[Processing.js #2002] loadXML() is broken

10 messages

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 3:49 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 06:49 AM

Tested on 2.08b.

```
  String url="http://www.matteosistisette.com/upf-redes-x/php/_processing/formas_xml/cargaformas.php"
  XML xml=myLoadXML(url);
  println(xml);
```

prints the following:

```
  <html><body><parsererror style="display: block; white-space: pre; border: 2px solid #c77; padding: 0 1em 0 1em;
margin: 1em; background-color: #fdd; color: black"><h3>This page contains the following errors:</h3><div style="font-
family:monospace;font-size:12px">error on line 1 at column 1: Document is empty
</div><h3>Below is a rendering of the page up to the first error.</h3></parsererror></body></html>
```

This html code is _not_ generated by the server, it's generated by the browser. Apparently it is because the xml is seen as empty. But it is not. Or it would be not if it was loaded in the first place (Processing doesn't even make the http request!!!) Try to visit the url before. This is an example of the output:

```
<?xml version="1.0"?>
<data status="ok"><forma tipo="rectangulo" x1="110" y1="77" x2="110" y2="77" col="FF000000"/><forma
tipo="elipse" x1="67" y1="144" x2="136" y2="238" col="FF000000"/></data>
```

And this is NOT a SOP issue:
- the server sends the correct headers (or would send the correct headers if it got the request in the first place) to allow the crossdomain
- i tested it on the same domain
- the following DOES work:

```
  String[] textlines=loadStrings(url);
  String text=join(textlines,"\n");
  XML xml=parseXML(text);
  println(xml);
```

- I tested it in Firefox with Httpfox and no http request is done at all!!

No error appears on the JavaScript console.

At first i thought it was a synchronous vs asynchronous issue (Processing.js not emulating synchronous/blocking behavior as it does with loadStrings) but if it was the case, the request would be made anyway, even though the response would be received too late.

The whole thing is really crazy.

State: new
Milestone: 1.4.2
View this ticket online: http://processing-js.lighthouseapp.com/projects/41284/tickets/2002-loadXML-is-broken

Stop being notified of this ticket's changes: http://processing-js.lighthouseapp.com/watching/b130dd3bf51059ef3022c0d08b9bdde8c4829815
Update your Profile: http://processing-js.lighthouseapp.com/profile
Support: support@lighthouseapp.com or http://help.lighthouseapp.com

---

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 3:51 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 06:51 AM

The problem must actually be in the XML constructor, as I'm realising loadXML() is just a wrapper for it:

```
p.loadXML = function(uri) {
  return new XML(p, uri)
```

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 3:58 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 06:58 AM

Oh my god!!! I can't believe in these days forms in a bug tracker still f**k up html code in the input text. Unbelievable.

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 4:04 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 07:04 AM

I've found the error in the code.

Processing.js relies on the FILE EXTENSION to determine whether to load the data or parse it as a string!!!!!!!!!!

```
XMLElement.prototype = {
  parse: function(textstring) {
    var xmlDoc;
    try {
      var extension = textstring.substring(textstring.length - 4);
      if (extension === ".xml" || extension === ".svg") textstring = ajax(textstring); /////// HERE!!!
      xmlDoc = (new DOMParser).parseFromString(textstring, "text/xml");
      var elements = xmlDoc.documentElement;
      if (elements) this.parseChildrenRecursive(null, elements);
      else throw "Error loading document";
      return this
    } catch(e) {
      throw e;
    }
```

If the url doesn't end with .xml or .svg, then the url is parsed as if it was the data!!!!

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 4:23 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 07:23 AM

Actually it seems to be a very bad idea to delegate to the XMLElement.parse() function the decision whether to parse the input as data or consider it as an url and load it. It's simply not its job. Parse() should _always_ parse the input as data. The XML constructor should be able to discriminate (correctly, not based on ridiculous criteria such as the last four characters of the input string). And loadXML() should _always_ take the parameter as an url and attempt to load it, no matter what.

The parseXML() function for example ends up relying on XMLElement.parse(), so it will run into the exact opposite problem than loadXML() (though in that case it will result in not getting an error when it is expected to get en error, rather than getting an error when not expected).

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 4:24 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 07:24 AM

I meant "it seems to me", not "it seems to be".
Can't even edit my comment, another great feature of Github.

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 4:41 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 07:41 AM

Oh, I mentioned parseXML(), now I realise it doesn't even exist in Processing.js. See #2003.

[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sun, Mar 17, 2013 at 5:19 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 17th, 2013 @ 08:19 AM

I think I've also found the quick fix for loadXML() to work:

Replace

```
  p.loadXML = function(uri) {
    return new XML(p, uri)
  };
```

with

```
  p.loadXML = function(uri) {
    return new XML(uri)
  };
```

However the more profound issues described above need to be fixed.

Also be aware that Processing 2.08b has changed the API of the XML constructor, again.

I'm tempted to open a bug report on their bugtracker: "keep calm and speek with the Processing.js team".
[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Sat, Mar 30, 2013 at 9:59 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above here
==
matteosistisette updated this ticket at March 30th, 2013 @ 12:59 PM

> but this one will be relatively low-priority given that people can trivially set up a route for a url that looks like an .xml file link to point to .php or .py or .asp generator pages.

OMG PLEASE RECONSIDER!!!!!! You are making the assumption that the person writing the sketch always owns (or has control of) the server from where the data is pulled!!!!!!! AND that the url from which the data is pulled is there for the sketch's exclusive use (so one can always change its url without affecting any other clients).

IF you have to keep parsing the string input to decide whether to take it as XML data or as the url of the source of the data AT THE VERY LEAST the parsing MUST be done with more sensible criteria. And this, at least, should have high priority (at least higher than "won't fix").

Right now the algorithm is more or less:
- if it ends with ".xml" or ".svg", then it's the url/filename of the source of the data
- else it is an xml string
(it's really embarassing)

Consider this alternative, very very far from perfect but which would be already a HUGE improvement at zero cost:
- if it contains any "<" and/or ">" then it is an xml string
- else it is a url or filename of the source of the data.
Still wrong behavior but waaaaaaaaaaaaaaaaaaaaaaay better, and I can't see a single drawback over the current behavior.

Now (sorry for asking) did you consider this comment of mine carefully?:
"""
Actually it seems to be a very bad idea to delegate to the XMLElement.parse() function the decision whether to parse the input as data or consider it as an url and load it. It's simply not its job. Parse() should always parse the input as data. The XML constructor should be able to discriminate (correctly, not based on ridiculous criteria such as the last four characters of the input string). And loadXML() should always take the parameter as an url and attempt to load it, no matter what.
The parseXML() function for example ends up relying on XMLElement.parse(), so it will run into the exact opposite problem than loadXML() (though in that case it will result in not getting an error when it is expected to get en error, rather than getting an error when not expected).
"""

Are you sure the PARSE FUNCTION ITSELF really needs to discriminate for backward compatibility? Isn't it the XML CONSTRUCTOR that needs that?

Anyway, this _may_ become low priority (provided that the parsing/choice is done with sensible criteria) IF:
1 - the parseXML() method is implemented, and guaranteed to always parse the input as xml string,
2 - and loadXML() is guaranteed to always take the input as the url/filename

By the way, I think I provided the fix for point (2) :
"""
Replace
p.loadXML = function(uri) { return new XML(p, uri) }; with
p.loadXML = function(uri) { return new XML(uri) };
"""

State: wont-fix
[Quoted text hidden]

Lighthouse <no-reply@lighthouseapp.com> Tue, Mar 11, 2014 at 9:16 AM
Reply-To: ticket+processing-js.41284-2002-4xny6hcm@lighthouseapp.com
To: nihongo@gmail.com

// Add your reply above

matteosistisette updated ticket #2002 on Mar 11th, 2014 at 12:16 PM

> Have you even read my last comment?
>
> I can't believe you are keeping this bug beacuse of backward compatibility while you don't care to break compatibility with plain Processing. This breaks sketches that work just fine in Processing non-js.
>
> And if you really need to keep the embarassing decide-based-on-string-parsing behavior, you should at least improve the algorithm that decides whether a string is an url or a filename, so that the decision is made in a more sensible way (see my comment above). The current criteria is way too poor. This would be a trivial "fix" and has no drawbacks in terms of BC.

- **State:** wont-fix
- **Milestone:** 1.4.2

Update this ticket by email: `[state:new]`
Update your profile, or stop being notified
Support: email support or ask for help