Language (XML)

XSD, XPath, XSLT, XQuery, etc.

JavaScript Object Notation (JSON)

YAML, TOML, CSV

Books, Venues, Call-to-Action

Chapter #1:

Extensible Markup Language (XML)

Library in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <book id="42">
    <author>David West</author>
    <title>Object Thinking</title>
  </book>
  <book id='43'>
    <author>Martin Fowler</author>
    <title>Refactoring</title>
  </book>
</library>
```

Namespaces

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <library xmlns="https://innopolis.university/ssd16"
3    xmlns:a="https://www.amazon.com"
4    xmlns:t="https://www.twitter.com">
5    <book id="42">
6      <a:dp>0134757599</a:dp>
7      <t:author>@martinfowler</t:author>
8      <author>Martin Fowler</author>
9      <title>Refactoring</title>
10   </book>
11 </library>
```

Escaping

```
<?xml version="1.0" encoding="UTF-8"?>
<formulas>
  <f title='Fibonacci&amp;apos;s'> <!-- Fibonacci's -->
    <e>if x &lt; 2 return x</e> <!-- if x < 2 return x -->
    <e>else return f(x-1) + f(x-2)</e>
  </f>
</formulas>
```

XML Based Formats/Protocols

SOAP, RSS, Atom, SVG, XHTML, HTML5,
Open Office XML, XMPP,
SyncML, RDF, XMI, XMIR :)

https://en.wikipedia.org/wiki/Category:XML-based_standards →

Chapter #2:

XSD, XPath, XSLT, XQuery, etc.

XML Schema Definition (XSD)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="book">
    <xs:sequence>
      <xs:element name="author" minOccurs="1" maxOccurs="1"/>
      <xs:element name="title" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:decimal"/>
  </xs:complexType>
  <xs:element name="library">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="book" type="book" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML Path Language (XPath)

```
<library><book id=42><author>David West</..></..></..>
```

```
/library/book[@id='42']
```

```
//book[@id='42']
```

```
//book[first()]
```

```
//book[author='David West']
```

```
//book[author[text()='David West']]
```

XSL Transformations (XSLT)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="book">
    <item>
      <xsl:value-of select="title"/>
      <xsl:text> by </xsl:text>
      <xsl:value-of select="title"/>
    </item>
  </xsl:template>
  <xsl:template match="node()|@*">
    <xsl:copy>
      <xsl:apply-templates select="node()|@*"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

Chapter #3:

JavaScript Object Notation (JSON)

[JSON]

JSON for the Library

```
[
  {
    "author": "David West",
    "id": 42,
    "title": "Object Thinking"
  },
  {
    "author": "Martin Fowler",
    "id": 43,
    "title": "Refactoring"
  }
]
```

https://www.yegor256.com/2015/11/16/json-vs-xml.html →

JSON to JavaScript Object and Backwards

```
var a = JSON.parse('{"age":  25}').age;
```

```
JSON.stringify({age:  25});
```

Chapter #4:

YAML, TOML, CSV

Yet Another Markup Language (YAML)

```yaml
library:
- id: 42
  author: David West
  title: Object Thinking
- id: 43
  author: Martin Fowler
  title: Refactoring
```

TOML

```
[library.a]
  id = 42
  author = "David West"
  title = "Object Thinking"
[library.b]
  id = 43
  author = "Martin Fowler"
  title = "Refactoring"
```

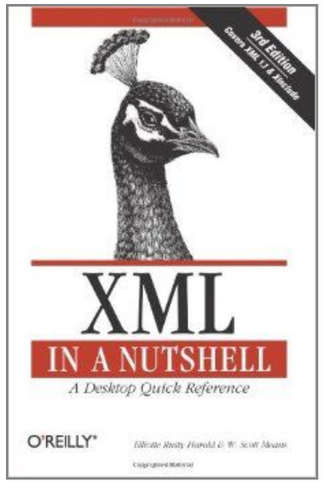
Comma-Separated Values (CSV)

```
1 Id,Author,Title
2 42,David West,Object Thinking
3 43,"Martin Fowler","Refactoring"
```

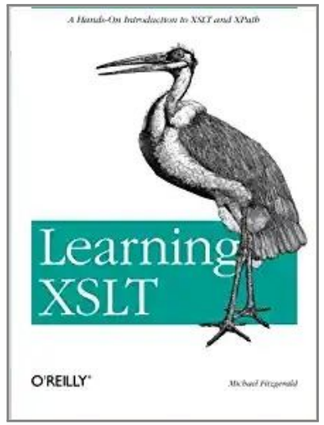
https://en.wikipedia.org/wiki/List_of_file_formats →

Chapter #5:

Books, Venues, Call-to-Action



Elliotte Rusty Harold and W. Scott Means. *XML in a Nutshell: A Desktop Quick Reference*. O'Reilly Media, 2000. doi:10.5555/557805



Michael Fitzgerald. *Learning XSLT: A Hands-on Introduction to XSLT and XPath*. O'Reilly Media, 2003

Call to Action:

In your application, make sure your data is represented in XML, at least in one place, and being transformed by XSLT.

Design your own data format.

Still unresolved issues:

- How to map XML/JSON to objects?
- How to print object to XML/JSON?
- How to create a common binary format?
- How to restore the popularity of XSLT?

Bibliography

Michael Fitzgerald. *Learning XSLT: A Hands-on Introduction to XSLT and XPath*. O'Reilly Media, 2003.

Elliotte Rusty Harold and W. Scott Means. *XML in a Nutshell: A Desktop Quick Reference*. O'Reilly Media, 2000. doi:10.5555/557805.