

White Paper

XMLRPC Schnittstelle GUTSCHEINmarketing



Version 1.0.0

- Stand 18.03.2004 -

Umfang: 31 Seiten

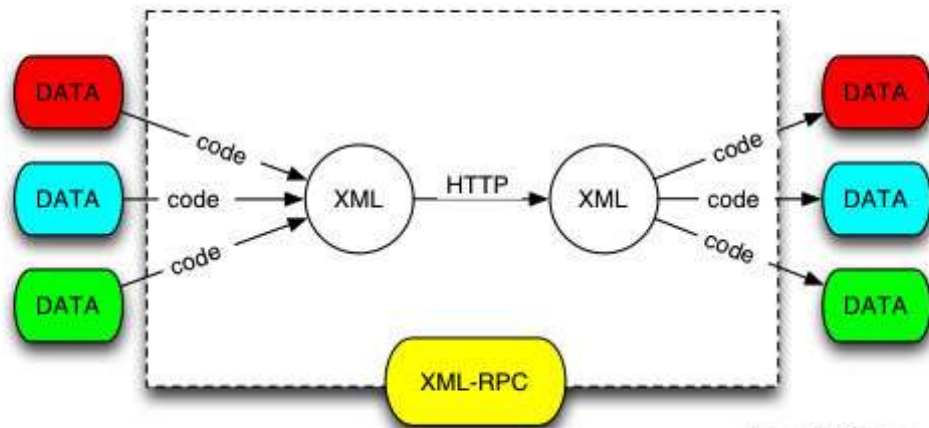
© copyright 4D f/x, 2004
4D f/x Kommunikationsservice
Hauptstr. 2
82229 Seefeld
Deutschland
www.4dfx.de

Inhaltsverzeichnis

Einleitung.....	5
1. Verbindungsaufbau.....	7
1.1. Voraussetzungen.....	7
1.1.1. URL.....	7
1.1.2. URI.....	7
1.1.3. Benutzerkennung.....	7
1.1.4. Passwort.....	8
1.1.5. Authentisieren.....	8
1.1.6. Client.....	8
1.2. Schematischer Aufbau.....	8
1.2.1. Verbindungskanal öffnen/initialisieren.....	8
1.2.1.1. Gültigkeit sessionid.....	9
1.2.2. Benutzer ausweisen/authentisieren.....	9
1.2.3. Schließen des Verbindungskanals.....	11
2. Methoden.....	13
2.1. Allgemeines Vorgehen.....	13
2.1.1. Schemata.....	13
2.1.2. Verbindungsaufbau.....	13
2.1.3. Authentisieren.....	13
2.1.4. Methodenaufrufe.....	13
2.1.5. Verbindung schließen.....	14
2.2. Methoden nur mit Authentisierung.....	14
2.2.1. isValidCoupon.....	14
2.2.2. getCouponValues.....	15
2.2.2.1. getCouponValues bei geldwertem Vorteil.....	16
2.2.2.2. getCouponValues bei Warenwert.....	18
2.2.2.3. getCouponValues bei Rabattgutschein.....	19
2.2.3. lockCoupon.....	20
2.2.4. unlockCoupon.....	22
2.3. Methoden ohne Authentisierung.....	24
2.3.1. system.listMethods.....	24
2.3.2. system.methodHelp.....	25
2.3.3. system.methodSignature.....	26
2.3.4. getLocalMessage.....	28
2.3.4.1. Übersetzung einer Fehlermeldung.....	28
3. Fehlercodes.....	30
3.1. Allgemeine XMLRPC-Fehler.....	30
3.2. Allgemeine XML-Fehler.....	30
3.3. XMLRPC-Service-Fehlermeldung.....	30
3.4. Fehlermeldung des XMLRPC-Client.....	30
4. Tabellenverzeichnis.....	31
Tabellenverzeichnis.....	31

Einleitung

Dieses Papier beschreibt die XMLRPC-Schnittstelle von www.gutscheinmarketing.de, die zur direkten Serverkommunikation dient. Dabei werden die notwendigen Schritte zu einem Verbindungsaufbau beschrieben. Zusätzlich werden die derzeit verfügbaren Methoden erörtert.



Source: JY Stervinou

Weitere Informationen zu XMLRPC finden sich unter <http://www.xmlrpc.com>. Die Spezifikationen sind zu finden unter <http://www.xmlrpc.com/spec>.

Um einen schnellen Einstieg zu gewährleisten, steht eine XMLRPC-Client-Klasse unter GPL zur Verfügung. Diese ist momentan in PHP verfügbar und ist nicht Gegenstand dieses Dokumentes. Eine Übertragung nach Java stellt kein Hindernis dar und wird vorgenommen, sobald eine entsprechende Resonanz vorhanden ist.

1. Verbindungsaufbau

1.1. Voraussetzungen

Um eine Verbindung zum XMLRPC-Service von www.gutscheinmarketing.de aufbauen zu können, bedarf es einer Zugangskennung und eines Passwortes. Diese können nach Belieben (so weit nicht bereits vergeben) gewählt werden und müssen mit www.gutscheinmarketing.de abgestimmt werden.

Im Gegensatz zu den sonst in anderen Quellen beschriebenen Verfahren, werden Kennung und Passwort **nicht** für eine http-Authentisierung verwendet. Aus Sicherheitsgründen findet eine Authentisierung mittels Zufallsschlüssel und beidseitig bekanntem Geheimnis statt.

Bei entsprechender Resonanz (und Sponsoring) ist eine weitere Möglichkeit mittels gnuPG implementierbar.

Jeder am Service teilnehmende Server erhält eine eindeutige Kennung sowie ein nur dem Service und dem jeweiligen Server bekanntes Passwort. Dieses ist durch geeignete Schutzmassnahmen auf den Servern sicher zu hinterlegen.

In diesem Papier wird für die Beispiele folgende Serverkennung mit dem Passwort gewählt:

Definition: **userid:** *test-server*

Definition: **password:** *supergeheim*

Das Passwort sollte auf dem Server als md5-Hash-Wert hinterlegt werden und nur berechtigten Personen und dem entsprechenden Prozeß zugänglich sein. Wir verwenden im Weiteren statt des Passwortes den md5-Hash-Wert:

Definition: **secret :=md5(password)**

1.1.1. URL

Um den XMLRPC-Service von www.gutscheinmarketing.de zu erreichen ist die URL

service.gutscheinmarketing.de

anzuwählen.

1.1.2. URI

Um die in diesem Papier beschriebenen Methoden aufrufen zu können, muss auf die URI

[/valueserver/index.php](#)

zugegriffen werden.

1.1.3. Benutzerkennung

Die Benutzerkennung (**userid**) ist eine eindeutige Adresse für www.gutscheinmarketing.de, die festlegt

1. welche Gutschein-Bereiche abgefragt werden können;
2. welche IP-Adressen für diese Abfrage autorisiert sind. Dabei können neben singulären Adressen, Adressebereiche sowie der gesamte, verfügbare Adressraum berücksichtigt werden.

1.1.4. Passwort

Beim Passwort (*password*) handelt es sich um ein Geheimnis, das nur beiden Servern, mit den entsprechend dazu berechtigten Personen, bekannt sein sollte. Dieses ist in der Form auf dem Server zu hinterlegen, das nur die berechtigte Person und der entsprechende Service darauf Zugriff haben. Es ist empfehlenswert, statt des Passwortes in Klartext, den md5-Hash-Wert (*secret*) des selbigen auf dem Server zu hinterlegen und den Klartext an anderer Stelle geeignet zu verwahren.

1.1.5. Authentisieren

Zur Authentisierung wird das beiden Seiten bekannte Geheimnis mit einem übermittelten Zufallswert geeignet verrechnet. Entspricht die Antwort nicht dem erwarteten Ergebnis wird eine Verbindung abgelehnt.

1.1.6. Client

Sämtliche Methoden-Aufrufe basieren auf einem vorhandenen Client-Objekt. Das Vorgehen dazu ist abhängig von der verwendeten Programmiersprache. Mögliche Lösungen finden sich unter <http://www.xmlrpc.com/directory/1568/implementations>.

1.2. Schematischer Aufbau

Um eine gültige Verbindung zum XMLRPC-Service von www.gutscheinmarketing.de aufzubauen, müssen folgenden Schritte eingehalten werden:

1. Verbindungskanal öffnen oder initialisieren mit Nennung des Benutzers.
2. Den Benutzer ausweisen oder authentisieren.
3. Schließen des Verbindungskanal.

1.2.1. Verbindungskanal öffnen/initialisieren

Zum Öffnen des Verbindungskanals wird die Benutzerkennung (*userid*) an die Methode *openChannel* unter der URI */valueserver/index.php* auf dem Server mit der URL *service.gutscheinmarketing.deals string* geschickt.

Ist die Benutzerkennung bekannt, wird eine *sessionid* als *string* zurückgeliefert.

**Diese *sessionid* ist Voraussetzung für jede weitere Kommunikation!
No *sessionid* no communication!**

Ist die Benutzerkennung nicht bekannt oder kann nicht von der vorhandenen IP-Adresse zugegriffen werden, erfolgt eine Fehlermeldung entsprechend XMLRPC-Standard.

Parameter	Value	Type
Methode	<i>openChannel</i>	
Werte	<i>userid</i>	<i>string</i>
Response success	<i>sessionid</i>	<i>string</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 1: openChannel(string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>openChannel</methodName>
  <params>
    <param>
      <value><string>userid</string></value>
    </param>
  </params>
</methodCall>
```

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>sessionid</string></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

1.2.1.1. Gültigkeit *sessionid*

Für das weitere Vorgehen ist eine gültige *sessionid* notwendig und Voraussetzung!

Eine *sessionid* ist für eine Zeitdauer von 1 Stunde gültig.

Daher wird dringend empfohlen, jede nicht mehr benötigte Verbindung zum Abschluss zu beenden! (s. 1.2.3 Schließen des Verbindungskanals, S. 11)

1.2.2. Benutzer ausweisen/authentisieren

Als nächstes muss der Benutzer ausgewiesen werden. Dazu wird der md5-verrechnete Wert der String-Konkatenation aus *sessionid* und *secret* (d.h. md5 (*password*)) zusammen mit der *sessionid* an die Methode *authenticate* unter der URI */valueserver/index.php* auf dem Server mit der URL *service.gutscheinmarketing.de* als *string* geschickt.

1. Verbindungsaufbau

Entspricht der übertragene Wert dem vom Server vorausberechneten, d.h. wird der Benutzer richtig ausgewiesen wird der Integerwert 1 zurückgegeben.

Kann eine gültige Authentisierung nicht durchgeführt werden, weil entweder die **sessionid** oder der berechnete Wert fehlen oder falsch sind, erfolgt eine entsprechende Fehlermeldung.

Definition: **expected** := md5(concat(**sessionid**,**secret**))

Parameter	Value	Type
Methode	<i>authenticate</i>	
Werte	<i>sessionid</i>	<i>string</i>
	<i>expected</i>	<i>string</i>
Response success	'1'	<i>int</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 2: authenticate(string,string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>authenticate</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>expected</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><int>1</int></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

```

        <member>
            <name>faultString</name>
            <value><string>errorstring</string></value>
        </member>
    </struct>
</value>
</fault>
</methodResponse>

```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

1.2.3. Schließen des Verbindungskanals

Bedingt durch die Gültigkeit der **sessionid** von einer Stunde ist es notwendig, jede nicht mehr benötigte Verbindung zu schließen. Dies entspricht auch gutem Programmierstil.

Dazu wird die Methode *close* unter der URI */valueserver/index.php* auf dem Server mit der URL *service.gutscheinmarketing.de* mit der **sessionid** als *string* aufgerufen.

Kann die Verbindung beendet werden wird der Integerwert 0 zurückgegeben.

Parameter	Value	Type
Methode	<i>close</i>	
Werte	<i>sessionid</i>	<i>string</i>
Response success	'0'	<i>int</i>
error		
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 3: close(string)

Call:

```

<?xml version="1.0"?>
<methodCall>
    <methodName>close</methodName>
    <params>
        <param>
            <value><string>sessionid</string></value>
        </param>
    </params>
</methodCall>

```

Response success:

```

<?xml version="1.0"?>
<methodResponse>
    <params>
        <param>
            <value><int>0</int></value>
        </param>
    </params>
</methodResponse>

```


2. Methoden

Die XMLRPC-Schnittstelle von www.gutscheinmarketing.de stellt grundsätzlich Methoden zur Verfügung, für die eine Benutzer-Authentizierung (entsprechend 1 Verbindungsaufbau, S. 7) notwendig ist.

Die Ausnahmen werden in ##### beschrieben.

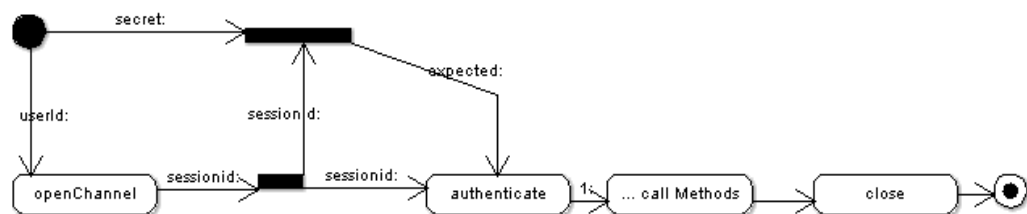
2.1. Allgemeines Vorgehen

Zunächst wird ein geeignetes XMLRPC-Client-Objekt erzeugt. Das Vorgehen dazu ist abhängig von der verwendeten Programmiersprache. Mögliche Lösungen finden sich unter <http://www.xmlrpc.com/directory/1568/implementations>.

Eine speziell optimierte PHP-Klasse für den Zugriff stellt www.gutscheinmarketing.de unter GPL zur kostenlosen und freien Verfügung. Weitere Angaben dazu und eine Beschreibung finden sich unter #####.

2.1.1. Schemata

Das Schaubild verdeutlicht die Vorgehensweise.



2.1.2. Verbindungsaufbau

Mit dem erzeugtem XMLRPC-Client wird eine Verbindung zum XMLRPC-Service von www.gutscheinmarketing.de mit der URL `service.gutscheinmarketing.de` und der URI `/valueserver/index.php` aufgebaut. Die Beschreibung dazu findet sich unter 1.2.1 Verbindungskanal öffnen, S. 8.

Als Ergebnis eines erfolgreichen Verbindungsaufbaues wird eine Sessionkennung (**sessionid**) zurückgegeben. Diese Sessionkennung ist Voraussetzung für jede weitere Kommunikation.

No sessionid no communication!

2.1.3. Authentisieren

Nach erfolgreichem Verbindungsaufbau muss der Benutzer authentisiert werden. Dazu wird ein beidseitiges Geheimnis mit einem Zufallswert berechnet und zusammen mit der Sessionkennung an den XMLRPC-Service gesendet. Die Beschreibung dazu findet sich unter 1.2.2 Benutzer ausweisen, S. 9.

Eine erfolgreiche Anmeldung ist Voraussetzung um die weiteren Methoden aufrufen zu können.

2.1.4. Methodenaufrufe

Ist die Anmeldung erfolgreich durchgeführt, können Methoden in beliebiger Anzahl und Reihenfolge aufgerufen werden. Diese Methoden liefern entweder ein gültiges Ergebnis oder eine Fehlermeldung.

2.Methoden

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.1.5. Verbindung schließen

Wird eine Verbindung nicht mehr benötigt, sollte diese **immer** geschlossen werden. Die Beschreibung dazu findet sich unter 1.2.3 Schließen des Verbindungskanals, S. 11.

2.2. Methoden nur mit Authentisierung

Im folgenden werden Methoden beschrieben, die nur nach einer erfolgreichen Authentisierung aufgerufen werden können.

2.2.1. isValidCoupon

Die Methode liefert den Booleschen Wert von true, wenn der zusammen jeweils als *string* mit der Sessionkennung (**sessionid**) übermittelte Gutscheincode **couponid** gültig ist. Ein Gutscheincode wird als gültig ausgewiesen, wenn

1. der Gutschein aktiviert wurde, d.h. persönliche Angaben wurden ergänzt oder bestätigt und der Gutschein wurde an den Empfänger per Email versendet;
2. der aktivierte Gutschein noch nicht anderweitig eingelöst wurde.
3. das für den Gutschein definierte Verfallsdatum noch nicht überschritten ist.

Ansonsten erfolgt eine Fehlermeldung.

Parameter	Value	Type
Methode	<i>isValidCoupon</i>	
Werte	<i>sessionid</i>	<i>string</i>
	<i>couponid</i>	<i>string</i>
Response success	<i>True</i>	<i>boolean</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 4: isValidCoupon(string,string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>couponid</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

```

    </param>
  </params>
</methodCall>

```

Response success:

```

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><boolean>1</boolean></value>
    </param>
  </params>
</methodResponse>

```

Response error:

```

<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>

```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.2.2. getCouponValues

Die Methode liefert als XMLRPC-Struktur die Werte für einen gültigen Gutschein mit der Gutscheinnummer **couponid** zurück. Dazu muss jeweils die Sessionkennung **sessionid** und die Gutscheinnummer **couponid** als string an die Methode gesendet werden.

Die Struktur der Antwort ist bei vorhandenem und gültigen Gutscheincode abhängig vom Gutschein.

Ist ein Gutscheincode nicht gültig wird eine Fehlermeldung zurückgegeben.

2.Methoden

Parameter	Value	Type
Methode	<i>isValidCoupon</i>	
Werte	<i>sessionId</i>	<i>string</i>
	<i>couponid</i>	<i>string</i>
Response success	Siehe nächste Kapitel	
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 5: getCouponValues(string,string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionId</string></value>
            <value><string>couponid</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.2.2.1. *getCouponValues bei geldwertem Vorteil*

Diese Antwort wird gegeben, wenn der übertragene Gutscheincode einem Geldwertgutschein zugeordnet ist.

2.2.2.1.getCouponValues bei geldwertem Vorteil

Key	Value	Type
type	money	<i>string</i>
price	[cdata]	<i>double</i>
currency	[cdata]	<i>string</i>
invoicetext	[cdata]	<i>string</i>
minimumorder	[cdata]	<i>double</i>
validuntil	[cdata]	<i>dateTime.iso8601</i>

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>type</name>
          <value><string>money</string></value>
        </member>
        <member>
          <name>price</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>currency</name>
          <value><string>[CDATA]</string></value>
        </member>
        <member>
          <name>invoicetext</name>
          <value><string>[CDATA]</string></value>
        </member>
        <member>
          <name>minimumorder</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>validuntil</name>
          <value><dateTime.iso8601>[CDATA]
</dateTime.iso8601></value>
        </member>
      </struct></value>
    </param>
  </params>
</methodResponse>
```

2.2.2.2. *getCouponValues bei Warenwert*

Diese Antwort wird gegeben, wenn der übertragene Gutscheincode einem Warenwertgutschein zugeordnet ist.

Key	Value	Type
type	goods	<i>string</i>
quantity	[CDATA]	<i>double</i>
article	[CDATA]	<i>string</i>
invoicetext	[CDATA]	<i>string</i>
minimumorder	[CDATA]	<i>double</i>
validuntil	[CDATA]	<i>dateTime.iso8601</i>

Response success:

```

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>type</name>
          <value><string>goods</string></value>
        </member>
        <member>
          <name>quantity</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>article</name>
          <value><string>[CDATA]</string></value>
        </member>
        <member>
          <name>invoicetext</name>
          <value><string>[CDATA]</string></value>
        </member>
        <member>
          <name>minimumorder</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>validuntil</name>
          <value><dateTime.iso8601>[CDATA]
</dateTime.iso8601></value>
        </member>
      </struct></value>
    </param>
  </params>
</methodResponse>

```

2.2.2.3. getCouponValues bei Rabattgutschein

Diese Antwort wird gegeben, wenn der übertragene Gutscheincode einem Rabattgutschein zugeordnet ist.

Key	Value	Type
type	percent	string
percent	[CDATA]	double
invoicetext	[CDATA]	string
minimumorder	[CDATA]	double
validuntil	[CDATA]	dateTime.iso8601

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><struct>
        <member>
          <name>type</name>
          <value><string>percent</string></value>
        </member>
        <member>
          <name>percent</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>invoicetext</name>
          <value><string>[CDATA]</string></value>
        </member>
        <member>
          <name>minimumorder</name>
          <value><double>[CDATA]</double></value>
        </member>
        <member>
          <name>validuntil</name>
          <value><dateTime.iso8601>[CDATA]
</dateTime.iso8601></value>
        </member>
      </struct></value>
    </param>
  </params>
</methodResponse>
```

2.2.3. lockCoupon

Diese Methode sperrt einen Gutschein, wenn die Einlösung erfolgreich war, z.B. wenn ein Bestellvorgang durch Zahlung abgeschlossen werden soll. Dazu wird an die Methode die Sessionkennung **sessionid** und der gültige Gutscheincode **couponid** jeweils als *string* gesendet.

Kann der Gutschein erfolgreich gesperrt werden wird eine Kennung zurückgegeben. Diese Kennung kann dazu verwendet werden, um den Gutschein gegebenenfalls wieder zu entsperren, falls der Kunde z.B. die Bestellung storniert oder den Zahlvorgang abbricht (!).

Optional kann ein eigener Kommentar (z.B. eine OrderId oder eine Kundennummer) als *string* der Methode mit übermittelt werden. Dieser Wert ist momentan ohne besondere Bedeutung, könnte aber in Zukunft z.B. für eine Suchfunktionalität herangezogen werden.

Kann ein Gutschein nicht gesperrt werden, so liegt dies daran, dass parallel der Gutschein durch einen anderen Vorgang gesperrt wurde. In diesem Fall wird eine Fehlermeldung zurückgegeben. **Etwaige Rechnerkorrekturen sind geeignet vorzunehmen.**

Parameter	Value	Type
Methode	<i>lockCoupon</i>	
Werte	<i>sessionid</i>	<i>string</i>
	<i>couponid</i>	<i>string</i>
	<i>comment (optional)</i>	<i>string</i>
Response success	<i>trackid</i>	<i>string</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errormsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 6: lockCoupon(string,string) oder lockCoupon(string,string,string)

Call ohne Kommentar:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>couponid</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

Call mit Kommentar:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>couponid</string></value>
            <value><string>comment</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>[CDATA]</string></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.2.4. unlockCoupon

Diese Methode gibt einen gesperrten Gutschein wieder frei. Dazu muss der Methode jeweils als *string* die Sessionkennung **sessionid**, der Gutscheincode **couponid** und der Transaktionscode der Sperrung **trackid** übergeben werden.

Kann der Gutschein entsperrt werden und damit für eine erneute Einlösung freigegeben werden, wird ein Transaktionscode für diesen Vorgang als *string* zurückgegeben.

Optional kann ein eigener Kommentar (z.B. eine OrderId oder eine Kundennummer) als *string* der Methode mit übermittelt werden. Dieser Wert ist momentan ohne besondere Bedeutung, könnte aber in Zukunft z.B. für eine Suchfunktionalität herangezogen werden.

Kann ein Gutschein nicht freigegeben werden, wird eine Fehlermeldung zurückgegeben. Dafür kann es unterschiedliche Ursachen geben:

1. die Transaktionskennung fehlt oder ist falsch;
2. der Gutscheincode entspricht nicht dem bei der Sperrung oder fehlt. Aus Sicherheitsgründen wird auf eine gesonderte Meldung dafür verzichtet.

Parameter	Value	Type
Methode	<i>unlockCoupon</i>	
Werte	<i>sessionid</i>	<i>string</i>
	<i>couponid</i>	<i>string</i>
	<i>trackid</i>	<i>string</i>
	<i>comment (optional)</i>	<i>string</i>
Response success	<i>trackid</i>	<i>string</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 7: unlockCoupon(string,string,string) oder lockCoupon(string,string,string,string)

Call ohne Kommentar:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>couponid</string></value>
            <value><string>trackid</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

Call mit Kommentar:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>isValidCoupon</methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><string>sessionid</string></value>
            <value><string>couponid</string></value>
            <value><string>trackid</string></value>
            <value><string>comment</string></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodCall>
```

2.Methoden

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>[CDATA]</string></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.3. Methoden ohne Authentisierung

Diese Methodenaufrufe sind Gutscheinunabhängig und liefern Informationen zu den Methoden oder lokalisierte Fehlermeldungen.

2.3.1. system.listMethods

Diese Methode ohne weitere Parameter aufgerufen liefert eine Liste aller vom XMLRPC-Service angebotenen Methoden.

Parameter	Value	Type
Methode	<i>system.listMethods</i>	
Werte		
Response success	alle verfügbaren Methoden	<i>string[]</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 8: system.listMethods()

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>system.listMethods</methodName>
  <params>
    </params>
</methodCall>
```

Response:

```

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array><data>
          <value><string>openChannel</string></value>
          <value><string>authenticate</string></value>
          <value><string>close</string></value>
          :
          :
          <value><string>system.listMethods</string></value>
          <value><string>system.methodHelp</string></value>
          <value><string>system.methodSignature</string></va-
lue>
        </data></array>
      </value>
    </param>
  </params>
</methodResponse>

```

2.3.2. system.methodHelp

Diese Methode liefert den zu einer mittels *string* übermittelten Methode **methode** vorhandenen Erklärungstext **helptext** in englisch.

Wird die Methode nicht gefunden, wird eine Fehlermeldung zurückgegeben.

Parameter	Value	Type
Methode	<i>system.methodHelp</i>	
Werte	<i>methode</i>	<i>string</i>
Response success	<i>helptext</i>	<i>string</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 9: system.methodHelp(string)

Call:

```

<?xml version="1.0"?>
<methodCall>
  <methodName>system.methodHelp</methodName>
  <params>
    <param>
      <value><string>methode</string></value>
    </param>
  </params>
</methodCall>

```

2.Methoden

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>[CDATA]</string></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30. Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.3.3. system.methodSignature

Diese Methode liefert die zu einer mittels *string* übermittelten Methode **methode** vorhandenen Signaturen als Array.

Wird die Methode nicht gefunden, wird eine Fehlermeldung zurückgegeben.

Parameter	Value	Type
Methode	<i>system.methodSignature</i>	
Werte	<i>methode</i>	<i>string</i>
Response success	<i>signatures</i>	<i>string[]</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 10: system.methodSignature(string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>system.methodSignature</methodName>
  <params>
    <param>
      <value><string>methode</string></value>
    </param>
  </params>
</methodCall>
```

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><array>
        <data>
          <value><array>
            <data>
              <value><string>[CDATA]</string></value>
              .
              .
            </data>
          </array></value>
          [<value><array> ... </array></value>]
          .
          .
          [<value><array> ... </array></value>]
        </data>
      </array></value>
    </param>
  </params>
</methodResponse>
```

Response error:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>errorcode</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>errorstring</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Mögliche Fehlercodes und -meldungen finden sich unter 3. Fehlercodes, S. 30
Um eine lokalisierte Fehlermeldung zu erhalten wird die Methode *getLocalMessage* (s. S. 28) aufgerufen.

2.3.4. getLocalMessage

Diese Methode liefert den zu einer Meldung mit der Kennung **id** vom Typ **type** entsprechend der Landeskenung **locale** vorhandenen Text oder eine Fehlermeldung, wenn eine entsprechende Übersetzung nicht vorhanden ist.

Derzeit werden deutsch (*de_DE*) und englisch (*en_US*) unterstützt. Eine Übersetzung in andere Sprachen, kann bei Anforderung jederzeit erfolgen.

Parameter	Value	Type
Methode	<i>getLocalMessage</i>	
Werte	<i>type</i>	<i>string</i>
	<i>id</i>	<i>int</i>
	<i>locale</i>	<i>string</i>
Response success	<i>signatures</i>	<i>string[]</i>
error	<i>errorcode</i>	<i>int</i>
	<i>errmsg</i>	<i>string</i>
URI	<i>/valueserver/index.php</i>	
URL	<i>service.gutscheinmarketing.de</i>	

Tabelle 11: *getLocalMessage*(string,int,string)

Call:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>getLocalMessage</methodName>
  <params>
    <param>
      <value><string>type</string></value>
      <value><int>id</int></value>
      <value><string>locale</string></value>
    </param>
  </params>
</methodCall>
```

Response success:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>[CDATA]</string></value>
    </param>
  </params>
</methodResponse>
```

2.3.4.1. Übersetzung einer Fehlermeldung

Um eine Fehlermeldung in einer entsprechend anderen Landessprache zu erhalten (soweit Übersetzung vorliegt) wird die Methode *getLocalMessage* mit dem Typ

type := error

und dem entsprechenden Fehlercode **errorcode** und der Landeskenung entsprechend ISO aufgerufen.

3. Fehlercodes

Bereich	Fehlerursache
<199	allgemeine XMLRPC-Fehler
201-799	allgemeine XML-Fehler
801-999	XMLRPC-Service-Fehlermeldung von www.gutscheinmarketing.de
>1001	Fehlermeldung des XMLRPC-Client

3.1. Allgemeine XMLRPC-Fehler

Fehlerwert	Beschreibung
1	Unbekannte Methode
2	Ungültiger Rückgabewert: schalten Sie in den Debug-Modus ("?debug=1") um die Rückgabewerte zu kontrollieren
3	Ungültige Parameter an Methode übergeben
4	Kann nicht prüfen: Methode unbekannt
5	Kein 200 OK vom Server empfangen

3.2. Allgemeine XML-Fehler

t.b.d

3.3. XMLRPC-Service-Fehlermeldung

Fehlerwert	Beschreibung
801	Benutzer unbekannt
802	Keine gültige Anmeldung
803	Keine gültige Verbindung
804	Kein gültiger Gutschein
805	Gutschein bereits eingelöst durch anderen Server
806	Keinen Gutschein zum Freigeben gefunden oder Gutschein bereits freigegeben

3.4. Fehlermeldung des XMLRPC-Client

Fehlerwert	Beschreibung
1001	Kein Benutzer angegeben
1002	Kein Geheimnis angeben
1003	Keine Gutscheinnummer oder Gutscheinnummer ist leer oder NULL
1004	Kein gültiger Gutschein
1005	Keine gültige Session oder Session nicht initialisiert

4. Tabellenverzeichnis

Tabellenverzeichnis

Tabelle 1: openChannel(string)	8
Tabelle 2: authenticate(string,string)	10
Tabelle 3: close(string)	11
Tabelle 4: isValidCoupon(string,string)	14
Tabelle 5: getCouponValues(string,string)	16
Tabelle 6: lockCoupon(string,string) oder lockCoupon(string,string,string)	21
Tabelle 7: unlockCoupon(string,string,string) oder lockCoupon (string,string,string,string)	23
Tabelle 8: system.listMethods()	24
Tabelle 9: system.methodHelp(string)	25
Tabelle 10: system.methodSignature(string)	26
Tabelle 11: getLocalMessage(string,int,string)	28