



common.education.

präsentiert

Webservices / WebApi revisited (2)

Services konsumieren



Präsentator
Klaus-Peter Luttkus
E-Mail: luttkus@eposgmbh.com



EPOS

EPOS GmbH
DV-Consult & Training

einfach. gut. lernen

Weserstr. 81 28757 Bremen

Tel.: 0421 / 65 99 80 Fax: 0421 / 65 99 865

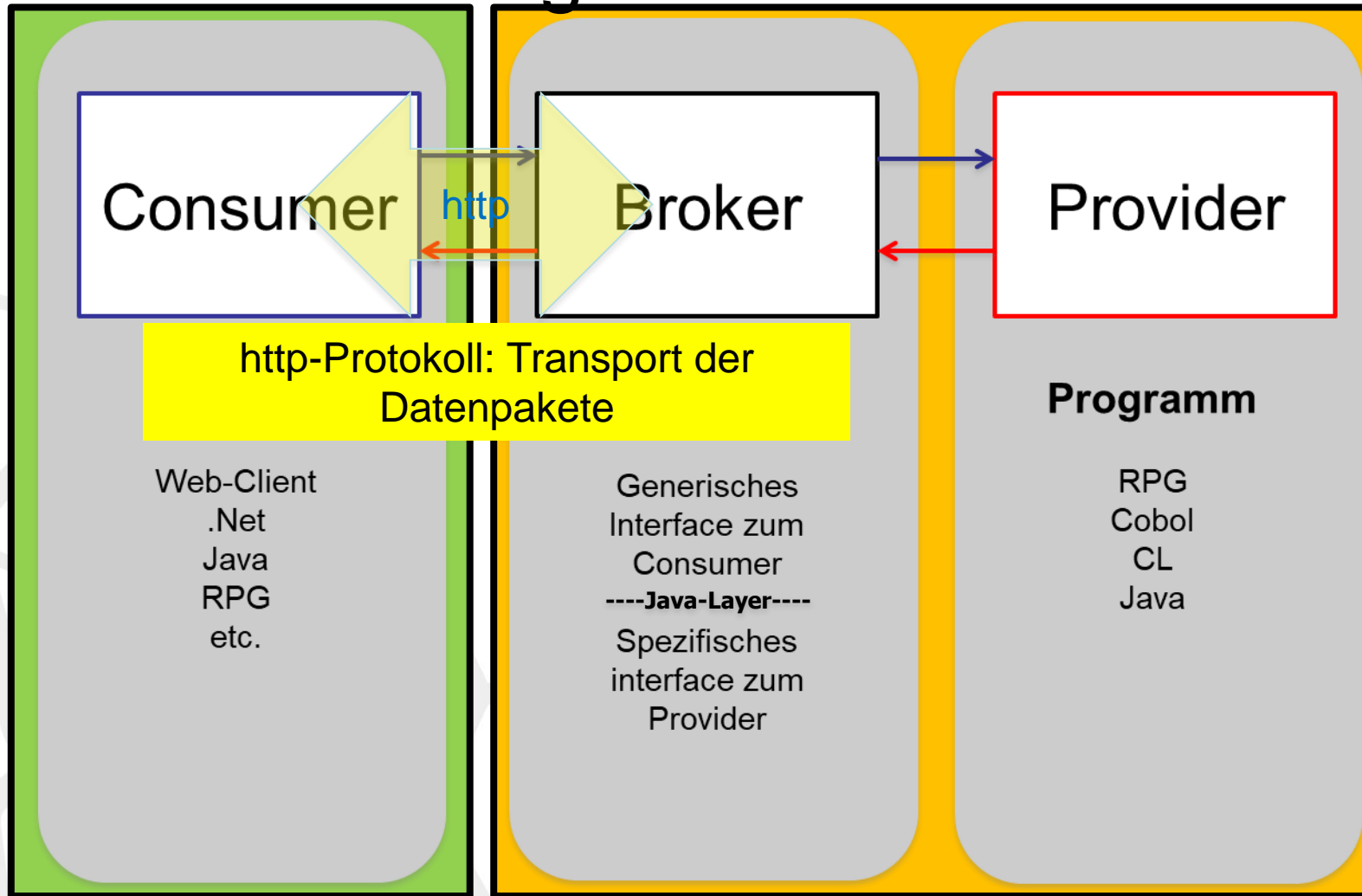
Internet: <http://www.eposgmbh.com>

Agenda



- AXIS API + RPG
 - REST und SOAP
 - JSON- und XML Parsing
- SQL als Webservice Consumer
 - REST
 - CRUD – Create Update Delete + Display

Kurze Einführung in IBM i Web Service Infrastruktur



Integrated Web Services - IWS

AXIS - API

Voraussetzungen

The program temporary fixes (PTFs) that are needed for each of the supported releases of the IBM i operating system in order to use the APIs:

7.3: SI60805, SI60808

7.2: SI60806, SI60809

7.1: SI60807, SI60810

Funktionen

Function	Is used to
<code>axisTransportCreate()</code>	Create a transport object.
<code>axisTransportDestroy()</code>	Destroy a transport object.
<code>axisTransportReset()</code>	Reset the transport object to its initial state.
<code>axisTransportSetProperty()</code>	Set a transport property.
<code>axisTransportGetProperty()</code>	Get a transport property.
<code>axisTransportSend()</code>	Send bytes over transport.
<code>axisTransportFlush()</code>	Flush the transport of any buffered data.
<code>axisTransportReceive()</code>	Receive data from the transport.
<code>axisTransportGetLastErrorCode()</code>	Get transport error code from the last unsuccessful transport operation.
<code>axisTransportGetLastError()</code>	Get transport error string from the last unsuccessful transport operation.

Interfaces / Prototypen

/QIBM/ProdData/OS/WebServices/V1/client/include/Axis.rpgleinc

Workflow

- **Transportobjekt erstellen**
 - `axisTransportCreate()`
 - Erstellt Verbindung
 - Benötigt URI
 - Benötigt HTTP Version
- **Service Request senden**
 - `axisTransportSend()`
 - Buffert Request
 - Benötigt HTTP-Methode: GET, POST, PUT, DELETE
 - `axisTransportSetProperty()`
 - `axisTransportFlush()`
 - Sendet Request-Buffer
- **Service Response empfangen**
 - `axisTransportReceive()`
 - Stellt Response Buffer in Programm-Variable
- **Service Buffer freigeben (zerstören)**
 - `axisTransportDestroy()`

AXIS - API

Das Serviceprogramm HTTP



```
000158 ctl-opt nomain;
000159
000160 ctl-opt alwnull(*usrctl);
000161 ctl-opt option(*nodebugio:*srcstmt:*nounref);
000162
000163 /include /QIBM/ProdData/OS/WebServices/V1/client/include/Axis.rpgleinc
000164
000167 /include QRP191209,HTTP_H
000168 /include QRP191209,HTTP_C
000170
000171
000172
000174 dcl-s gblURI like(tmpURI) inz;
000175
000177 dcl-s gblRequest like(tmpHTTPBuffer) inz;
000178 dcl-s gblResponse like(tmpHTTPBuffer) inz;
```

Prototypen für AXIS


Datenbeschreibungen - Templates

Konstanten

Globale Variablen

Das Serviceprogramm HTTP – createTransport

```
000183 dcl-proc createTransport export;
000184     dcl-pi *n ind;
000185         inEndPoint like(tmpURI) const;
000186         outHandle like(tmpHandle);
000187     end-pi;
000188
000189     dcl-s retCode ind inz(*on);
000190
000191     monitor;
000192         gblURI = inEndPoint;
000193         outHandle = axiscTransportCreate(%trim(gblURI):AXISC_PROTOCOL_HTTP11);
000194         if (outHandle = *NULL);
000195             retCode = *off;
000196         endif;
000197     on-error;
000198         retCode = *off;
000199     endmon;
000200
000201     return retCode;
000202
000203 end-proc;
```

A red arrow points from the right side of the slide towards the function call `axiscTransportCreate(%trim(gblURI):AXISC_PROTOCOL_HTTP11)` on line 000193.

Das Serviceprogramm HTTP – setHTTPMethod

```
000206 dcl-proc setHTTPMethod export;
000207     dcl-pi *n ind;
000208         inHandle like(tmp1Handle) const;
000209         inMethod char(7) const;
000210     end-pi;
000211
000212     dcl-s lclProperties char(100);
000213     dcl-s lclContentType char(100);
000214     dcl-s lclHeader char(100);
000215
000216     dcl-s retCode ind inz(*on);
000217
000218     lclProperties = %trim(inMethod) + NULL_TERM;
000221
000222     monitor;
000223         axiscTransportSetProperty(inHandle: AXISC_PROPERTY_HTTP_METHOD: %addr(lclProperties));
000225     on-error;
000226         retCode = *off;
000227     endmon;
000228
000229     return retCode;
000230
000231 end-proc;
```



Das Serviceprogramm HTTP – sendRequest

```
000234 dcl-proc sendRequest export;
000235   dcl-pi *n ind;
000236       inHandle like(tmplHandle) const;
000237       inRequest like(tmplHTTPBuffer) const;
000238   end-pi;
000239
000240   dcl-s lclStatus int(10) inz;
000241   dcl-s retCode ind inz(*on);
000242
000243   monitor;
000244       gblRequest = inRequest;
000245       lclStatus = axiscTransportSend(inHandle: %ADDR(gblRequest): %len(%trim(gblRequest)) : 0);
000246       if (lclStatus = -1);
000247           retCode = *off;
000248       else;
000249           lclStatus = axiscTransportFlush(inHandle);
000250           if (lclStatus = -1);
000251               retCode = *off;
000252           endif;
000253       endif;
000254   on-error;
000255       retCode = *off;
000256   endmon;
000257
000258   return retCode;
000259
000260 end-proc;
```

Two red arrows originate from the right side of the code block. One arrow points to the 'if (lclStatus = -1);' statement on line 000246, and the other points to the 'else;' statement on line 000248.


Das Serviceprogramm HTTP – getResponse

```
000263 dcl-proc getResponse export;
000264     dcl-pi *n ind;
000265         inHandle like(tmplHandle) const;
000266         outResponse like(tmplHTTPBuffer);
000267     end-pi;
000268
000269     dcl-s lclStatus int(10) inz;
000270     dcl-s retCode ind inz(*on);
000271
000272     monitor;
000274         lclStatus = axiscTransportReceive(inHandle: %ADDR(gblResponse) %SIZE(gblResponse): 0);
000275         if (lclStatus = 0);
000276             retCode = *off;
000277         endif;
000278         outResponse = gblResponse;
000279     on-error;
000280         retCode = *off;
000281     endmon;
000282
000283     return retCode;
000284
000285 end-proc;
```

A diagram consisting of two red arrows. One arrow points from the top-left towards the parameter %ADDR(gblResponse). The other arrow points from the top-right towards the parameter %SIZE(gblResponse): 0). Both parameters are circled in red.


Das Serviceprogramm HTTP – destroyService

```
000421 dcl-proc destroyService export;
000422     dcl-pi *n ind;
000423         inHandle like(tmp1Handle) const;
000424     end-pi;
000425
000426     dcl-s retCode ind inz(*on);
000427
000428
000429     monitor;
000430         axiscTransportDestroy(inHandle);
000431     on-error;
000432         retCode = *off;
000433     endmon;
000434
000435     return retCode;
000436
000437 end-proc;
```

A red arrow points from the right side of the slide towards the 'monitor;' statement on line 000429.

Das Serviceprogramm HTTP – setLogging

```
000287 dcl-proc setLogging export;
000288     dcl-pi *n ind;
000289         inLogStatus ind const;
000290         inLogPath char(255) const;
000291     end-pi;
000292
000293     dcl-s retCode ind inz(*on);
000294
000295     monitor;
000296         if inLogStatus;
000297             axiscAxisStartTrace(%trim(inLogPath): *NULL);
000298         endif;
000299     on-error;
000300         retCode = *off;
000301     endmon;
000302
000303     return retCode;
000304
000305 end-proc;
```

A red arrow points from the right towards the function call `axiscAxisStartTrace(%trim(inLogPath): *NULL);` on line 000297.

Das Serviceprogramm HTTP – setSSLProperties

```
000308 dcl-proc setSSLProperties;
000309     dcl-pi *n ind;
000310         inHandle like(tmpHandle) const;
000311     end-pi;
000312
000313     dcl-s SSLCert          char(100);
000314     dcl-s SSLPassword      char(100);
000315     dcl-s NullStr          char(1) inz(NULL_TERM);
000316     dcl-s None             char(5);
000317
000318     dcl-s retCode ind inz(*on);
000319
000320     monitor;
000321         // Set SSL information - turn off SSLv2 and sslv3.
000322         NONE = 'NONE' + NULL_TERM;
000323         SSLPassword = 'changeit' + NULL_TERM;
000324         SSLCert = '/qopensys/QIBM/ProdData/JavaVM/jdk60/32bit/jre/lib/security/cacerts' + NULL_TERM;
000325         axiscTransportSetProperty(inHandle: AXISC_PROPERTY_HTTP_SSL:
000326                                     %addr(SSLCert):
000327                                     %addr(SSLPassword):
000328                                     %addr(NULLSTR):
000329                                     %addr(NONE):
000330                                     %addr(NONE));
000331
000332     on-error;
000333         retCode = *off;
000334     endmon;
000335
000336     return retCode;
000337 end-proc;
```


Code ist dem Download beigefügt



SQL-Service Consumer

Zur Erinnerung!

Methoden

Folgende Methoden des HTTP-Protokolls unterstützt IWS:

- Um eine Ressource zu erstellen → POST.
- Um eine Ressource zu holen → GET.
- Um eine Ressource zu aktualisieren → PUT.
- Um eine Ressource zu löschen DELETE.

Beispiele

- GET /kunde/4711 → hole Kunde 4711 (URL)
- GET /kunde → hole alle Kunden (URL)
- POST /kunde/4712 → erstelle einen neuen Kunden 4712 (Payload) + (URL)
- PUT /kunde/4712 → aktualisiere Kunde 4712 (Payload)
- DELETE /kunde/4712 → lösche Kunde 4712 (URL)

Die SQL-Funktion: SYSTOOLS.HTTPGETLOB

```
>>--+-HTTPGETBLOB--+-(url,--+------+-)-----><
'-HTTPGETCLOB-'      '-httpHeader-'
```

url

Specifies the URL of the resource being accessed. This argument is defined as a VARCHAR(2048) value.

httpHeader

Specifies an optional header XML document. This argument is defined as a CLOB(10K) value.

Weitere Funktionen:

- HTTPPUTCLOB
- HTTPPOSTCLOB
- HTTPDELETCLOB

Code ist dem Download beigefügt



Danke für Ihre Aufmerksamkeit!

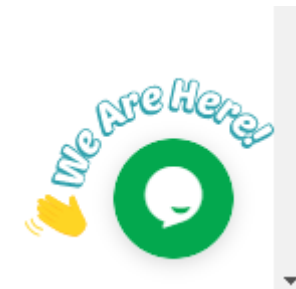


Häufig kommen die Fragen erst am Arbeitsplatz!
Schreiben Sie mir! Ich helfe Ihnen gern weiter!

luttkus@eposgmbh.com



Oder kommen Sie auf unseren
Stand und chatten Sie mit mir!

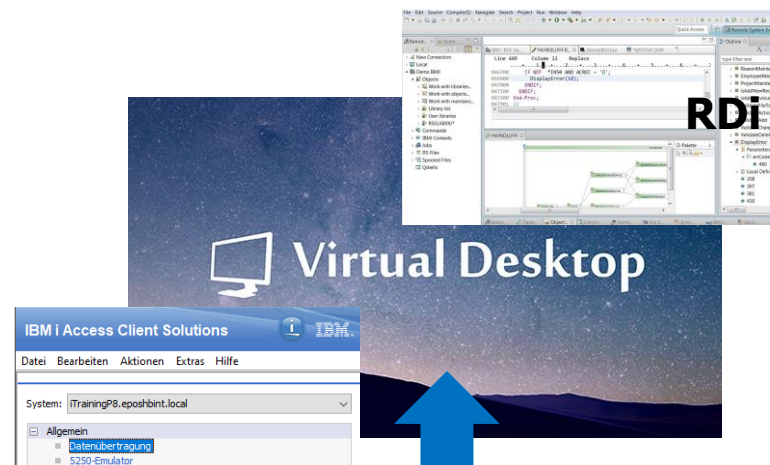


Online - Schulungen



Bild und Ton

Desktopzugriff



einfach.gut.lernen

Schulungen für IBM Power Server (IBM i, iSeries, AS/400, LINUX)

Seit 25 Jahren entwickeln wir bei der EPOS mit hoher Fachkompetenz praxisnahe, offene Seminare und Workshops für IBM Power Systeme (AS/400, iSeries). Unser Wissen, gepaart mit langjähriger praktischer Erfahrung, ist die didaktische Grundlage für unsere Trainings und Seminare. An oberster Stelle steht hierbei aber stets der passgenaue Lösungsansatz, denn wir wissen, dass IT-Mitarbeiter nicht viel Zeit haben und häufig Seminare zur spezifischen Problembewältigung suchen. Aus diesem Grund modularisieren wir unsere Seminare so, dass man leicht in ein Thema einsteigen kann. Das spart Zeit und verhindert Wissensballast.

Schauen Sie sich in Ruhe unser Seminarangebot zu den Bereichen IBM i und LINUX an. Unsere Seminare werden sowohl an Ein- und Unerfahrenen als auch an langjährige Mitarbeiter, die ihr Wissen praxisgerecht erweitern oder die neuesten Technologien kennenlernen wollen.

Und falls dennoch das von Ihnen gewünschte Thema nicht von uns angeboten wird, sprechen Sie uns an. Wir versuchen, alle Kunden zu befriedigen. Wenn Sie spezielle Anforderungen haben. Daher planen wir auch individuelle Seminare ganz nach Ihren Vorgaben – als Präsenz- oder Online-Seminar. Das Seminar kann je nach Wunsch in unserem Haus stattfinden oder bei Ihnen vor Ort.

[Gemeinsam finden wir eine Lösung!](#)

News



20. FEBRUAR 2020
1. Ausbildungszyklus – 2020



20. FEBRUAR 2020
Neue iLearn – Webinare

Terminübersicht

KURS	TITEL	
EPSIL01	ACS Sitzung	€599,00 Buchten

Regelmäßige Schulungs-Info!

E-Mail an: training@eposgmbh.com

Mikro-Schulungen:

- Themenbereiche:
 - ACS
 - Auf dem Client und auf dem Server
 - Rdi
 - WebServices und SQL
 - Single Signon