Question
Paul Simon · Jan 19, 2018

How to Interoperate with CareEvolution

I've been trying to interoperate with careevoltion using their backend-services.
Spec: http://docs.smarthealthit.org/authorization/backend-services/

This involves creating a JWT (JSON Web Token) that I have been unable to do using %OAuth2.JWT:ObjectToJWT.

I downloaded jwt.pfx and then ran the following openssl commands to create some pem files.

```
openssl pkcs12 -in jwt.pfx -out file.nokey.pem -nokeys
openssl pkcs12 -in jwt.pfx -out file.withkey.pem

openssl rsa -in file.withkey.pem -out file.key

cat file.nokey.pem file.key > file.combo.pem3
```

I then ran some node code to create the private portions of the RSA key:

```
var fs = require('fs');
var rsaPemToJwk = require('rsa-pem-to-jwk');
var pem = fs.readFileSync('file.key');
console.log(pem);
var jwk = rsaPemToJwk(pem, {use: 'sig'}, 'private');
console.log(JSON.stringify(jwk));

C:\Users\Paul\Desktop\jwt>node rsa-pem-to-jwk.txt > c:\tmp\priv.txt
```

I then changed the node code so that it created the public portions of the RSA key.

```
var jwk = rsaPemToJwk(pem, {use: 'sig'}, 'public');

C:\Users\Paul\Desktop\jwt>node rsa-pem-to-jwk.txt > c:\tmp\public.txt
```

I then ran the following COS code:

```
KILL JOSE

S JOSE("keyalg")="RSA"
S JOSE("sigalg")="RS256"

S A=+$H-47117
S B=A*86400
S C=$P($HOROLOG,",",2)
S EPOCH=B+C
// add 4 minutes
S D=60*4
```

```
S EPOCH=EPOCH+D

set body=##class(%DynamicObject).%New()
do body.%Set("iss","JWTClientCredentials","string")
do body.%Set("aud","https://fhir.careevolution.com/Master.Adapter1.WebClient/identity
server/connect/token","string")
do body.%Set("exp",EPOCH,"string") // 1516032893
do body.%Set("nbf",EPOCH,"string") // 1516032893
do body.%Set("jti","db5ba9b3-602c-4b5f-beb9-4e8d76028c01","string")
do body.%Set("sub","07e5a735-a4f7-e711-8136-0a69c1b3225b","string")
do body.%Set("user_name","psimon","string")
do body.%Set("clientid","JWTClientCredentials","string")

// copied from /tmp/priv.txt (node output)

s localpriv="{ ""kty"": ""RSA"", ""n"": ""s0hguxNL5Xb6_Fk3u_fnZrUXuRnjj1wIEGXlsnqbu4r
ptiDYeX9dEQm29eDe5fhXWjrPtU33WA_zz0Y9R5z7EaX4ZlZG7PlFlm5vu3bEu-qeuMp8Xb3hPMyND-87JGrb
B1t5i-BgMpjQoelhYT8iqzE1fEtUGYyNn2dhSwIVwqIaRt4ly65oyW9Ea7VL92kmCEgUmIn-lrEfvygfbEL2H
-9dW42dAMvEwLCQGM7iX1zuHRAH9Ec9kwnaBb2kfHzoSJXZm5WcuPYJIWZrvu1RbHcZBFff0GwCDTH1bfECmr
6c-6BPeeUMw6resnbM4v3dTrbKD64HQbC33x4_Xs3pTQ"", ""e"": ""AQAB"", ""d"": ""FVOizh45hQ5
mROaIDsAqsrkQHWDLBR65htnYPScAp4qayqOVnL5d38z8Cru5SDoGiiE83CBuL_eV1S5R09cEttC7f9D7lu0A
Lijs-avjM0dxoiHUMYKI7KaYkTCwJGDhtTpYdx810k8DYn9UqjDMevjbl_GOC4wAvNmbZUTWOdTDfVXm7D42v
cUnh0e7nTYu7qGBRcfRrduzjBZMSddZoz8suWbJlQowUt6-bYOUPixeclML8oOKrtuQ_Y6fI-SNkaM5MrrJku
-s1KXiOt5ZGXkpqE-PUQsw1DAzs6MfS6CcuFdYadlBd6sn6bOLNJN3DffDJQGHQUIUefTJ2I_YMQ"", ""p""
: ""7CSQiUHeP_C2QBgQEsC_yF0Q1AFHmNEn9ua0lJ20LJSu7zzYCdE6E5X-j-9w37nzdzOClceEI2-2OAvV_
Ukh3R78q9_-Mtc9AUk2BunwzfaAdJHl0YT0IpJEI-IluMpOX9RETTK3KXpjOFBefHOeHS0DyWLRXDaAt2k_9c
1Q1Lc"", ""q"": ""wlvJ3Ap_i1vxt5q6zRT6JBxstvXLi0hQgHZnchvNluOfhzBTfYTSiK0n6Od-yj1c90i
Zt97B4XKWfps29xwd6wqS6X-QtG59AvlU0IOH4FSoYvyVy4yqpMSzAjgmhwZRi-API7XWd1KSmkTB8t-Xfdxj
dnC6kM62MqtsdC-Vhs"", ""dp"": ""WFFomV1AQUvG7fvR7yGV2Nst0wzTeU0olEg-26KL42yMbL-l0S4me
XLM7YpQ_evvKfLi8R_YxOQgE6AhnYR_nNLdD29MBDnKADQgd7-BJ5b8_hwfBxihslhgEcef8hf_7glWrkS8ik
_S0hoE7KjVRvYyB1zlDob35yD_IfBzPcs"", ""dq"": ""h5_NiIK65eBPGDQczicpNjGvmyyB0LuxkTMOlI
3aNMS5-Xg7ioc48q8B_oAr9axERzqeKbSDznJLmiVtgZpZNj62rcGalI3VJlIeYTKnil8I8aoYTWXnXf5bNY2
sTV32NQfIkHmMxOSqhqoz4Wia0a9tyfJ_FUG8urMT6dUkPKk"", ""qi"": ""WsLO8yNdoSMF3sYAISKhx_N
J5ua6PISaezUpYk6WgENHfUnwXXiDpDgx8AKae2vRpPaGoHhJkiih5JHTtVWNGUXB0-DItnX0jodM78NT31Xe
fKuE4Pg6nLtrpPlQpRqL6Gj1en2FcVtQ5TblOaFO9dkBoBB-gGWK9dy0WPR5D9w"" }"

// copied from /tmp/public.txt (node output)

s localpub="
{""kty"":""RSA"",""use"":""sig"",""n"":""ALaMFdNTmzUu1wnMGNmsIkBMJUS4EdMAe2HXpQ-k4aDk
94znJuCi9OJnJQU4lm4nqgEWdzMVVQMlqZ3ihlXx5vpFzmb4m4EkAcmK0ULVCLL9QRqvuwbOjeXW8pJlcH-EA
QKnkMzF6GbtmmDubK46bx2GaQh8rLYAfLvoIwXfyrXIday2VdRNz_3yGzLoxr5QxcFml46479mY3xXwmTSMvV
NHEN4F1xPnsLQoNPH3guA104dm8S-f2z_vczHBdSrgiJibesdOP3ugYlxJvDDT3WSf3WW7cjAn8M3vZmCNTPW
e6JEewsc6cDBiFWn9UUomulFUV_uucTsV_NXMd8nQ_Hs"",""e"":""AQAB""}"

SET ST=##class(%OAuth2.JWT).ObjectToJWT(.JOSE,.body,localpriv,localpub,.JWT)
```

If you run the above code in a terminal session - it does *not* work:

```
USER>d $system.OBJ.DisplayError(ST)

ERROR #5002: Cache error: <INVALID OREF>zGetJWK+2^%OAuth2.JWT.1
```

Killing JOSE and re-running the code works ... however, the JWT is invalid since I cannot get the POST that contains the JWT to successfully return an accesstoken?

```
USER>K JOSE
```

```
USER>SET ST=##class(%OAuth2.JWT).ObjectToJWT(.JOSE,.body,localpriv,localpub,.JWT)

?USER>ZW JWT
JWT="eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJpc3MiOiJodHRwczovL2ZoaXIuY2FyZWV2b2x1dGlv
bi5jb20vTWFzdGVyLkFkYXB0ZXIxLldlYkNsaWVudC8iLCJhdWQiOiJodHRwczovL2ZoaXIuY2FyZWV2b2x1d
Glvbi5jb20vTWFzdGVyLkFkYXB0ZXIxLldlYkNsaWVudC8iLCJleHAiOiIxNTE2MTIxNjA5IiwibmJmIjoiMT
UxNjEyMTYwOSIsImp0aSI6ImRiNWJhOWIzLTYwMmMtNGI1Zi1iZWI5LTRlOGQ3NjAyOGMwMSIsInN1YiI6IjA
3ZTVhNzM1LWE0ZjctZTcxMS04MTM2LTBhNjljMWIzMjI1YiIsInVzZXJfbmFtZSI6InBzaW1vbiIsImNsaWVu
dGlkIjoiT0F1dGhUZXN0In0."
```

So, can anyone please tell me what I'm doing wrong here?
To summarise what I'm trying to do is use the pfx cert to sign the JWT and then use that JWT to get hold of an accesstoken.
The attached C# code will apparently create a JWT that will work with careevolutions back-end services - but so far I've been unable to compile the code in VS-2017.

Finally, can Cache can be configured to support back-end services (similar to the above).
Spec: http://docs.smarthealthit.org/authorization/backend-services/

Thanks,
-- Paul.

#FHIR #JSON #Caché

Source URL:https://community.intersystems.com/post/how-interoperate-careevolution