



Skicka och hämta filer med automatik till
och från Försäkringskassan

Innehållsförteckning

Revisionshistorik	3
Inledning.....	4
1 Föresättningar.....	4
1.1 Registrera.....	4
1.2 Certifikat.....	4
2 Skicka filer med automatik.....	5
2.1 Adresser	5
3 Syntax exempel för cURL	6
3.1 Syntax för inlämning med cURL	6
3.2 Syntax för hämtning av filer med cURL.....	6
3.3 Adresser för cURL	8
4 Testning	9
4.1 Testmiljö variabler.....	9
5 Felhantering	10
6 Verktyg	11
6.1 Programvara cURL	11
6.2 Programvara Openssl.....	11
6.3 Programvara wget.....	11
7 Certifikathantering.....	12
8 Produkter som stöds via cURL	13
8.1 Publika cURL tjänster i produktions system	13
8.2 Publika cURL tjänster i test system	13
9 Kontaktpersoner.....	14
9.1 Tekniska frågor produktion/test miljöer	14
10 Exempelfiler	15
10.1 Skicka filer.....	16
10.2 Unixmiljö - sendSHSFiles.....	17
10.3 Hämta enstaka filer	18
10.4 Unixmiljö – getOneSHSFile	19
10.5 Hämta flera filer	20
10.6 Unixmiljö – getAllSHSFiles	21

Datum
2017-10-18

Version
F

Datum
2017-10-18Version
F

Revisionshistorik

Revision	Datum	Vem	Varför
A	2009-04-16	Henrik Wallner	Version A granskad och klar.
D	2014-10-08	Daniel Embretsen	Version D granskad och klar.
E	2017-09-15	Johan Lindfors	
F	2017-10-18	Johan Lindfors	Version F granskad och klar.

Inledning

Detta dokument vänder sig till kunder och partners som vill automatisera filöverföringen med Försäkringskassan. Den automatiserade filöverföringen görs med ett verktyg som stödjer protokollen HTTP GET och HTTP POST samt SSL. Genom att till exempel använda en klient från en tredjeparts-leverantör kan filer hämtas och levereras till/från en Unix- eller Windowsserver. Det finns ett flertal produkter på marknaden. Som ett exempel i detta dokument på en möjlig tredjeparts produkt använder vi cURL samt Openssl.

Försäkringskassan tar inget ansvar för cURL eller någon annan av dessa produkter. Vid problem får användaren vända sig till leverantören.

1 Förutsättningar

1.1 Registrera

Kunden behöver registrera kunduppgifter för den tjänst som ska användas.

1.2 Certifikat

För att den automatiska filöverföringen ska fungera krävs ett servercertifikat. Instruktioner för att beställa ett certifikat finns på nedanstående URL.

<https://eid.expisoft.se/>

2 Skicka filer med automatik

2.1 Adresser

För att skicka filer till Försäkringskassan används HTTP PUT eller POST. Formatet för URL är:

`https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>/<filnamn>`

eller

`https://shssat.forsakringskassan.se:8020/rest/auto/<produktidentitet>/`

För testmiljön är adressen

`https://shssat.forsakringskassan.se:8020/rest/auto/<produktidentitet>/`

`<produktidentitet>` är det unika id på produkten som skall skickas in (t.ex. ”d623458e-8b20-4868-afae-98f230049329” vilket motsvarar produkten ”FK.Sjukanmälan.CURL”).

`<filnamn>` är inte obligatoriskt i URL:en utan kan anges som parameter i klientanropet (se vidare exempel med cURLs om klient).

Svaret tillbaka om det har gått bra listar transaktionsidentiteten som meddelandet har fått:

`<transaktionsidentitet>`

3 Syntax exempel för cURL

<my.crt>	Fil som det egna servercertifikatet
<my.key>	Fil som innehåller nyckeln till det egna certifikatet (om den inte finns i <my.crt> filen)
<CA.crt>	Root certifikatet från certifikatutgivaren (CA) som används till att utfärda ert certifikat och nyckel. Används i våra exempel på cURL anrop.

3.1 Syntax för inlämning med cURL

OBS! Möjligheten att använda ett MIME multipart/form-data format för att skicka in flera filer i ett meddelande är borttagen från 2017-11-12.

3.1.1 Syntax för att skicka in en fil i ett meddelande (POST):

```
curl --cert <my.crt> --cacert <CA.crt> --upload-file <fil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>" -H "content-type: text/xml"
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt> --upload-file <fil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>" -H "content-type: text/xml"
```

3.1.2 Syntax för att skicka in en fil i ett meddelande (PUT):

```
curl --cert <my.crt> --cacert <CA.crt> -d @<fil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>"
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt> -d @<fil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>"
```

3.2 Syntax för hämtning av filer med cURL

<my.crt>	Fil som det egna servercertifikatet
<my.key>	Fil som innehåller nyckeln till det egna certifikatet (om den inte finns i <my.crt> filen)
<CA.crt>	Root certifikatet från certifikatutgivaren (CA) som används till att utfärda ert certifikat och nyckel. Används i våra exempel på cURL anrop.

För att kunna hämta en fil eller snarare ett meddelande, som kan innehålla en eller flera filer, måste transaktionsidentiteten ingå i URL:en. Att hämta filer sker oftast i två steg. I det första steget görs en fråga om det finns några filer att hämta för en specifik produkt. Svaret returnerar transaktionsidentiteter på separat rader. Varje transaktionsidentitet motsvarar ett meddelande som kan innehålla

Datum	Version
2017-10-18	F

en eller flera filer. Identiteten för behörighetskontroll på den som hämtar filer tas från certifikatet (organisationsnummer).

För att hämta filer från försäkringskassan så används HTTP GET.

Syntax för att hämta en lista med möjliga transaktioner för en viss produkt:

```
curl --cert <my.crt> --cacert <CA.crt>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>"
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>"
```

Svaret som returneras tillbaka blir då:

```
<transaktionsidentitet 1>  
<transaktionsidentitet 2>  
...  
<transaktionsidentitet n>
```

Önskas också originalfilnamnet med för transaktionerna så är syntaxen följande:

```
curl --cert <my.crt> --cacert <CA.crt>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>?getfilenames=true"
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>?getfilenames=true"
```

I detta fall blir svaret:

```
<transaktionsidentitet 1>/<filnamn>  
<transaktionsidentitet 2>/<filnamn>  
...  
<transaktionsidentitet n>/<filnamn>
```

I nästa steg kan filerna hämtas genom att ange transaktionsidentiteten, från svaret i första frågan, i anropet. Utan att ange ett specifikt filnamn så ser syntaxen ut som följande:

```
curl --cert <my.crt> --cacert <CA.crt> --output <svarsfil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>/<transaktionsidentitet>
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt> --output <svarsfil>  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>/<transaktionsidentitet>
```

<svarsfil> anger namnet på den lokala filen där cURL lagrar informationen som hämtas.



Datum	Version
2017-10-18	F

Om filnamnet används i URL i samband med hämtning från Försäkringskassan så är syntaxen för detta:

```
curl --cert <my.crt> --cacert <CA.crt> --remote-name  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>/<transaktionsidentitet>/<filnamn>
```

eller om certifikat och nyckel ligger i olika filer

```
curl --cert <my.crt> --key <my.key> --cacert <CA.crt> --remote-name  
"https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>/<transaktionsidentitet>/<filnamn>
```

<filnamn> spelar i detta fall ingen roll för SHS utan det styr bara vilket namn som cURL skall ha för lokal lagring av svaret tillsammans med parametern --remote-name.

3.3 Adresser för cURL

Produktion:

https://shs.forsakringskassan.se:7020/rest/auto/<produktidentitet>

Testmiljön

https://shssat.forsakringskassan.se:8020/rest/auto/<produktidentitet>

Datum
2017-10-18Version
F

4 Testning

4.1 Testmiljö variabler

Kunden har möjlighet att testa sin implementation efter samråd med Försäkringskassan. Konfigurerings behövs inom Försäkringskassan för att det hela skall fungera även i testmiljö.

Observera att vissa externa partners testar mot sina unika produktidentiteter för sina respektive flöden.

Adress för test

<https://shssat.forsakringskassan.se:8020/rest/auto/<produktidentitet>>

Produktidentitet för generell cURL test:

74d2caeb-6ecc-f533-4822-5da8113ab479

Certifikat för produktion och testning

FK IT har en säkerhetspolicy som gör att vi för tillfället enbart använder servercertifikat för produktion i samtliga externt uppkopplade miljöer. Det måste inte vara samma certifikat som ska användas i produktion och test men det måste vara samma CA.

Gå in på <https://eid.expisoft.se/> för instruktioner hur du beställer ett Steria/expisoft server certifikat.

Observera att FK IT inte supportar några certifikatfrågor, detta måste er CA göra (Steria/expisoft i detta fall).

Produktnamn

FK.CURL.Test - Denna information behöver aldrig anges i URL anropen.

Transaktionsidentitet

Kan erhållas vid behov från försäkringskassan för att testa uthämtning av fil.

Exempel på URL

<https://shssat.forsakringskassan.se:8020/rest/auto/74d2caeb-6ecc-f533-4822-5da8113ab479>

Datum
2017-10-18Version
F

5 Felhantering

Detta kapitel beskriver vilka felmeddelanden som kommer tillbaka från funktionen i olika fel fall när cURL används som klient.

FEL: SHS-server är avstängd vid inskick av filer

SVAR: "curl: (7) couldn't connect to host" eller ett HTML-dokument med titeln "Error 500-- Internal Server Error"

FEL: SHS-server är avstängd vid uthämtning av filer med transaktionsid

SVAR: "curl: (7) couldn't connect to host" eller ett HTML-dokument med titeln "Error 500-- Internal Server Error"

FEL: Icke existerande produktidentitet anges vid inskick av filer

SVAR: "ERROR: Access denied"

FEL: Icke existerande produktidentitet anges vid uthämtning av filer

SVAR: Tom lista tillbaka om bara produktidentiteten anges och den inte finns.

FEL: Ej behörig produktidentitet anges vid inskick av filer

SVAR: "ERROR: Access denied"

FEL: Ej behörig produktidentitet anges vid uthämtning av filer

SVAR: Tom lista tillbaka om bara produktidentiteten anges (ingen transaktionsidentitet alltså) och behörighet saknas till denna produktidentitet.

FEL: Icke existerande transaktionsidentitet anges vid uthämtning av filer

SVAR: "ERROR: Not valid MIME format"

Datum
2017-10-18Version
F

6 Verktyg

6.1 Programvara cURL

Hemsida: curl.haxx.se

Typ: Freeware

cURL är en freeware programvara som kan användas för en massa olika protokoll (FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE). Programmet stödjer SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies och många andra saker. För komplett information angående programmet cURL så hänvisas till hemsidan och manualen.

Förklaring till de parametrar som används i exempel i detta dokument:

<code>--cert <my.crt></code>	Anger egen servercertifikatfil att använda
<code>--key <my.key></code>	Anger egen fil med privata nyckel
<code>--cacert <CA.crt></code>	Anger certifikatfil för CA att använda
<code>--output <svarsfil></code>	Anger fil som resultatet lagras i
<code>--remote-name</code>	Anger att resultatet lagras med samma namn lokalt som namnet på filen som hämtas
<code>--upload-file <fil></code>	Anger fil som skall skickas till servern

Andra parametrar som kan vara bra:

<code>--verbose</code>	Skriver ut mer detaljerad information om vad som händer i kommunikationen, att användas vid tex felsökning
------------------------	--

6.2 Programvara Openssl

Hemsida: www.openssl.org

Typ: Freeware

OpenSSL projektet har utvecklat en öppen källkodsimplementering av transportprotokoll. Mer information om detta fås via hemsidan till Openssl.

6.3 Programvara wget

Hemsida: www.gnu.org

Typ: Freeware

7 Certifikathantering

Certifikaten levereras från externa Certifikatutfärdare, CA (Steria/Expisoft) i ett format för att installeras i en webbläsare tex. .p12, .pfx. För att få dem att fungera med cURL eller annan programvara för automatik måste certifikaten göras om till ett annat format s.k. PEM-format.

För att göra denna omvandling behövs en tredjeparts produkt t.ex. Openssl. För att bryta isär en .p12-fil till ett certifikat och en privat nyckel ska följande kommando köras:

openssl pkcs12 -in <cert.p12> -clcerts -nodes -out <mycert.crt>

Openssl frågar efter ett lösenord. Man ska då ange det lösenord som man fått från sin CA. Openssl sparar då certifikatet och den privata nyckeln (okrypterad) i filen <mycert.crt>. Det är denna fil, <mycert.crt>, som ska anges till parametern --cert i cURL-anropet. Om nyckeln hamnar i en egen fil, så skall den filen användas med parametern --key i cURL-anropet.

OBSERVERA, det är viktigt att nyckeln i detta fall skyddas på annat sätt, t.ex. genom att sätta snäva rättigheter på filen där nyckeln ligger lagrad.

För andra instruktioner hur man konverterar certifikat mellan olika format så finns det bra instruktioner online, sök efter just ert behov av konvertering.

Datum
2017-10-18Version
F

8 Produkter som stöds via cURL

Samtliga asynkrona externa tjänster som Försäkringskassan erbjuder fungerar med denna teknik. Kontakta Försäkringskassan IT gällande era behov. För att kunna ansluta er till dessa tjänster måste ni alltid kontakta shssupport@forsakringskassan.se.

8.1 Publika cURL tjänster i produktions system

Namn	Identitet
FK.Sjukanmälan.CURL	d623458e-8b20-4868-afae-98f230049329

8.2 Publika cURL tjänster i test system

Namn	Identitet
FK.CURL.Test	74d2caeb-6ecc-f533-4822-5da8113ab479

9 Kontaktpersoner

9.1 Tekniska frågor produktion/test miljöer

SHS support

Mail: shssupport@forsakringskassan.se

10 Exempelfiler

Följande kapitel ger exempel på kommandofiler för att hämta och lämna filer i Unix respektive windowsmiljö. Försäkringskassan tar inget ansvar och ger ingen support för dessa filer utan de är tänkta som en start för att kunna göra egna kommandobaserade program för denna funktion.

Den information som kan behöva anpassas i nedanstående skriptfiler är:

- namn på egna certifikatfilen och på CA-certifikatfilen
- namn på filen för nyckeln till egna certifikatet
- id på produkten som skall skickas/hämtas
- URL för test eller produktionsmiljö



Datum
2017-10-18

Version
F

10.1 Skicka filer

Windowsmiljö – sendSHSFiles.bat

```
:: Fil för att skicka en fil till SHS
:: sendSHSFiles.bat [-P product] file1
:: -P product = Vilket produkt UUID som skall användas
:: file1 = fil som skall med detta meddelande

@echo OFF

ECHO "INITIERAR..."

:: lagrar filnamnen
SET FK_FILENAMES=?filenames=%1

:: Definierar produkten om inget skickas in
:: FK.Sjukanmälan.XML.sat-saturnus
SET FK_ALLTOPPRODUCT=3aac5986-bda6-ff8d-4b6d-351b84166c32

:: läs argumenten
IF "%1"=="-P" SET FK_ALLTOPPRODUCT=%2
IF "%1"=="-P" SET FK_FILENAMES=?filenames=%3

:: test
SET FK_SHSURL=https://shssat.forsakringskassan.se:8020/rest/
:: produktion
SET FK_SHSURL=https://shs.forsakringskassan.se:7020/rest/

SET FK_CLIENTCERTFILE=my.crt
SET FK_CLIENTKEYFILE=my.key
SET FK_CACERTFILE=ca.crt
SET FK_CURLFILE=curl --cert %FK_CLIENTCERTFILE% --key %FK_CLIENTKEYFILE% --cacert %FK_CACERTFILE%
SET FK_CURLCOMMAND=%FK_CURLFILE% %FK_MIMEFILE% "%FK_SHSURL%%FK_ALLTOPPRODUCT%%FK_FILENAMES%"

:: Kör kommandot för att skicka filer
ECHO "SKICKAR INFORMATION ..."
ECHO %FK_CURLCOMMAND%
%FK_CURLCOMMAND%

:: Ta bort definierade variabler
ECHO "RENSAR..."
SET FK_ALLTOPPRODUCT=
SET FK_FILENAMES=
SET FK_SHSURL=
SET FK_CLIENTCERTFILE=
SET FK_CLIENTKEYFILE=
SET FK_CACERTFILE=
SET FK_CURLFILE=
SET FK_CURLCOMMAND=

ECHO "KLAR"
```



Datum
2017-10-18

Version
F

10.2 Unixmiljö - sendSHSFiles

```
#!/usr/bin/ksh

# sendSHSFiles file1 [file2] [...]
# skickar in alla angivna filer till SHS

echo "INITIERAR..."

# sätter produktid om inget annat anges
# motsvarar produkten FK.Sjukanmälan.XML.sat-saturnus
fk_productUUID=3aac5986-bda6-ff8d-4b6d-351b84166c32

echo "LADDAR FILER..."

fk_filenames=""?filenames=$1"
fk_numfiles=1

until [[ $# -eq 1 ]]; do
    ((fk_numfiles+=1))
    fk_filenames="$fk_filenames,$2"
    shift
done

# test
fk_shsurl=https://shssat.forsakringskassan.se:8020/rest/
# produktion
fk_shsurl=https://shs.forsakringskassan.se:7020/rest/

fk_clientcertfile=my.crt
fk_clientkeyfile=my.key
fk_cacertfile=ca.crt

fk_curlfile="/usr/local/bin/curl --cert $fk_clientcertfile --key $fk_clientkeyfile --cacert $fk_cacertfile"
fk_curlurl="$fk_shsurl$fk_productUUID$fk_filenames"

fk_curlcommand="$fk_curlfile $fk_curlurl"

echo "CURL KOMMANDO..."
echo "$fk_curlcommand"
$fk_curlcommand

echo "AVSLUTAR..."
```

Datum
2017-10-18Version
F

10.3 Hämta enstaka filer

Windowsmiljö – getSHSFile.bat

```
:: Fil för att hämta en produkt från en SHS
::
:: getSHSFile.bat [-P product] -T transaction
:: -P product = UUID för produkten som filerna tillhör
:: -T transaction = transaktionsid för filerna/meddelandet
@echo OFF

ECHO "INITIERAR..."

:: Definiera produkten om det inte skickas in något värde
:: FK.Sjukanmälan.XML.sat-saturnus
SET FK_ONEFROMPRODUCT=3aac5986-bda6-ff8d-4b6d-351b84166c32

:: Kolla om parametern -P är satt
IF "-P"=="%1" SET FK_ONEFROMPRODUCT=%2

:: Kolla om parametern -T är satt
IF "-T"=="%1" SET FK_TRANSACTION=%2
IF "-T"=="%3" SET FK_TRANSACTION=%4

:: Inställningar
:: test
SET FK_SHSURL=https://shssat.forsakringskassan.se:8020/rest/
:: produktion
SET FK_SHSURL=https://shs.forsakringskassan.se:7020/rest/

SET FK_CLIENTCERTFILE=my.crt
SET FK_CLIENTKEYFILE=my.key
SET FK_CACERTFILE=ca.crt
SET FK_CURLFILE=curl --remote-name --cert %FK_CLIENTCERTFILE% --key %FK_CLIENTKEYFILE% --cacert %FK_CACERTFILE%
SET FK_CURLCOMMAND=%FK_CURLFILE% "%FK_SHSURL%%FK_ONEFROMPRODUCT%/%%FK_TRANSACTION%"

:: Kör kommandot
ECHO "HÄMTAR INFORMATION (%FK_TRANSACTION%)."
echo %FK_CURLCOMMAND%
%FK_CURLCOMMAND%

:: Tar bort definierade variabler
SET FK_ONEFROMPRODUCT=
SET FK_TRANSACTION=
SET FK_SHSURL=
SET FK_CLIENTCERTFILE=
SET FK_CLIENTKEYFILE=
SET FK_CACERTFILE=
SET FK_CURLFILE=
SET FK_CURLCOMMAND=

ECHO "KLAR"
```

Datum
2017-10-18Version
F

10.4 Unixmiljö – getOneSHSFile

```
#!/usr/bin/ksh

# getOneSHSFile -T transaction [-P product]
# -P product = Hamtar filer for produkten med UUID product
# -T transaction = Hamta filer for en specifik transaktion
#

echo "INITIERAR..."

# satter produktid om inget annat anges
# motsvarar produkten FK.Sjukanmälan.XML.sat-saturnus
fk_productUUID=3aac5986-bda6-ff8d-4b6d-351b84166c32
# satter transaktionsid om inget annat anges
fk_transactionUUID=

until [[ $# -eq 0 ]]; do
    if [ "$1" == "-T" ]; then fk_transactionUUID=/${2}; fi
    if [ "$1" == "-P" ]; then fk_productUUID=${2}; fi
    shift
done

# test
fk_shsurl=https://shssat.forsakringskassan.se:8020/rest/
# production
fk_shsurl=https://shs.forsakringskassan.se:7020/rest/

fk_clientcertfile=my.crt
fk_clientkeyfile=my.key
fk_cacertfile=ca.crt

fk_curlfile="/usr/local/bin/curl --cert $fk_clientcertfile --key $fk_clientkeyfile --cacert $fk_cacertfile"
fk_curlurl="$fk_shsurl$fk_productUUID$fk_transactionUUID"

fk_curlcommand="$fk_curlfile --remote-name $fk_curlurl"

echo "CURL KOMMANDO..."
echo "$fk_curlcommand"
$fk_curlcommand

echo "AVSLUTAR..."
```

Datum
2017-10-18Version
F

10.5 Hämta flera filer

Windowsmiljö

```
:: Fil för att hämta flera produkter från en SHS
::
:: getSHSAllFiles.bat [-F] [-P product]
:: -F = Hämta filer med orginalfilnamn
1:: -P product = UUID för produkten som skall hämtas (annars används den som finns definierad nedan)
::
@echo OFF

ECHO "INITIERAR..."

:: Definierar produkten om inget skickas in
:: FK.Sjukanmälan.XML.sat-saturnus
SET FK_ALLFROMPRODUCT=3aac5986-bda6-ff8d-4b6d-351b84166c32

:: Kolla om parametern -P är satt
IF "-P"=="%1" SET FK_ALLFROMPRODUCT=%2
IF "-P"=="%2" SET FK_ALLFROMPRODUCT=%3

:: Kolla om parametern -F är satt
IF "-F"=="%1" SET FK_ORIGINALFILE=?getfilenames=true
IF "-F"=="%3" SET FK_ORIGINALFILE=?getfilenames=true

:: Filnamn temp lagring av transaktionsid
SET FK_TEMPFILENAME=temp-trans.txt
:: test
SET FK_SHSURL=https://shssat.forsakringskassan.se:8020/rest/
:: production
::SET FK_SHSURL=https://shs.forsakringskassan.se:7020/rest/

SET FK_CLIENTCERTFILE=my.crt
SET FK_CLIENTKEYFILE=my.key
SET FK_CACERTFILE=ca.crt
SET FK_CURLFILE=curl --cert %FK_CLIENTCERTFILE% --key %FK_CLIENTKEYFILE% --cacert %FK_CACERTFILE%
SET FK_CURLCOMMAND=%FK_CURLFILE% "%FK_SHSURL%%FK_ALLFROMPRODUCT%%FK_ORIGINALFILE%"

:: Kör kommandot till temporära filen
ECHO "HÄMTAR INFORMATION TILL %FK_TEMPFILENAME%..."
ECHO %FK_CURLCOMMAND%
%FK_CURLCOMMAND% > %FK_TEMPFILENAME%

::ECHO "Lista med transaktionsid..."
FOR /F %t IN (%FK_TEMPFILENAME%) DO ECHO %t%

:: Anropa hämtning för respektive transaktion
FOR /F %t IN (%FK_TEMPFILENAME%) DO CALL getSHSFile -P %FK_ALLFROMPRODUCT% -T %t%

:: Ta bort definierade variabler
ECHO "RENSAR..."
SET FK_ALLFROMPRODUCT=
SET FK_ORIGINALFILE=
SET FK_TEMPFILENAME=
SET FK_SHSURL=
SET FK_CLIENTCERTFILE=
SET FK_CLIENTKEYFILE=
SET FK_CACERTFILE=
SET FK_CURLFILE=
SET FK_CURLCOMMAND=

ECHO "KLAR"
```



Datum
2017-10-18

Version
F

10.6 Unixmiljö – getAllSHSFiles

```
#!/usr/bin/ksh

# getAllSHSFiles [-F] [-P product]
# -F = Hamta filer med originalnamn
# -P product = Hamtar filer för produkten med UUID product
#

echo "INITIERAR..."
fk_filenames=

# sätter produktid om inget annat anges
# motsvarar produkten FK.Sjukanmälan.XML.sat-saturnus
fk_productUUID=3aac5986-bda6-ff8d-4b6d-351b84166c32

until [[ $# -eq 0 ]]; do
    if [ "$1" == "-F" ]; then fk_filenames="?getfilenames=true"; fi
    if [ "$1" == "-P" ]; then fk_productUUID=$2; fi
    shift
done

# test
fk_shsurl=https://shssat.forsakringskassan.se:8020/rest/
# produktion
fk_shsurl=https://shs.forsakringskassan.se:7020/rest/

fk_clientcertfile=my.crt
fk_clientkeyfile=my.key
fk_cacertfile=ca.crt

fk_curlfile="/usr/local/bin/curl --cert $fk_clientcertfile --key $fk_clientkeyfile --cacert $fk_cacertfile"
fk_curlurl="$fk_shsurl$fk_productUUID$fk_filenames"

fk_curlcommand="$fk_curlfile $fk_curlurl"

echo "CURL KOMMANDO..."
echo "$fk_curlcommand"

# kör kommandot
$fk_curlcommand > temp_txid

cat temp_txid | while read txid
do
    getOneSHSFile -T $txid -P $fk_productUUID
done

echo "AVSLUTAR..."
```