

XML mit JDito

AID 071 DE



© 2014 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: MW, KN. Version 10.2. Zuletzt geändert 19.06.2017

| Version | Änderungen |
|---------|--|
| 10.2 | Anpassung der Formatierungen |
| 10.1 | Anpassung der Beispiele |
| 10.0 | Anpassung an ADITO4.6 |
| 2.0 | Letzter Stand vor Übernahme in Versionierung |

Inhaltsverzeichnis

| | | |
|-----------|--------------------------------------|----------|
| 1. | XML mit Rhino und JDito | 4 |
| 1.1. | XML | 4 |
| 1.2. | XML schreiben | 4 |
| 1.2.1. | Dateien schreiben | 7 |
| 1.2.2. | XML-Struktur erzeugen | 4 |
| 1.3. | XML lesen | 7 |
| 1.3.1. | Dateien lesen | 7 |
| 1.3.2. | Informationen aus XML holen | 7 |
| 1.4. | Tabelle aus XML füllen | 8 |

1. XML mit Rhino und JDito

Dieses Dokument beschreibt in kurzer Form die Verwendung von XML in JavaScript und ADITO.

1.1. XML

- wohlgeformt
- valide
- Elemente und Attribute
- Baumstruktur, Root-Element

1.2. XML schreiben

1.2.1. XML-Struktur erzeugen

```
Ziel:  
<course>  
<title>XML mit Rhino</title>  
<trainer>  
<firstname>Cloud</firstname>  
<name>Strife</name>  
</trainer>  
<participants>  
<participant id="1">  
<title>Herr</title>  
<name>Wallace</name>  
<firstname>Barret</firstname>  
</participant>  
<participant id="2">  
<title>Frau</title>  
<name>Lockheart</name>  
<firstname>Tifa</firstname>  
</participant>  
<participant id="3">  
<title>Herr</title>  
<name>Gainsborough</name>  
<firstname>Aeris</firstname>  
</participant>  
<participant id="4">
```

```

<title>Frau</title>
<name>Kisaragi</name>
<firstname>Yuffie</firstname>
</participant>
<participant id="5">
<title>Herr</title>
<name>Valentine</name>
<firstname>Vincent</firstname>
</participant>
<participant id="6">
<title>Frau</title>
<name>Turk</name>
<firstname>Jenova</firstname>
</participant>
</participants>
</course>

```

1.2.1.1. Root-Element erzeugen:

```
var course = <course/>;
```



Vor dem Start-Tag darf kein Whitespace (Leerzeichen, Zeilenumbruch etc.) vorkommen!

1.2.1.2. Einmaliges Element der ersten Ebene erzeugen:

```
course.title = "XML mit Rhino";
```

1.2.1.3. Element der ersten Ebene mit einmaligen Elementen der zweiten Ebene:

```
course.trainer.firstname = "Cloud";
course.trainer.name = "Strife";
```

1.2.1.4. Element der ersten Ebene mit mehrmals auftretenden Elementen der zweiten Ebene:

Möglichkeit 1:

```

course.participants.participant = "";
course.participants.participant.title = "Herr";
course.participants.participant.name = "Wallace";
course.participants.participant.firstname = "Barrett";
course.participants.participant[1] = "";
course.participants.participant[1].title = "Frau";
course.participants.participant[1].name = "Lockheart";

```

```
course.participants.participant[1].firstname = "Tifa";
```

Möglichkeit 2:

```
course.participants[0] = "";  
course.participants[0].participant.title = "Herr";  
course.participants[0].participant.name = "Wallace";  
course.participants[0].participant.firstname = "Barret";  
course.participants[1] = "";  
course.participants[1].participant.title = "Frau";  
course.participants[1].participant.name = "Lockheart";  
course.participants[1].participant.firstname = "Tifa";
```

Möglichkeit 3:

```
course.participants.participant = "";  
course.participants.participant[0] = "";  
course.participants.participant[0].title = "Herr";  
course.participants.participant[0].name = "Wallace";  
course.participants.participant[0].firstname = "Barett";  
course.participants.participant[1] = "";  
course.participants.participant[1].title = "Frau";  
course.participants.participant[1].name = "Lockheart";  
course.participants.participant[1].firstname = "Tifa";
```

Attribute schreiben:

```
course.participants.participant = "";  
course.participants.participant[0] = "";  
course.participants.participant[0].@id = "1";  
course.participants.participant[0].title = "Herr";  
course.participants.participant[0].name = "Wallace";  
course.participants.participant[0].firstname = "Barret";  
course.participants.participant[1] = "";  
course.participants.participant[1].@id = "2";  
course.participants.participant[1].title = "Frau";  
course.participants.participant[1].name = "Lockheart";  
course.participants.participant[1].firstname = "Tifa";
```

1.2.2. Dateien schreiben

```
var details = new Array();
details[0] = "c:/training.xml"; //Pfad zur Datei
details[1] = course; // Daten
swing.doClientIntermediate(swing.CLIENTCMD_STOREDATA, details);
```

1.3. XML lesen

1.3.1. Dateien lesen

```
var filename = "c:/training.xml";
var daten = swing.doClientIntermediate(swing.CLIENTCMD_GETDATA, new
Array(filename));
var course = new XML(daten);
```

1.3.2. Informationen aus XML holen

Gib mir den Titel des Kurses:

```
question.showMessage(course.title);
```

Gib mir alle Elemente unter course:

```
question.showMessage(course.*);
```

Gib mir eine Liste mit den Namen aller Teilnehmer:

```
question.showMessage(course.participants.participant.name);
```

Gibt mir eine Liste mit den Namen aller männlichen Teilnehmer:

```
question.showMessage(course.participants.participant.(title=="Herr")
.name);
```

Gib mir den Namen des ersten Teilnehmers in der Liste:

```
question.showMessage(course.participants.participant[0].name);
```

Gib mir den Namen des ersten männlichen Teilnehmers in der Liste:

```
question.showMessage(course.participants.participant.(title=="Herr")
[0].name);
```

Gib mir den Namen der Teilnehmer mit den IDs 4 und 6:

```
question.showMessage(course.participants.participant.(@id == 4 ||
@id == 6).name);
```

1.4. Tabelle aus XML füllen

Aufgabenstellung:

Fülle eine Tabelle mit allen Teilnehmern des Kurses.

Tabelle füllen:

```
var filename = "c:/training.xml";
var daten = swing.doClientIntermediate(swing.CLIENTCMD_GETDATA, new
Array(filename));
var course = new XML(daten);

var xmldata = new Array();
var j = 0;
for each (i in course.participants.participant)
{
var id = i.@id;
xmldata[j] = new Array();
xmldata[j][0] = id;
xmldata[j][1] = id;
xmldata[j][2] = i.name;
j++;
}

var r = new Array();
for (var k=0; k<xmldata.length; k++)
{
r[k] = new Array(xmldata[k][0], xmldata[k][1], xmldata[k][2]);
}

result.object(r);
```

Mehrstufige Auswahl, zweite Tabelle:

```
var filename = "c:/training.xml";
var daten = swing.doClientIntermediate(swing.CLIENTCMD_GETDATA, new
Array(filename));
var course = new XML(daten);

var selected = text.decodeFirst(vars.getString("$comp.Tabelle_0"));
```



```
var xmldata = new Array();
var j = 0;
for each (i in course.participants.participant.(@id == selected))
{
var id = i.@id;
xmldata[j] = new Array();
xmldata[j][0] = id;
xmldata[j][1] = i.firstname;
j++;
}

var r = new Array();
for (var k=0; k<xmldata.length; k++)
{
r[k] = new Array(xmldata[k][0], xmldata[k][1]);
}

result.object(r);
```