Calculating the HMAC Signature

Required Authentication Information

Information	Description
{"video-id":"%videoId%","exp-time":"%expiryTime%""}	The message is the basis for the calculation of the signature, comprising • video-id: ID of the video • exp-time: time at which the signature is expired (epoch timestamp)
Shared secret	The key can be retrieved in the security settings in your VideoManager (see "VideoManager Manual: Security Policy Configuration").

The following code samples shows how to calculate a HMAC signature.

> Note that the sample functions accept a token lifetime value in minutes. This is then converted to seconds and added to the current epoch timestamp.

Java Code Sample

Java Example

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;
import java.math.*;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.time.Duration;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class VMProToken {

    public static void main(String[] args) {
        final String videoID = "212zpS6bjN77eixPUMUEjR";
        final String sharedSecret = "abc123";
        final Duration lifeTime = Duration.of(5, ChronoUnit.MINUTES); //token expires in 5 minutes

        try {
            final String token = generateToken(videoID, sharedSecret, lifeTime);
            System.out.println(token);
        } catch (NoSuchAlgorithmException | InvalidKeyException e) {
            e.printStackTrace();
        }
    }

    private static String generateToken(String videoId, String sharedSecret, Duration lifeTime)
            throws NoSuchAlgorithmException, InvalidKeyException {
        final String HASH_PATTERN = "{\"video-id\":\"%s\", \"exp-time\": %s}";
        final String HASH_ALGORITHM = "HmacSHA256";

        final long expiryTime = Instant.now().plus(lifeTime).getEpochSecond();

        final String tokenCalcBase = String.format(HASH_PATTERN, videoId, expiryTime);
        final Mac hmac = Mac.getInstance(HASH_ALGORITHM);
        final byte[] keyBytes = DatatypeConverter.parseHexBinary(sharedSecret);
        final SecretKeySpec secretKey = new SecretKeySpec(keyBytes, HASH_ALGORITHM);
        hmac.init(secretKey);
        final byte[] hmacBytes = hmac.doFinal(tokenCalcBase.getBytes());
        final String hash = String.format("%064x", new BigInteger(1, hmacBytes));

        return expiryTime + "~" + hash;
    }
}
```

Ruby Code Sample

Ruby-Example

```
require 'openssl'
require 'date'

videoId = "212zpS6bjN77eixPUMUEjR"
sharedSecret = ["abc123"].pack('H*') #Hex2Bin
lifeTime = 5

expiryTime = (Time.now.to_i + (lifeTime*60)).to_s

message = sprintf("{\"video-id\":\"%s\", \"exp-time\": %s}", videoId, expiryTime)
hmac = OpenSSL::HMAC.hexdigest('sha256', sharedSecret , message)
token = expiryTime + "~" + hmac

printf("\nToken: %s\n", token)
```

PHP Code Sample

PHP Example

```
<?php

$videoId = "212zpS6bjN77eixPUMUEjR";
$sharedSecret = "abc123";
$lifeTime = 5;

function generateToken($videoId, $sharedSecret, $lifeTime)
{
  $expiryTime = time() + ($lifeTime*60);
  $data = sprintf("{\"video-id\":\"%s\", \"exp-time\": %s}" , $videoId, $expiryTime);
  $hash = hash_hmac ( "sha256", $data , hex2bin($sharedSecret) );
  $token = sprintf ("%s~%s", $expiryTime , $hash);
  return $token;
}

$token = generateToken($videoId, $sharedSecret, $lifeTime);
echo $token;
?>
```