

27.5.0 XML-RPC

27.5.0.1 Was ist XML-RPC?

- XML-RPC ist ein einfaches, sprachunabhängiges Protokoll für entfernte Funktionsaufrufe über ein lokales Netz oder das Internet, das auch in Gambas implementiert wurde.
- XML-RPC ermöglicht Funktionsaufrufe über das HTTP-Protokoll. Die Bezeichnung 'Remote Procedure Call' oder kurz RPC steht für entfernte Funktionsaufrufe.
- XML dient zur Kodierung der Übergabeparameter an die entfernten Prozeduren auf einem RPC-Server aber auch zur Kodierung der Funktionswerte. Daher der Name XML-RPC.
- Ein XML-RPC-Server stellt eine Schnittstelle für den Fern-Zugriff auf eine Liste von implementierten Funktionen bereit. Diese Liste kann von einem RPC-Client abgerufen werden.
- XML-RPC funktioniert nach dem Client-Server-Modell: Ein Funktionsaufruf von einem XML-RPC-Client wird per HTTP-POST an einen XML-RPC-Server geschickt (Request), der einen Service über eine Liste von implementierten Funktionen anbietet, die von einem Client mit passenden Argumenten aufgerufen werden können. Beachten Sie, dass Argumente in XML-RPC keinen Namen erhalten. Allein die Position eines Arguments entscheidet über dessen Bedeutung! Der Server wertet das im Body (HTTP-POST) enthaltene XML-Dokument aus und verwendet dessen Inhalt als Argumente für den Aufruf der gewünschten Funktion. Der Funktionswert wird wieder in XML 'verpackt' und zum rufenden XML-RPC-Client übertragen (Response). Neben nativen Datentypen wie DateTime (ISO_8601: https://de.wikipedia.org/wiki/ISO_8601), Boolean, Integer, String, Float oder Base64 werden auch Arrays (RpcArray) und der Datentyp Struct (RpcStruct) von der Komponente gb.xml.rpc unterstützt.

Ein vollständiges Beispiel zur Einführung in die Funktionsweise von XML-RPC stellt Thomas Bayer in einem anwendungsbezogenen Artikel auf <https://www.oio.de/public/xml/xml-rpc.htm> vor.

Die folgenden Hinweise dürfen in diesem Buch-Kapitel nicht fehlen: Die Komponente gb.xml.rpc nutzt einen eigenen, internen Mini-HTTP-Server, um XML-RPC-Dienste anzubieten. Die Komponente ist *nicht* dafür ausgelegt, den internen Mini-HTTP-Server in der gleichen Weise zu verwenden, wie es zum Beispiel die PHP-Webdienste tun. RPC ist daher kein Web-Service im üblichen Sinn – wohl aber ein Service, den ein lokaler HTTP-Server bietet. Ein XML-RPC-Server ist ein HTTP-Server, genauso wie zum Beispiel Apache2 oder Lighttpd. Im Gegensatz zu den vorher genannten Servern laufen XML-RPC-Server normalerweise nicht auf dem Port 80 und liefern auch keine Webseiten aus.

Die Komponente gb.xml.rpc besitzt die Klassen RpcServer, RpcClient, RpcFunction, RpcArray, RpcStruct und RpcType. Die Klassen *RpcServer* und *RpcClient* werden in eigenen Kapiteln beschrieben. Dort werden auch die relevanten Projekte vorgestellt.

27.5.0.2 Client-Anfrage (Request) und Server-Antwort (Response)

Die Client-Anfrage (Request) und die Server-Antwort (Response) enthalten jeweils einen mehrzeiligen XML-Abschnitt, in dem u.a. die Argumente (Request) oder der Funktionswert (Response) stehen. Für das Projekt mit den Argumenten *summand1=12* und *summand2=8* sowie dem Funktionswert *summe = 20* sehen die XML-Abschnitte so aus:

Request-Ausschnitt:

```
POST /RPC HTTP/1.1
...
Content-Length: 230
Content-Type: text/xml
<?xml version="1.0"?>
<methodCall>
  <methodName>add2integer</methodName>
  <params>
    <param>
      <value><int>12</int></value>
    </param>
    <param>
      <value><int>8</int></value>
    </param>
  </params>
</methodCall>
```

Response-Ausschnitt:

```
...
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <int>20</int>
      </value>
    </param>
  </params>
</methodResponse>
```

27.5.0.3 RPC-Funktion

Ein Objekt der Klasse `RpcFunction` (gb.xml.rpc) können Sie erzeugen:

```
Dim hRpcFunction As RpcFunction
hRpcFunction = New RpcFunction(Name As String, DataTypes As Integer[], Ret As Integer)
```

- Name – Name der Funktion
- DataTypes – Array der Datentypen der Argumente der Funktion
- Ret – Datentyp des Funktionswertes

Die Klasse `RpcFunction` können Sie wie ein Read-Only-Array verwenden:

```
Dim hRpcFunction As RpcFunction
Dim anInteger As Integer
anInteger = hRpcFunction [ Index As Integer ]
```

27.5.0.3.1 Eigenschaften

In der folgenden Tabelle finden Sie die Eigenschaften der Klasse `RpcFunction`:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der Argumente der RPC-Funktion an.
Help	String	Legt den Hilfetext für die RPC-Funktion fest oder gibt den Hilfetext zurück.
MethodName	String	Gibt den Namen der Methode zurück.
ReturnType	Integer	Gibt den Datentyp des Funktionswertes zurück → Kapitel 27.5.0.6.2

Tabelle 27.5.0.3.1 : Eigenschaften der Klasse `RpcFunction`

27.5.0.4 RPC-Array

Ein Objekt der Klasse `RpcArray` (gb.xml.rpc) können Sie erzeugen:

```
Dim hRpcArray As RpcArray
hRpcArray = New RpcArray()
```

Die Klasse `RpcArray` können Sie wie ein Read-Write-Array verwenden:

Lesen:

```
Dim hRpcArray As RpcArray
Dim aVariant As Variant
aVariant = hRpcArray [ Index As Integer ]
```

Schreiben:

```
Dim hRpcArray As RpcArray
Dim aVariant As Variant
hRpcArray [ Index As Integer ] = aVariant
```

Die Klasse `RpcArray` verfügt nur über die Nur-Lese-Eigenschaft `Count` vom Datentyp `Integer`, welche die Anzahl der Elemente im Array zurück gibt.

27.5.0.4.1 Methoden

Hier finden Sie die Übersicht zu ausgewählten Methoden der Klasse *RpcArray*:

Methoden	Rückgabtyp	Beschreibung
Add(Value As Variant, DataType As Integer)	-	Fügt ein Element in das RPC-Array ein.
Clear()	-	Alle Elemente im RPC-Array werden gelöscht.
Copy()	RpcArray	Ein RPC-Array wird 1:1 kopiert und als Funktionswert zurückgegeben.
Pop()	Variant	Entfernt das letzte Element eines RPC-Arrays und gibt dieses Element als Funktionswert vom Typ Variant zurück. Wenn das originale RPC-Array leer ist, wird ein Fehler ausgelöst.
Push(Value As Variant, DataType As Integer)	-	Fügt ein Element am Ende eines RPC-Arrays ein.
Remove(Index As Integer)	-	Löscht das Element mit dem angegebenen Index im RPC-Array.
Reverse()	RpcArray	Die Reihenfolge der Array-Elemente wird invertiert – die Array-Elemente E0..En werden umsortiert auf En..E0. Das invertierte RPC-Array wird als Funktionswert zurückgegeben.

Tabelle 27.5.0.4.1 : Ausgewählte Methoden der Klasse RpcArray

27.5.0.5 RPC-Struct

Ein Objekt der Klasse *RpcStruct* (gb.xml.rpc) können Sie erzeugen:

```
Dim hRpcStruct As RpcStruct
hRpcStruct = New RpcStruct()
```

Die Klasse *RpcStruct* besitzt lediglich die Nur-Lese-Eigenschaft *Count* vom Datentyp Integer, welche die Anzahl der Felder in der Struktur zurück gibt.

27.5.0.5.1 Methoden

Hier finden Sie die Übersicht zu den sieben Methoden der Klasse *RpcStruct*:

Methoden	Rückgabtyp	Beschreibung
Add(Name As String, Value As Variant, DataType As Integer)	-	Es wird ein Feld mit dem angegebenen Feldnamen, dem Feld-Wert und der Angabe des passenden Datentyps in eine Struktur hinzugefügt.
Clear()	-	Es werden alle Felder in der Struktur gelöscht.
Copy()	RpcStruct	Eine Struktur wird 1:1 kopiert und als Funktionswert zurückgegeben.
Key(Index As Integer)	String	Es wird der Schlüssel zu dem gegebenen Feld-Index als Funktionswert zurückgegeben.
Value(Index As Integer)	Variant	Es wird der Feld-Wert zu dem gegebenen Feld-Index als Funktionswert zurückgegeben.
Remove(Index As Integer)	-	Löscht das Element mit dem angegebenen Index in der Struktur.
Reverse()	RpcStruct	Die Reihenfolge der Elemente wird invertiert – die Struktur-Elemente E0..En werden umsortiert auf En..E0. Die invertierte Struktur wird als Funktionswert zurückgegeben.
DataType(Index As Integer)	Integer	Definiert den Datentyp eines Feldes in der Struktur oder gibt dessen Datentyp zurück.

Tabelle 27.5.0.5.1 : Methoden der Klasse RpcStruct

27.5.0.6 RPC-Typ

Die Klasse `RpcType` (`gb.xml.rpc`) ist statisch und enthält neben 2 (statischen) Konvertierungsmethoden neun Konstanten. Die Klassen `RpcType` und `XmlRpc` besitzen die gleichen Methoden und Konstanten und können somit synonym verwendet werden.

27.5.0.6.1 Methoden

Hier finden Sie die Übersicht zu den zwei Methoden der Klasse `RpcType`:

Methode	Rückgabebetyp	Beschreibung
Function <code>ToString(type As Integer)</code>	String	Konvertiert einen Konstanten-Integer-Wert in seine (String-)Beschreibung.
Function <code>ToType(type As String)</code>	Integer	Konvertiert eine (String-)Beschreibung in seinen Konstanten-Wert.

Tabelle 27.5.0.6.1 : Methoden der Klasse `RpcType`

27.5.0.6.2 Übersicht zu den Konstanten der Klasse `RpcType`

- `xInteger` → 0
- `xBoolean` → 1
- `xDouble` → 2
- `xString` → 3
- `xBase64` → 4
- `xDate` → 5 (→ `dateTime.iso8601` – Beispiel: 2017-08-31T16:47+00:00 (in UTC))
- `xStruct` → 6
- `xArray` → 7
- `xVoid` → 8