```
/*
 * Copyright (C) 2000-2018 Deutsche Post DHL Group. All rights reserved.
 * This file and its contents are the sole property of Deutsche Post DHL Group.
 *
 * Redistribution and use in source and binary forms, with or without
modification,
 * are permitted provided that the following conditions are met:
 *
 *   1. Redistributions of source code must retain the above copyright notice,
 *      this list of conditions and the following disclaimer.
 *   2. Redistributions in binary form must reproduce the above copyright
notice,
 *      this list of conditions and the following disclaimer in the
documentation
 *      and/or other materials provided with the distribution.
 *   3. Neither the name of the copyright holder nor the names of its
contributors
 *      may be used to endorse or promote products derived from this software
 *      without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
 * IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE
 * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
 * ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
using System;
using System.Security.Cryptography;
using System.Text;

/*
 * Deutsche Post DHL Group - Internetmarke.
 *
 * Example of an MD5 signing / hashing algorithm, to generate the hashes to
 * be enclosed in the Internetmarke interface SOAP headers.
 *
 * You may freely use this snippet in your own programs for Internetmarke,
 * according to the license regulations.
 */
namespace DPDHL
{
    /// <summary>
    /// Example snippet for the MD5 hash calculation for SOAP headers.
    /// </summary>
    class MD5Sample
    {
        // Length of the hash string for the SOAP header
        const int HASH_STRING_LENGTH = 8;
        const string SEPARATOR = "::";

        /// <summary>
        /// Starting method for the example program snippet.
        /// </summary>
        /// <param name="args">Command line parameters, not used here.</param>
        static void Main(string[] args)
        {
            MD5Sample md5Sample = new DPDHL.MD5Sample();

            const string partnerId = "DPPAR";
            const string signatureKey = "Das_Ist_der_neue_Key";

            string partnerSignature = md5Sample.sign(partnerId, signatureKey);

            Console.WriteLine("Partner Signature: " + partnerSignature);
        }

        /// <summary>
        /// Generate an Internetmarke SOAP header signature, based on the given 
attributes.
        /// The signature (4 bytes hash in Base16 format) is returned in a 
string.
        /// </summary>
        /// <param name="partnerId">The PartnerID assigned to you by the 
Internetmarke team.</param>
        /// <param name="signatureKey">The signature key assigned to you by the 
Internetmarke team.</param>
        /// <returns>The signature, as 8 char hash in Base16 encoding.</returns>
        public string sign(string partnerId, string signatureKey)
        {
            return sign(partnerId, "1", signatureKey);
        }

        /// <summary>
        /// Generate an Internetmarke SOAP header signature, based on the given 
attributes.
        /// The signature (4 bytes hash in Base16 format) is returned in a 
string.
        /// </summary>
        /// <param name="partnerId">The PartnerID assigned to you by the 
Internetmarke team.</param>
        /// <param name="keyPhase">The key phase to use, as described in the 
Internetmarke specs.</param>
        /// <param name="signatureKey">The signature key assigned to you by the 
Internetmarke team.</param>
        /// <returns>The signature, as 8 char hash in Base16 encoding.</returns>
        public string sign(string partnerId, string keyPhase, string 
signatureKey)
        {
            DateTime timestamp = DateTime.Now;
            // Day - Month - Year - Hour - Min - Sec
            string requestTimestamp = timestamp.ToString("ddMMyyyy-HHmmss");
            // For debugging, set a fixed date time to verify hash
            requestTimestamp = "28012014-142729";

            string temp = partnerId + SEPARATOR +
                requestTimestamp + SEPARATOR +
                keyPhase + SEPARATOR +
                signatureKey;

            // A bit of debugging, to see what is going on...
            Console.WriteLine("Input string for MD5 hash: " + temp);

            try
            {
                MD5 md5 = new MD5CryptoServiceProvider();
                byte[] bsig = Encoding.ASCII.GetBytes(temp);

                // Do the MD5 hashing using a built-in lib
                byte[] md5ByteHash = md5.ComputeHash(bsig);

                // Now convert to string, as 4 Base16 encoded bytes (8 char)
                StringBuilder md5HashString = new StringBuilder(32);
                foreach (byte md5Byte in md5ByteHash)
                {
                    md5HashString.Append(md5Byte.ToString("x2"));
                }

                return md5HashString.ToString().Substring(0,
HASH_STRING_LENGTH);
            }
            catch (ArgumentException ex)
            {
                // Add logging or error handling here...
                Console.WriteLine("Can't generate signature! " + ex);
            }

            // Exit if hash could not be calculated.
            return null;
        }
    }
}
```