

XMLHandler Class

Private beta

Core functionality

Class

XMLHandler(xmlUrl:String);

Here you set the basic URL, where the to be queried XMLfile is to be found.

Functions

addRequest(new Array(operation, variable));

To add requests and services, this will append (?)“operation = variable”(&) to the xmlUrl, you can add as many as you need.

addMultipleRequests(request:Array);

With the addMultipleRequests you can send in a Full array, which on every array item has an array item as **new Array(service, operator);**

appendUrl(appendage:String);

If the XML source has a different way of creating a XML request without using operations and variables, but its just an appendage to an URL you can choose to append an url.

```
lastFMxml.appendURL("/artist/thebeatles/similar.xml");
```

handleXML(sliceAt:String):

When the url is as you wished for you can handle the XML. This means it will auto-parse the set xmlUrl. When setting sliceAt, it will slice and store the url from a specific child such as Item. So the results in the XMLhandler are the Item nodes and their children. If you don't set a sliceAt, the handleXML will automatically set the sliceAt standard to “Item”.

getSliceLength;

returns the amount of slices stored in the array

getParents();

Returns the names of all parent nodes in the handled xml which contain a child. It returns their name and their related parent.

getChildren(parent:String, sliceid:Number);

When left empty it returns all the names of the children nodes in the handled xml which contain a value or an attribute. For instance this can return: artist, title, format, tracks;

When parent is used it returns all the individual items stored in the XML as parentname, value and id. You have the optional choice to put in a specific child name to see which values are stored at a specific id. For instance (artist) will return an array like this (1, Autechre), (2, Smashing Pumpkins), (3, Squarepusher) etc.

getItemValue(parentName:String, childname:String, sliceid:String);

This will return one specific value can be string or array, belonging to one item and its child. For instance ("ItemAttributes", "Title", 8) will return the title value of item 8. You can leave sliceid blank, than it will return the title of every item stored in the array.

newRequests();

Empties the current values stored in the array and the request array, therefore being able to run a new xml handle.

Events

Event:HANDLE_COMPLETE

Fires when handleXML is completed, before that you cannot use the get functionality.