

Example code that builds the hash based on the PatronValidate method.

```
1  NSString *key = @"4B3C1D34-7BA6-4D5E-8112-1C9A6BB7CD74";
2  NSString *url = @"http://orion.polarislibrary.com/PAPIService/REST/public/v1/1033/100/1/patron/21756003332022";
3  NSString *hash = [PAPIHash calculateHash:key HTTPMethod:@"GET" URL:url Password:@"1234"];
```

Filename: PAPIHash.h

```
1  //
2  //  PAPIHash.h
3  //
4  //  Created by Jeffrey Young on 11/7/11.
5  //  Copyright (c) 2011 Polaris Library Systems. All rights reserved.
6  //
7
8  #import <Foundation/Foundation.h>
9
10 @interface PAPIHash : NSObject
11
12 + (NSString*)calculateHash:(NSString *)key HTTPMethod:(NSString *)httpMethod URL:(NSString *)url Password:(NSString *)pwd;
13
14 + (NSString *)encodeBase64:(NSData *)objData;
15
16 + (NSString*)rfc1123String;
17
18 @end
```

Filename: PAPIHash.m

```
1  //
2  //  PAPIHash.m
3  //
4  //  Created by Jeffrey Young on 11/7/11.
5  //  Copyright (c) 2011 Polaris Library Systems. All rights reserved.
6  //
7
8  #import "PAPIHash.h"
9  #import <CommonCrypto/CommonDigest.h>
10 #import <CommonCrypto/CommonHMAC.h>
11
12 @implementation PAPIHash
13
14 + (NSString*)calculateHash:(NSString *)key HTTPMethod:(NSString *)httpMethod URL:(NSString *)url Password:(NSString *)pwd {
15
16     // Add date - RFC1123
17     // Format: Tue, 08 Nov 2011 04:18:16 GMT
18
19     NSString *date = [self rfc1123String];
20     NSString *data = [NSString stringWithFormat:@"%s%s%s", httpMethod, url, date, pwd];
21
22     const char *cKey = [key cStringUsingEncoding:NSUTF8StringEncoding];
23     const char *cData = [data cStringUsingEncoding:NSUTF8StringEncoding];
24
25     unsigned char cHMAC[CC_SHA1_DIGEST_LENGTH];
26
27     CCHmac(kCCHmacAlgSHA1, cKey, strlen(cKey), cData, strlen(cData), cHMAC);
28 }
```

```

29     NSData *HMAC = [[NSData alloc] initWithBytes:cHMAC
30                     length:sizeof(cHMAC)];
31
32     NSString *hash = [self encodeBase64:HMAC];
33
34     return hash;
35 }
36
37 // https://github.com/mikeho/QSUtilities
38 static const char _base64EncodingTable[64] = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
39
40
41 + (NSString *)encodeBase64:(NSData *)objData {
42     const unsigned char * objRawData = [objData bytes];
43     char * objPointer;
44     char * strResult;
45
46     // Get the Raw Data length and ensure we actually have data
47     int intLength = [objData length];
48     if (intLength == 0) return nil;
49
50     // Setup the String-based Result placeholder and pointer within that placeholder
51     strResult = (char *)calloc(((intLength + 2) / 3) * 4, sizeof(char));
52     objPointer = strResult;
53
54     // Iterate through everything
55     while (intLength > 2) { // keep going until we have less than 24 bits
56         *objPointer++ = _base64EncodingTable[objRawData[0] >> 2];
57         *objPointer++ = _base64EncodingTable[((objRawData[0] & 0x03) << 4) + (objRawData[1] >> 4)];
58         *objPointer++ = _base64EncodingTable[((objRawData[1] & 0x0f) << 2) + (objRawData[2] >> 6)];
59         *objPointer++ = _base64EncodingTable[objRawData[2] & 0x3f];
60
61         // we just handled 3 octets (24 bits) of data
62         objRawData += 3;
63         intLength -= 3;
64     }
65
66     // now deal with the tail end of things
67     if (intLength != 0) {
68         *objPointer++ = _base64EncodingTable[objRawData[0] >> 2];
69         if (intLength > 1) {
70             *objPointer++ = _base64EncodingTable[((objRawData[0] & 0x03) << 4) + (objRawData[1] >> 4)];
71             *objPointer++ = _base64EncodingTable[(objRawData[1] & 0x0f) << 2];
72             *objPointer++ = '=';
73         } else {
74             *objPointer++ = _base64EncodingTable[(objRawData[0] & 0x03) << 4];
75             *objPointer++ = '=';
76             *objPointer++ = '=';
77         }
78     }
79
80     // Terminate the string-based result
81     *objPointer = '\0';
82
83     // Return the results as an NSString object
84     return [NSString stringWithCString:strResult encoding:NSUTF8StringEncoding];
85 }
86
87 + (NSString*)rfc1123String
88 {
89     NSDate *today = [NSDate date];

```

```
90
91     static NSDateFormatter *df = nil;
92     if(df == nil)
93     {
94         df = [[NSDateFormatter alloc] init];
95         //df.locale = [[[NSLocale alloc] initWithLocaleIdentifier:@"en_US"] autorelease];
96         df.timeZone = [NSTimeZone timeZoneWithAbbreviation:@"GMT"];
97         df.dateFormat = @"EEE', ' dd MMM yyyy HH':'mm':'ss 'GMT'";
98     }
99     return [df stringFromDate:today];
100 }
101
102
103 @end
```