art : script

jart.handlers.special.JArtScripting

art:script

Integration vom EcmaScript. Ermöglicht die Erstellung von Logikblöcken mit Zugriff auf alle Java Libraries ohne der Erstellung einer extra Klasse. Die Scripts werden compiliert im Cache gehalten und sind somit auch von der Performance den Java Klassen nicht signifikant Unterlegen. Es kann entweder eine Script Datei (über @script) oder auch ein inline Script (unter Verwendung von @script-id) angegeben werden. Die Handler Klasse steht dabei über $ zur Verfügung.

Steht auch als Extension Function script zur Verfügung

Speziell Funtionsaliase:

- $.E(*name*) >> new org.jdom.Element(*name*)
- $.add(*element*) >> $.outNode.addContent(*element*)

Einbindung von java Packages: importPackage(Packages.*package name*);

- importPackage(Packages.jart);

Attribute

Globale Attribute

@script

Dieses Attribut wird aus Sicherheitsgründen nicht evaluiert und kann daher nicht dynamisch gesetzt werden!

Name der Script Datei. Die Dateiendung bestimmt den Script Typ. (z.b .js für JavaScript).

@script-id

Dieses Attribut wird aus Sicherheitsgründen nicht evaluiert und kann daher nicht dynamisch gesetzt werden!

Innehalb des JART Files und aller dort geladenen Includes Eindeutige ID für das inline Script (wird für die identifizierung des compilierten cache benötigt). Die Endung bestimmt den Script Typ. (z.b .js für JavaScript)

@include

Dieses Attribut wird aus Sicherheitsgründen nicht evaluiert und kann daher nicht dynamisch gesetzt werden!

Beistrichgetrennte Liste der zu Includierenden Scriptdateien,

Beispiele

```
*******************************************************************************
*** JART Code:
*******************************************************************************
<art:script script="sc01.js" for-num="10"/>


*******************************************************************************
*** Script Code der Datei sc01.js:
*******************************************************************************
var forNum = parseInt($.getAtt("for-num"));

for(var i = 0; i < forNum; i++){
    var e = $.E("test-node-x" + i);
    $.add(e);
}

var els = $.filter.selectNodes("/data/*", $.outNode);

for(var i = 0; i < els.size(); i++){
    var e = els.get(i);
    e.setAttribute("checked", "node: " + e.getName());
}


*******************************************************************************
*** Ergebniss:
*******************************************************************************
<data>
 <test-node-x0 checked="node: test-node-x0"/>
 <test-node-x1 checked="node: test-node-x1"/>
 <test-node-x2 checked="node: test-node-x2"/>
 <test-node-x3 checked="node: test-node-x3"/>
 <test-node-x4 checked="node: test-node-x4"/>
 <test-node-x5 checked="node: test-node-x5"/>
 <test-node-x6 checked="node: test-node-x6"/>
 <test-node-x7 checked="node: test-node-x7"/>
 <test-node-x8 checked="node: test-node-x8"/>
 <test-node-x9 checked="node: test-node-x9"/>
</data>



*******************************************************************************
*** Inline Script:
*******************************************************************************
<art:script script-id="test01.js">
   var e = $.E("hallo-c");
   $.add(e);
</art:script>
```