

H.761 Support of a DOM Level 4 Core specification in NCLua scripts

Alan L. V. Guedes
TeleMídia - PUC-Rio
alan@telemidia.puc-rio.br

Sergio Colcher
Informatics Department - PUC-Rio
colcher@inf.puc-rio.br

ABSTRACT

Web developers commonly run HTML template engines at the client-side using JavaScript. One upper hand is to give more development flexibility comparing to the server-side. In this way, we may cite web frameworks webcomponents, ReactJS, Vue, and others. Those frameworks use the DOM API to access the template data inserted in the HTML document. Then the same DOM API is used to insert more HTML code. Currently, NCL not support access to the current document. But it can modify the document by NCL Editing commands in NCLua scripts. We propose NCLua scripts partially support the DOM Code API.

KEYWORDS

NCL, Ginga

1 BACKGROUND

Web developers commonly run HTML template engines at the client-side using JavaScript. One upper hand is to give more development flexibility comparing to the server-side. In this way, we may cite web frameworks webcomponents, ReactJS, Vue, and others. Those frameworks use the DOM API to access the template data inserted in the HTML document. Then the same DOM API is used to insert more HTML code.

Currently, H.761 NCL[1] do not support access to the current document. But it can modify the document by NCL Editing commands in NCLua scripts.

2 PROPOSAL

We propose NCLua scripts partially support the DOM API. More precisely, we propose the NCLua support the document object from the DOM Core API. We details our document object in what follows. This element is based on the lua-gumbo¹. Its structure and API of this tree mostly follows the DOM Level 4 Core specification².

Properties:

- **documentElement**: The root Element of the document (i.e. the `<html>` element).
- **head**: The `<head>` Element of the document.
- **body**: The `<body>` Element of the document.
- **title**: A string containing the document's title (initially, the text contents of the `<title>` element in the document markup).

¹<https://craigbarnes.gitlab.io/lua-gumbo/#document>

²<https://dom.spec.whatwg.org/>

- **forms**: An ElementList of all `<form>` elements in the document.
- **images**: An ElementList of all `` elements in the document.

Methods:

- **getElementById**(elementId): Returns the first Element node in the tree whose id property is equal to the elementId string.
- **getElementsByTagName**(tagName): Returns an ElementList containing every child Element node whose locaName is equal to the given tagName argument.
- **getElementsByClassName**(classNames): Returns an ElementList containing every child Element node that has all of the given class names. Multiple class names can be specified by passing a string with several names separated by whitespace.

3 USE CASE

The following code uses the `getElementById` to get a media object. Then it uses an NCLua command to add a property to it at runtime.

```
1 local foo = document:getElementById("template")
2 local targetid = foo.childNodes[1].id
3 addInterface (baseId, documentId, targetid,
4 "<property name=\"background\" value=\"green\">")
```

Listing 1: NCL code fragment using the proposed approach.

REFERENCES

- [1] ITU. 2009. *H.761: Nested Context Language (NCL) and Ginga-NCL for IPTV Services*. Technical Report. ITU, Geneva, Switzerland. <https://www.itu.int/rec/T-REC-H.761>