

# HowTo: STARFACE XML-RPC nutzen

Ein STARFACE-Modul kann eine XML-RPC-Schnittstelle bereitstellen. Der grundsätzliche Gebrauch wird hier beschrieben.

## Aufruf und Zugang

Um Daten über die XML-RPC-Schnittstelle aufzurufen, existiert die "Seite" bzw. das Skript *xml-rpc* auf der STARFACE-Anlage. An diese Seite müssen die Anfragen gesendet werden.

```
http://<STARFACE-IP>/xml-rpc
```

## Authentifizierung

### Benutzerbasierte Authentifizierung

Üblicherweise wird für den Zugriff auf die XML-RPC-Schnittstelle die Authentifizierung über STARFACE-Benutzer durchgeführt. Nicht authentifizierter Zugriff wird verweigert. Die Zugangsdaten bestehend aus STARFACE-Login-ID und -Password müssen bei jeder Anfrage mitgesendet werden. Dazu wird ein zusätzlicher Parameter im HTTP-Aufruf angehängt.

```
http://<STARFACE-IP>/xml-rpc?de.vertico.starface.auth=<AUTHENTICATIONSTRING>
```

Als Authentifizierungswert wird eine eindeutige Zeichenfolge verwendet, die mittels eines MD5-Wertes von Login-ID, Asterisk (\*) und Passwort gebildet wird.

#### Beispiel

Für einen Benutzer mit der Login-ID "265" und dem Passwort "Geheim" an einer STARFACE-Anlage mit der IP "192.168.0.100" ergibt sich als Aufruf-URL für XML-RPC (pseudo-Code):

Bis STARFACE 6.4.2.12:

```
AUTH = md5("265"*"Geheim")  
http://192.168.0.100/xml-rpc?de.vertico.starface.auth=AUTH
```

Ab STARFACE 6.4.2.19:

```
NEWAUTH = "265":sha512("265"*sha512("Geheim").toLowerCase()).toLowerCase()  
http://192.168.0.100/xml-rpc?de.vertico.starface.auth=NEWAUTH
```

### Schlüsselbasierte Authentifizierung

Alternativ kann das Modul, wenn es vor allem für zentrale, programmatische Zugriffe genutzt wird, Authentifizierung über einen Schlüssel, einen pre-shared Key, anbieten. Dabei fällt die beschriebene Authentifizierung über STARFACE-Benutzer weg und wird über einen zusätzlichen Parameter im XML-RPC-Aufruf selber erweitert. Dieser heißt üblicherweise *auth*. Der pre-shared Key wird entweder im Modul einprogrammiert oder kann über die Modulkonfiguration vom Administrator gesetzt werden.

Die URL für den Aufruf entspricht damit der Basis-URL.

#### Beispiel

Für eine STARFACE-Anlage mit der IP "192.168.0.100" ergibt sich als Aufruf-URL für XML-RPC:

```
http://192.168.0.100/xml-rpc
```

## Ein- und Ausgabeparameter

Jede XML-RPC-Funktion kann beliebig viele Ein- und Ausgabeparameter bereitstellen. Der Reiter **XML-RPC** in der Modulkonfiguration beschreibt die Parameter der in diesem Modul zur Verfügung stehenden Funktionen. Es ist dabei darauf zu achten, dass die richtigen Typen verwendet werden. So werden STARFACE-Login-IDs als String übertragen (als numerische Werte übertragen würde "0001" die führenden Nullen verlieren).

Werden Parameter bei der Eingabe erwartet, so sind sie auch zu übergeben. Dabei erwartet die STARFACE immer genau einen Parameter vom Typ (xml-rpc)-struct, der wiederum die einzelnen Parameter mit ihren jeweiligen Typen beinhaltet. Bei der Funktionsrückgabe wird ebenfalls ein (xml-rpc)-struct übergeben, das die entsprechenden Rückgabewerte enthält.

Je nach eingesetzter Programmiersprache stellen die verschiedenen XML-RPC-Implementierungen diverse Automatismen bereit, die XML-RPC-Typen in der Sprache zugehörige Typen zu wandeln.

### Beispiel

Eine fiktive Funktion eines Moduls mit schlüsselbasierter Authentifizierung erwartet als Eingabe eine Gruppen-ID und einen booleschen Wert *testing*. Als Rückgabe wird *success*, also Erfolg angezeigt. Das vollständige XML für Methodenaufruf und -rückgabe sähe dann so aus:

#### Aufruf

```
<methodCall>
  <methodName>MODULKONFIGURATION.FUNKTIONSNAME</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>auth</name>
            <value>
              <string><PRE-SHARED-KEY></string>
            </value>
          </member>
          <member>
            <name>groupid</name>
            <value>
              <string>0001</string>
            </value>
          </member>
          <member>
            <name>testing</name>
            <value>
              <boolean>1</boolean>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

#### Rückgabe

```
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>success</name>
            <value>
              <boolean>0</boolean>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```