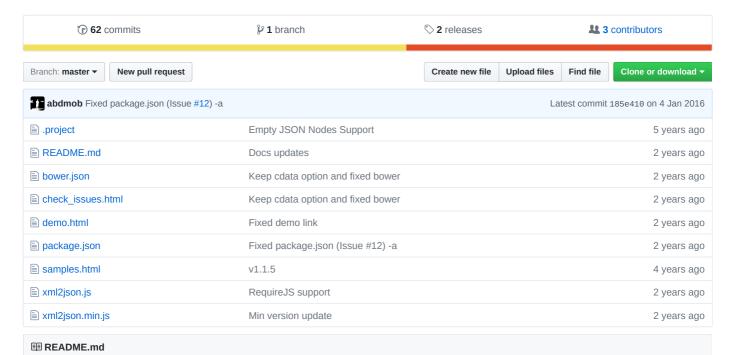


x2js - XML to JSON and back for JavaScript



x2js - XML to JSON and vice versa for JavaScript

This library provides XML to JSON (JavaScript Objects) and vice versa javascript conversion functions. The library is very small and has no any dependencies.

API functions

- new X2JS() to create your own instance to access all library functionality
- new X2JS(config) to create your own instance with additional config
- <instance>.xml2json
 Convert XML specified as DOM Object to JSON
- <instance>.json2xml Convert JSON to XML DOM Object
- <instance>.xml_str2json
 Convert XML specified as string to JSON
- <instance>.json2xml_str Convert JSON to XML string
- <instance>.asArray
 Utility function to work with a JSON field always in array form
- <instance>.asDateTime
 Utility function to convert the specified parameter from XML DateTime to JS Date
- <instance>.asXmlDateTime
 Utility function to convert the specified parameter to XML DateTime from JS Date or timestamp

Config options

- escapeMode : true|false Escaping XML characters. Default is true from v1.1.0+
- attributePrefix: "<string>" Prefix for XML attributes in JSon model. Default is "_"

- arrayAccessForm: "none"|"property" The array access form (none|property). Use this property if you want X2JS generates additional property asArray to access in array form for any XML element. Default is none from v1.1.0+
- emptyNodeForm: "text"|"object" Handling empty nodes (text|object) mode. When X2JS found empty node like it will be transformed to test: "for 'text' mode, or to Object for 'object' mode. Default is 'text'
- enableToStringFunc : true|false Enable/disable an auxiliary function in generated JSON objects to print text nodes with text/cdata. Default is true
- arrayAccessFormPaths: [] Array access paths use this option to configure paths to XML elements always in "array form". You can configure beforehand paths to all your array elements based on XSD or your knowledge. Every path could be a simple string (like 'parent.child1.child2'), a regex (like /.*.child2/), or a custom function. Default is empty
- skipEmptyTextNodesForObj : true|false Skip empty text tags for nodes with children. Default is true.
- stripWhitespaces : true|false Strip whitespaces (trimming text nodes). Default is true.
- datetimeAccessFormPaths: [] DateTime access paths. Use this option to configure paths to XML elements for XML datetime elements. You can configure beforehand paths to all your datetime elements based on XSD or your knowledge. Every path could be a simple string (like 'parent.child1.child2'), a regex (like /.*.child2/), or a custom function. Default is empty.
- useDoubleQuotes: true|false Use double quotes for output XML formatting. Default is false.
- xmlElementsFilter: [] Filter incoming XML elements. You can pass a stringified path (like 'parent.child1.child2'), regexp or function
- jsonPropertiesFilter: [] Filter JSON properties for output XML. You can pass a stringified path (like 'parent.child1.child2'), regexp or function
- keepCData: true|false If this property defined as false and an XML element has only CData node it will be converted to text without additional property "__cdata". Default is false.

Online demo

JSFiddle at http://jsfiddle.net/abdmob/gkxucxrj/1/

Basic Usage

XML to JSON

```
// Create x2js instance with default config
var x2js = new X2JS();
var xmlText = "<MyRoot><test>Success</test><test2><item>val1</item><item>val2</item></test2></MyRoot>";
var jsonObj = x2js.xml_str2json( xmlText );
```

JSON to XML

```
// Create x2js instance with default config
var x2js = new X2JS();
var jsonObj = {
    MyRoot : {
        test: 'success',
        test2 : {
            item : [ 'val1', 'val2' ]
        }
    }
};
var xmlAsStr = x2js.json2xml_str( jsonObj );
```

Working with arrays

Configure XML structure knowledge beforehand

```
var x2js = new X2JS({
    arrayAccessFormPaths : [
    "MyArrays.test.item"
```

```
]
});

var xmlText = "<MyArrays>"+
    "<test><item>success</item><item>second</item></test>"+
    "</MyArrays>";

var jsonObj = x2js.xml_str2json( xmlText );
console.log(jsonObj.MyArrays.test2.item[0]);
```

Or using the utility function

Working with XML attributes

Accessing to XML attributes

```
// Create x2js instance with default config
var x2js = new X2JS();

var xmlText = "<MyOperation myAttr='SuccessAttrValue'>MyText</MyOperation>";
var jsonObj = x2js.xml_str2json( xmlText );

// Access to attribute
console.log(jsonObj.MyOperation._myAttr);

// Access to text
console.log(jsonObj.MyOperation.__text);
// Or
console.log(jsonObj.MyOperation.toString());
```

Configuring a custom prefix to attributes

```
var x2js = new X2JS({
    attributePrefix : "$"
});

var xmlText = "<MyOperation myAttr='SuccessAttrValue'>MyText</MyOperation>";

var jsonObj = x2js.xml_str2json( xmlText );

// Access to attribute
console.log(jsonObj.MyOperation.$myAttr);
```

Working with XML namespaces

Parsing XML with namespaces

Creating JSON (for XML) with namespaces (Option 1)

Creating JSON (for XML) with namespaces (Option 2)

Working with XML DateTime

Configuring it beforehand

```
var x2js = new X2JS({
    datetimeAccessFormPaths : [
        "MyDts.testds" /* Configure it beforehand */
    ]
});

var xmlText = "<MyDts>"+
        "<testds>2002-10-10T12:00:00+04:00</testds>"+
        "</MyDts>";
var jsonObj = x2js.xml_str2json( xmlText );
```

Or using the utility function

Networking samples

Parsing AJAX XML response (JQuery sample)

```
$.ajax({
    type: "GET",
    url: "/test",
    dataType: "xml",
    success: function(xmlDoc) {
       var x2js = new X2JS();
       var jsonObj = x2js.xml2json(xmlDoc);
    }
});
```

Loading XML and converting to JSON

```
function loadXMLDoc(dname) {
    if (window.XMLHttpRequest) {
        xhttp=new XMLHttpRequest();
    }
    else {
        xhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xhttp.open("GET", dname, false);
    xhttp.send();
    return xhttp.responseXML;
}

var xmlDoc = loadXMLDoc("test.xml");
    var x2js = new X2JS();
    var jsonObj = x2js.xml2json(xmlDoc);
```

© 2017 GitHub, Inc. Terms Privacy Security Status Help

()

Contact GitHub API Training Shop Blog About