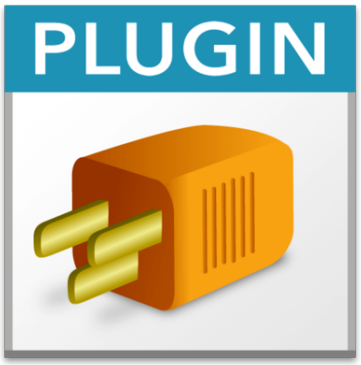
JWT RS256 authentication in FileMaker



Recently a client asked about JWT signatures. Here is an example calculation for **JWT with HS256** in FileMaker with MBS FileMaker Plugin. We prepare a header and payload string with some sample data. Inside the header we declare the algorithm to be HS256, which means we use HMAC with SHA256 as hashing algorithm. The payload is some JSON and depends on the web service you use, but for the example we specify it here directly. Next we encode both using Base64URL and do the HMAC and finally assemble everything:

```
Let ( [

secret = "secretkey";
header = "{\"alg\":\"HS256\",\"typ\":\"JWT\"}";
payload = "{\"loggedInAs\":\"admin\",\"iat\":1422779618}";

// calculate hash
encodedString = MBS( "Text.EncodeToBase64URL"; header; "UTF-8" ) &
"." & MBS( "Text.EncodeToBase64URL"; payload; "UTF-8" );
hash = MBS( "Hash.SHA256.HMAC"; secret; encodedString; 1+8 );

// and built final result:
result = encodedString & "." & hash
]; result )
```

Next we do **JWT RS256**, which means RSA signed JSON with SHA 256 as hashing algorithm. Like above we have some header and payload from fields, put them together with Base64URL and sign that with our RSA.Sign function. To sign we pass the text to sign, the key to use and the password for the key. The key is a RSA private key, which we just copied and pasted from a PEM file to the field, so it starts with a "-----BEGIN RSA PRIVATE KEY-----" line. If the key is encrypted, please pass the password, otherwise pass empty text for the password parameter. The output of the signature must be in Base64URL encoding to work with JWT.

Please note that you need a new base64url option there, which we add for version 10.3 of our plugin. If you like to try, email us for an early copy of the new plugin. Otherwise we have the 10.3pr1 next week.

Here is the full calculation:

```
Let ( [
secret = "secretkey";
header = JWT RS256::Header;
payload = JWT RS256::Payload;
encodedString = MBS( "Text.EncodeToBase64URL"; header; "UTF-8" ) &
"." & MBS( "Text.EncodeToBase64URL"; payload; "UTF-8" );

// calculate hash
hash = MBS( "RSA.Sign"; "SHA256";
"Text"; encodedString; "UTF-8";
"Text"; JWT RS256::Key; "UTF-8";
"text"; JWT RS256::Password; "UTF-8";
"base64url"; "");

// and built final result:
result = encodedString & "." & hash
]; result )
```

Please do not hesitate to contact us with questions. A sample file will be include with next plugin version.