

RUHR-UNIVERSITÄT BOCHUM

Bridging the Gap: Secure and lossless conversion of XML data structures to the JSON format

Bachelor thesis ■ March 30, 2017 – June 29, 2017

June 06, 2017

Advisors: Dennis Felsch & Paul Rösler

Jan Holthuis

- 1 Quick Recap
- 2 Progress report
- 3 The curious case of the > character
- 4 Next steps
- 5 Commercial break

Last time we learned that...

- Based on my current conversion criteria, *no existing solution* allows lossless XML→JSON→XML roundtrips...
- ...but I found a promising candidate named **JsonML**
 - Handles most stuff correctly
 - Resulting JSON is not very friendly because it uses Arrays almost for everything
 - Multiple implementations exist

- 1 Quick Recap
- 2 Progress report
- 3 The curious case of the > character
- 4 Next steps
- 5 Commercial break

Progress made in the last two weeks

Complex tests

- Most of my existing tests concentrate on small, isolated conversion problems
- Now I added some “real-world” examples
 - The official sample documents for RSS version 0.91, 0.92 and 2.0
 - The XML 1.0 specification as XML and XHTML
 - Two SVG images from the W3 SVG Web toolkit (one of them has embedded JavaScript)
 - The resulting XML of an API call to the MusicBrainz.org Web Service v2
 - The ODF v1.0 (Second Edition) specification in Flat OpenDocument Text format (*.fodt)

Progress made in the last two weeks

Character support tests

■ Added a bunch of character support tests

```
\$ ls -l code/test-documents/charsupport/
total 1048
-rw-r--r-- 1 jan jan    258 May 25 16:03 chars-u000009-u00000D.xml.testcase
-rw-r--r-- 1 jan jan   1374 May 25 00:19 chars-u000020-u00007E.xml.testcase
-rw-r--r-- 1 jan jan    304 May 25 00:19 chars-u00007F-u000084.xml.testcase
-rw-r--r-- 1 jan jan    609 May 25 00:19 chars-u000080-u00009F.xml.testcase
-rw-r--r-- 1 jan jan    565 May 25 00:19 chars-u000086-u00009F.xml.testcase
-rw-r--r-- 1 jan jan 982708 May 25 00:19 chars-u0000A0-u00D7FF.xml.testcase
-rw-r--r-- 1 jan jan    802 May 25 00:19 chars-u00FDD0-u00FDEF.xml.testcase
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u01FFFE-u01FFFF.xml.testcase
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u02FFFE-u02FFFF.xml.testcase
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u03FFFE-u03FFFF.xml.testcase
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u04FFFE-u04FFFF.xml.testcase
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u05FFFE-u05FFFF.xml.testcase
[...]
-rw-r--r-- 1 jan jan    267 May 25 00:19 chars-u0FFFFE-u0FFFFF.xml.testcase
-rw-r--r-- 1 jan jan    271 May 25 00:19 chars-u10FFFE-u10FFFF.xml.testcase
```

- 1 Quick Recap
- 2 Progress report
- 3 The curious case of the > character
- 4 Next steps
- 5 Commercial break

The curious case of the > character

Escaping

- Want a < char in a text node (not CDATA)? Escape it! \Rightarrow <
- What about >?
 - Escaping it to > is not necessary...
 - ... **except** "when it appears in the string ']]>' in content, when that string is not marking the end of a CDATA section." (XML 1.0, Section 2.4).

The curious case of the > character

Why does this matter?

- Two JavaScript-based converters failed this test:
 - JsonML
 - JXON
- Both are using DOMParser for XML, which part of the Browser-only Web API
 - JXON is using the `xmldom` Node.js implementation
 - JsonML is browser-only, so I added `xmldom` support using monkey patch for easier testing on the CLI

The curious case of the > character

JsonML-xmlDOM glue

```
/* Create fake browser context by using methods from xmldom module */
const ctx = {
  "DOMParser": xmldom.DOMParser,
  "document": (new xmldom.DOMParser()).parseFromString("<x/>", "text/xml"),
  "window": {
    "DOMParser": xmldom.DOMParser,
    "XMLSerializer": xmldom.XMLSerializer,
  },
};

/* Run code in fake browser context */
let filename = require.resolve("jsonml-tools/jsonml-xml.js")
let code = fs.readFileSync(filename, "utf-8");
vm.runInNewContext(code, ctx);
```

The curious case of the > character

Investigating the problem

- Both `xml.dom`-using converters fail. Coincidence?
- Found out that there was a regression in `xml.dom`...
- ... so `xml.dom` was the cause of the problem.
- A commit from 2012 (!) broke this behaviour
- There was a unittest, but the commit broke that too
- \Rightarrow I wrote a patch to make `xml.dom` great again

The curious case of the > character

Patching xmldom

```
diff --git a/dom.js b/dom.js
@@ -1053,7 +1053,7 @@ function serializeToString(node,buf,isHTML,nodeFilter,
    ↪ visibleNamespaces){
  case TEXT_NODE:
-   return buf.push(node.data.replace(/[\<&]/g,_xmlEncoder));
+   return buf.push(node.data.replace(/[\<&]/g,_xmlEncoder).replace(/\\>/g,']']'+
    ↪ _xmlEncoder('>')));

diff --git a/test/dom/serializer.js b/test/dom/serializer.js
@@ -5,7 +5,7 @@ wows.describe('XML Serializer').addBatch({
  'text node containing "]]>": function() {
    var doc = new DOMParser().parseFromString('<test/>', 'text/xml');
    doc.documentElement.appendChild(doc.createTextNode('hello ]]> there'));
-   console.assert(doc.documentElement.firstChild.toString() == 'hello ]]> there',doc.
    ↪ documentElement.firstChild.toString());
+   console.assert(doc.documentElement.firstChild.toString() == 'hello ]]&gt; there',doc
    ↪ .documentElement.firstChild.toString());
  },
```

- 1 Quick Recap
- 2 Progress report
- 3 The curious case of the > character
- 4 Next steps**
- 5 Commercial break

Next Steps

- Continue writing
- ???
- Profit!

- 1 Quick Recap
- 2 Progress report
- 3 The curious case of the > character
- 4 Next steps
- 5 Commercial break**

Nice formatting style for \LaTeX 's `minted`

(Shameless Self-Advertising)

- You use the RUB thesis \LaTeX template?
- You need nice code listings (using the `minted` package) that **use the RUB corporate design colors?**
- Check out the `rub` style from [github.com/Holzhaus/pygments-style-rub!](https://github.com/Holzhaus/pygments-style-rub)

```
#!/usr/bin/python3

from engine import RunForrestRun

"""Test code for syntax highlighting!"""

class Foo:
    def __init__(self, var):
        self.var = var
        self.run()

    def run(self):
        RunForrestRun() # run along!
```


Questions?

Reach out via email:

- **Jan Holthuis**
jan.holthuis@rub.de

