Gäller fr o m 1 januari 2013

eBREV Webb
– Teknisk handbok

Allmänt

Senaste versionen av den här handboken finns på stralfors.se.

Så skickar du eBrev Webb via WebService

1. Ni levererar ett ziparkiv till Strålfors.
2. Vi kontrollerar filens struktur och varnar för eventuella felaktigheter som inte hänger ihop med utskriftsdatat
3. Filen fördelas mellan de olika utskriftscentralerna där den skrivs ut, kuverteras och lämnas för utdelning.
4. Följesedel och faktura sänds till er.

Testrutiner

Implementeringen av Webbservices måste alltid testas.

Beskrivning av test

Strålfors och testansvarig hos er gör tidsplan och testscenarios.

Testerna sker helst i delmoment.

- inloggning och kommunikation mot testmiljö för Webbservice/PWS
- uppladdning av zip-arkivet och kontroll av ingående filer.
  kontroll av layout, typografi och kvalitet på brev.

Adressfilen för test ska innehålla några fiktiva mottagare.

Beskrivning av eBREV Webbservices

Kommunikationsöversikt

TFS/PWS publicerar WebServices för att underlätta kommunikation mot tjänsten eBREV Webb. PWS möjliggör flera metoder för sändning och hämtning av filer, anpassade till olika tekniska förutsättningar och krav på säkerhet.

Begrepp och termer

Allmänt om WebServices och SOAP

WebServices är en teknik för systemintegration med lösa kopplingar. Web Services utnyttjar bla SOAP (Simple Object Access Protocol) som utgör en programmeringsmodell för distribuerade system. Stödet i utvecklingsverktyget gör användandet av SOAP mycket enkelt.

SOAP innebär att fråga/svar-anrop skickas som textmeddelanden på XML-format. Varje meddelande består av Envelope, Header och Body samt eventuellt Attachments som är ostrukturerade data som kan bifogas ett meddelande. Transportprotokollet är i de flesta fall HTTP(S) vilket är brandväggsvänligt och innehåller en etablerad säkerhetsmodell.

Envelope

Envelope är toppnivån i SOAP-meddelandet och innehåller en Header, en Body och eventuellt en eller flera Attachments.

Header

Headern kan innehålla namngivna parametrar som inte ingår som data i själva meddelandet. Exempel på header-parametrar kan vara information för autenticering, transaktionsstöd eller referenser.

Body

Body innehåller information om vilken metod som ska anropas och argumenten till dessa. Komplexa datatyper kan användas under förutsättning av det finns mekanismer för att serialisera/deserialisera dessa som XML.

Filstruktur på zip-arkiv

För att skicka in data till eBREV Webb via Webbservice-tjänsten skickas ett komprimerat filarkiv i formatet zip. Ziparkivet ska innehålla tre alt. två stycken filer:

- brevfil (PDF eller Word doc)
- konfigurationsfil (cfg) med eller utan adresslista.
- adresslista (txt) om adresser ej återfinns i konfigurationsfilen.

Zip filen måste vara komprimerad på ett sådant sätt att den går att öppna med WinZip. Den får inte vara skyddad med lösenord eller liknande.

Brevfil

Brevfilen ska vara i Microsoft Word eller PDF-format. För insändare, av framförallt PDF, finns ett antal begränsningar och regler. Dessa återfinns på posten.se under hjälpavsnittet för eBREV Webb. Word dokument får endast innehålla bilder (gif, tiff, jpg,bmp) word grafik och text.

Adresslista

Listan på mottagare kan levereras på två olika sätt:

dressfilen ska vara en tabbavgränsad textfil namngiven med ändelsen ”.txt”, t ex

- Som en del av cfg filen (att föredra, se nästa kapitel)
- Som en separat tabbavgränsad textfil.

Oavsett metod tillämpas reglerna nedan:

- Adressfilen ska vara en tabbavgränsad textfil namngiven med ändelsen ".txt", ex. "adresser.txt".
- Filen ska vara sparat i ANSI/ISO-8550-1 format. UTF-8 godkänns inte.
- Bokstaven *Â* får ej användas i adresslistor.
- Endast en mottagare fylls i per rad
- Max 41 tecken i varje fält
- Max 5 000 adressater

Följande tabbavgränsade fält ska finnas med:

Name Address2 Address3 address4 Zip city country
Namn Adressrad1 Adressrad2 Adressrad3 Postnr Postort Landskod

Fältnamn	**Max antal tecken**	**Förklaring**
Namn (mottagare)	max 41 tecken	Obligatoriskt värde
Adressrad1	max 41 tecken	
Adressrad2	max 41 tecken	
Adressrad3	max 41 tecken	
Postnr	max 12 tecken	Obligatoriskt värde
Postort	max 41 tecken	Obligatoriskt värde
Landskod	max 2 tecken	Obligatoriskt värde, SE för Sverige
Vid inbetalningskort kompletteras med:		
Pg/Bg: PG = postgiro, BG = bankgiro		Obligatoriskt värde
Gironumber	beroende på typ, max 20 tecken	Obligatoriskt värde
Betalningsmottagare	max 30 tecken	Obligatoriskt värde
Belopp i kronor, ören		Obligatoriskt värde
Ocr nummer		
text1-text6	max 30 tecken	fritextfält som visas i meddelandefältet

Kontakta eBREV Webb teknisk support för adressfilsmallar.

Utlandsförsändelser

För att skicka brev till utländska mottagare krävs att en korrekt landskod anges.
Landskoden skall följa ISO 3166 standard.
Zip och City är obligatoriska värden oavsett vilket land som försändelsen skall till, adressfälten 1-3 är frivilliga och kan innehålla max 41 tecken.
Debitering sker enligt gällande portotabell.

Konfigurationsfil eBREV-WebbD.cfg

Konfigurationsfilen är unik för eBREV webbservices. XML filen innehåller data som avsändare, mottagare, uppdragstyp och liknande. Filen är en XML-struktur och teckenkoden ska vara UTF-8.

Definitioner på XML taggar

<SenderData>

Tag	**Beskrivning**
<UserID>	Värdet i fältet AnvändarID, ska vara det som valts vid registrering.
<BPN>	Värdet som Ni tilldelats, enligt välkomstbrevet. Det är ett numeriskt värde om tio positioner inkl. två inledande nollor (ex. 0022233311).
<Email>	Kontaktadress som används vid eventuella störningar.

<PriceInfo> alt <ProductOptions>

Tag	**Beskrivning**
<ColorType>	'1' för färg ”0” för svartvit utskrift
<DeliveryType>	’A' för 1:a klassbrev; 'B' för ekonomibrev
<DocumentId>	1 = Med försättsblad (dvs mottagare och avsändare skrivs på ett eget ark) 2 = Giro (endast B-Kort och endast via Posten.se) 3 = Giro (information ligger i bifogad adressfil) 4 = Utan försättsblad (mottagare och avsändare skrivs på dokumentets förstasida)

<SenderAddress>

Avsändaradressen i breven dit obeställbar post returneras. Max 41 tecken per adressrad.

Tag	**Beskrivning**
<addressLine1>	Avsändarens namn
<addressLine2>	Adressrad 2 (gata)
<addressLine3>	Adressrad 3
<addressLine4>	Adressrad 4
<Zipcode>	Max 8 tecken
<Country>	Endast SE som avsändare

<Attachments>

Det finns tre stycken attachment typer; PDF, DOC och Adresslista.
PDF och DOC pekar ut brevfilen och Adresslista pekar ut en tabbseparerad adressfil (då sådan används). Om du väljer att ange mottagarna i cfg-filen behövs inte denna sektion.

Tag	**Beskrivning**
<Type>	Filtyp. DOC, PDF alt Adresslista
<Name>	Filens namn
<Email>	Kontakt adress som används vid eventuella störningar.

Ex.

```
<Attachments>
<Type>PDF</Type>
<Name>pdf.pdf</Name>
</Attachments>
<Attachments>
<Type>Adresslista</Type>
<Name>listan.txt</Name>
</Attachments>
```

Recipients
Denna array innehåller mottagaradresser. När denna array återfinns i cfg-filen behövs ingen fristående adressfil. Fältens innehåll och begränsningar beskrivs i kapitlet *Adresslista.*

```
<Recipients>
<Letter>
<File>DOCUMENT.PDF</File>
<ZipCode>806 36</ZipCode>
<City>GÄVLE</City>
<Country>SE</Country>
<RecipientName>Gunilla Åberg</RecipientName>
<AddressLine1/>
<AddressLine2>Ängsullsvägen 6</AddressLine2>
<AddressLine3/>
</Letter>
</Recipients>
```

Med giro info skall även nedanstående ingå I Recipient taggen:

```
<GiroInfo>
<GiroType>PG</GiroType>
<GiroNumber>12345</GiroNumber>
<Amount>100,20</Amount>
<SenderName>robert</SenderName>
<OCR>848484848488</OCR>
<RefText1>Detta är en fritext</RefText1>
<RefText2/>
<RefText3/>
<RefText4/>
<RefText5/>
<RefText6/>
</GiroInfo>
```

API

Autentisering
För HTTP Basic anges användare och lösenord som en base64-kodad HTTP-header enligt RFC-1945.

Exceptions och felkoder
Metoden sendWithAddressing och send kastar exceptions när försändelsen inte är godkänd eller vid driftsstörningar, dessa fel måste hanteras av klientapplikationen.
En beskrivning av felkoderna återfinns nedan under rubriken ”Exceptions och felkoder”

Returvärden
Det returvärde (märke) som returneras från ”send” metoderna skall alltid sparas av kund. Det bör kopplas ihop med det data som du skickat in till Posten i en databas eller liknande.
Detta märke används av Posten servicedesk för att referera till er försändelse.
Detta märke skall även användas vid reklamation.

Metoder
sendWithAddressing

```
String sendWithAddressing( byte[] data,
String tfsMsgType,
String tfsSender,
String tfsReceiver )
```

Returvärde:
Sträng innehållande statuskod och transaktionsreferens (ex. 200 PWS.xxxx.xxxx.xxxx).
Beskrivning: Sänd data som ett argument i form av en minnesbuffert samt adresseringsparametrar. tfsMsgType samt tfsReceiver är i normala fall ”ebrevwebb”, tfsSender är kundunikt id.

Ping

```
String ping()
```

Beskrivning
Kontrollerar webbservicens status. Endast för test.
Vad noga med att inte anropa denna metod mellan varje anrop till sendxxx då detta förlänger processtiden för dina inskick.

Returvärde
En sträng. ”pong”.

Exceptions och felkoder

När PWS inte accepterar en försändelse kastar den ett s.k. ”exception”. Exempel av utskrift av ett felmeddelande från ett exception; Errorcode: (300) Invalid zip. Too few files, found: 2, expected: 3

I felmeddelandet återfinns en felkod som beskrivs nedan.

Kod	Beskrivning
200	Ok
300	Ej godkänd zip fil
301	Ej godkänd cfg fil
302	Konfigurationsfil saknas
303	Felaktigt dokument
304	Dokument (doc/pdf) saknas
305	Adresslista saknas
306	Ogiltig adresslista
307	Konverterings fel kontakta support
308	Kommunikationsfel ,kontakta support
309	Felaktig teckenuppsättning (character set). Adresslistan skall vara ANSI (ISO-8859-1).Config skall vara UTF-8
310	XML Parsning misslyckades, felaktig konfig fil
312	Dokumentnamned i CFG stämmer ej överens med filens namn
313	Zip filen är för stor (max 3MB)
314	Dokumentet är för stort (max 3MB)
315	SOAP header saknas (endast om du använder ”send”. Byt till ”sendToTfs” metoden istället.
316	Konverterings fel, kontakta support
317	För många mottagare i adressfilen (max är 5.000)
318	Ogiltigt PDF. Vanligast förekommande när dokumentet skapats av Ghostscript i kombination med TTF fonter. Kontrollera vilket dokument som skapat PDF’en samt att fonterna (typsnitten) som används är giltiga. fonter som har namn av typen xxxx+TTE är ej giltiga.
319	Felaktigt antal sidor i dokumentet. Min 1, max 6 (12 för B-post/Ekonomibrev).

Kundstöd

Se kontaktuppgifter på posten.se

FAQ

När kan man skicka in sina brev?
Tjänsten är öppen dygnet runt med undantag för service.
Vart vänder man sig vid kommunikationsproblem?
Se kontaktuppgifter på posten.se
Vilka SOAP-standarder stöds?
PWS stödjer de inofficiella W3C-standarderna:
SOAP 1.1, http://www.w3.org/TR/soap
WSDL 1.1, http://www.w3.org/TR/wsdl
Vilka implementationer av SOAP är kompatibla med PWS?

PWS bygger på Apache Axis 1.1. kompabilitet med andra SOAP-implementationer framgår av diverse interoperabilitetstester, t ex http://www.whitemesa.com/interop.htm.
I strävan att bibehålla kompabilitet har i möjligaste mån komplexa datatyper undvikits i parametrar och returvärden.

Exempel

Konfigurationsfil med separat adressfil

OBS! Teckenkodning i UTF-8
Filnamn=eBREV-WebbD.cfg
Exempel med fristående adresslista:

```
<?xml version="1.0" encoding="UTF-8"?>
<Config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\xml\config.xsd">
<SenderData>
<UserId>xxxx</UserId>
<CustomerId>yyyy</CustomerId>
<BPN>1234567</BPN>
<DocumentId>4</DocumentId>
</SenderData>
<PriceInfo>
<ColorType>0</ColorType>
<DeliveryType>B</DeliveryType>
</PriceInfo>
<senderAddress>
<addressLine1>Företaget AB</addressLine1>
<addressLine2>Box 123</addressLine2>
<addressLine3></addressLine3>
<addressLine4></addressLine4>
<zipCode>12345</zipCode>
<city>Stockholm</city>
<country>SE</country>
</senderAddress>
<Attachments>
<Type>DOC</Type>
<Name>doc.doc</Name>
</Attachments>
<Attachments>
<Type>Adresslista</Type>
<Name>adresses.txt</Name>
</Attachments>
</Config>
```

Konfigurationsfil med adresslista och giro information

Denna typ av config är att föredra framför den där mottagaradresserna levereras i en separat textfil.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Config>
<SenderData>
<BPN>0020000000</BPN>
<UserId>myuser</UserId>
<Email>kale@mycompany.com</Email>
</SenderData>
<ProductOptions>
<ColorType>0</ColorType>
<DeliveryType>B</DeliveryType>
<DocumentId>3</DocumentId>
</ProductOptions>
<SenderAddress>
<AddressLine1></AddressLine1>
<AddressLine2></AddressLine2>
<AddressLine3></AddressLine3>
<AddressLine4></AddressLine4>
<ZipCode>12345</ZipCode>
<City>Mycity</City>
<Country>SE</Country>
</SenderAddress>
<Recipients>
<Letter>
<File>DOCUMENT.PDF</File>
<ZipCode>75435</ZipCode>
<City>Uppsala</City>
<Country>SE</Country>
<RecipientName>Kalle Kanin</RecipientName>
<AddressLine1>Gatan 123</AddressLine1>
<AddressLine2 />
```

```
<AddressLine3 />
<GiroInfo>
<GiroType>PG</GiroType>
<GiroNumber>123-123</GiroNumber>
<Amount>100,01</Amount>
<PaymentReceiver>My Company</PaymentReceiver>
<OCR>848484848488</OCR>
<RefText1>Fritext</RefText1>
<RefText2></RefText2>
<RefText3></RefText3>
<RefText4></RefText4>
<RefText5></RefText5>
<RefText6></RefText6>
</GiroInfo>
</Letter>
</Recipients>
</Config>
```

Kodexempel

Kodexemplen som anropar PWS tjänsten i detta avsnitt är skrivna i Java (nyttjar produkten Axis från Apache; http://ws.apache.org/axis/) samt i C# (MS Visual Studio .NET).
Exemplen måste kompletteras med kundunika parametrar.
Observera att testmiljön och produktionsmiljön använder sig av olika URL'er samt inloggningsinformation.
Observera att dessa exempel inte är kompletta utan skall endast användas som en vägledning.

Sändning med Java (>=1.5)

```
import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
import javax.xml.namespace.QName;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
public class Pws2Sample {
public final String NAMESPACE = "http://pws.posten.se";
public final int BUFSIZE = 8152;
/**
* Endpoint make sure it is correct!
*/
public final String ENDPOINT = "http://localhost:9081/pws/services/PWS";
/**
* Authentication. Change here
*/
public String username = "YOUR_USER";
public String password = "YOUR_PASSWORD";
public byte[] getByteBuffer( InputStream is ) throws IOException {
ByteArrayOutputStream os = new ByteArrayOutputStream();
byte[] buf = new byte[BUFSIZE];
int len;
while( true ) {
len = is.read( buf, 0, buf.length );
if( len < 0 )
break;
os.write( buf, 0, len );
}
return os.toByteArray();
}
public void go(String fileName){
try {
InputStream is = new FileInputStream( fileName );
// Replace sender with your own user
String tfsMsgType = "EBREVWEBB";
String tfsSender = "Company X";
String tfsReceiver = "EBREVWEBB";
Service service = new Service();
```



```
Call call = (Call) service.createCall();
call.setTargetEndpointAddress( new URL( ENDPOINT ) );
call.setOperationName( new QName( NAMESPACE, "sendWithAddressing" ) );
call.setUsername( username );
call.setPassword( password );
String result = (String) call.invoke( new Object[] {getByteBuffer(is), tfsMsgType,
tfsSender, tfsReceiver} );
// The result contains 200 + mark (e.g. "200 SOAP.xxx.xxx")
```

```
System.out.println(result);
}
catch( Exception e ) {
// Message contains error code and description
// Consult your manual for more information.
e.getMessage();
}
}
public static void main( String[] args ) {
Pws2Sample sample = new Pws2Sample();
sample.go(args[0]);
}
}
```

Sändning med C# (MS Visual Studio .NET 3.5 WCF)

Kodfragment. Koden använder sig av en proxy klass (`SendClient`) som genererats av Visual Studio.

```
// Avoids the first (505) error
System.Net.ServicePointManager.Expect100Continue = false;
// Load the zip as a binary
string filename = “c:/mytestfile.zip”
System.IO.FileStream fs = File.OpenRead(filename);
byte[] data = new byte[fs.Length];
fs.Read(data, 0, data.Length);
EndpointAddress ea = new EndpointAddress(endpoint); // <-- Insert the correct URL
// Setup security
BasicHttpBinding basic = new BasicHttpBinding();
basic.Security.Mode = BasicHttpSecurityMode.TransportCredentialOnly;
basic.Security.Transport.ClientCredentialType = HttpClientCredentialType.Basic;
pws2.SendClient client = new pws2.SendClient(basic,ea);
client.ClientCredentials.UserName.UserName = "xxx"; // <-- Change here
client.ClientCredentials.UserName.Password = "yyy"; // <-- Change here
try
{
// The result contains 200 + mark (e.g. "200 PWS.xxx.xxx")
// Save this in a database for later reference!
string result = client.sendWithAddressing(data, "EBREVWEBB", "Your company", "EBREVWEBB");
// <-- Change TfsSnd (Your Company) here
lstStatus.Items.Add("Result:" + result);
}
catch (FaultException fe)
{
// Add proper error handling!
MessageBox.Show(fe.ToString());
}
catch (Exception eee)
{
MessageBox.Show(eee.ToString());}
```

eBREV WebbTeknisk Handbok SF 2013--01

Strålfors Svenska AB
08-508 830 00
www.stralfors.se