Introducing: The X3D JSON Loader (2.0 beta) and X3D JSON Prototype Expander (2.0 beta) – Your solution for XMLDOM -> X3D JSON -> XMLDOM development for X3DOM and X_ITE.

For a Quick Start, edit flipper.html and replace flipp.json with your JSON URL and put flipper.html, X3DJSONLD.js and loaderJQuery.js on your web server. Then open flipper.html in your web browser.

Good luck.

I still have extra stuff in the global scope of JavaScript—let people know this. Pull requests and forks are welcome, as long as you agree to the license.

License is here: https://github.com/coderextreme/X3DJSONLD/blob/master/LICENSE
Repository is here: https://github.com/coderextreme/X3DJSONLD/

For a web browser, a live, development version of the X3D JSON loader (I recommend downloading locally or forking) in your HTML, put:

```
<script type="text/javascript"
src="https://raw.githubusercontent.com/coderextreme/X3DJSONLD/master/X3DJSONLD.js">
</script>
```

somewhere in the script (see index.html),

call

```
loadX3DJS(document.implementation, json, url, xml, NS, loadSchema,
doValidate, function(element, xmlDoc) {
      Then append the element to your DOM:
      document.querySelector(selector).appendChild(element);
      x3dom.reload();
}
```

`selector` is the CSS selector which you want to append the X3DOM HTML code to.
`json` is the X3D JSON you want to display.
`url` is used for resolving URLs in the X3D JSON. Should be similar or the same as the URL you passed to retrieve the JSON from the server.
`xml` is the array or LOG for inclusion into X_ITE via createX3DFromString, this would normally work something like:

```
      var browser = X3D.getBrowser("X3D");
      browser.replaceWorld(browser.createX3DFromString(xml.join("\n"));
```

“X3D” is the CSS query selector.

NS is the namespace to use when creating elements in the DOM for the XML Serializer. `http://www.w3.org/1999/xhtml` normally works for X3DOM and `http://www.web3d.org/specifications/x3d-namespace` normally works for X_ITE. Leaving NS off is also acceptable, but may lead to results you don't like.

Sample code for X_ITE where #x_ite is the id of your X3DCanvas :

```
var content = xml.join("\n");
X3D(function() {
      var browser = X3D.getBrowser("#x_ite");
      browser.replaceWorld(browser.createX3DFromString(content);
      });
```

For the prototype expander a live, development version (I recommend downloading locally or forking) in your HTML, put:

```
<script type="text/javascript"
src="https://raw.githubusercontent.com/coderextreme/X3DJSONLD/master/PrototypeExpander.js"></script>
```

then call (does not modify extern protos yet):

```
json = protoExpander.prototypeExpander(url, json);
json = flattener(json);
```

`json` is the X3DJSON you want to expand protos for (also modifies the parameter as output)

There is a lot of useful code in loaderJQuery.js. index.html (for protos), flipper.html (for the base loader), prototypes.html and prototypes2.html are good examples.

To run XML -> JSON conversions, put your .x3d file in src/main/data, and cd to src/main/shell and run `sh several.sh ../data/file.x3d` You will find output in data, nashorn, java, and python folders (way down low for the latter).

To run the proto expander on the server, put your .json and .x3d files in src/main/data and cd to src/main/shell and run `sh runppp.sh` The XML will be in src/main/shell/data and the JSON will be in src/main/ppp. Good luck!

John Carlson