

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Security.Cryptography;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace Authentication.Model
10 {
11     public class Common
12     {
13         public static string Encrypt(String text)
14         {
15             RijndaelManaged RijndaelAlg = new RijndaelManaged();
16             CryptoStream cStream = null;
17
18             try
19             {
20                 byte[] key = { 0x02, 0x1A, 0x03, 0x04C, 0x05, 0x06, 0xAB, 0x08,  ↗
21                             0x09, 0xEF, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };
22                 byte[] IV = { 0x01, 0x11, 0x10, 0x99, 0x19, 0x06, 0x07, 0x28,  ↗
23                             0x09, 0x10, 0x11, 0x12, 0x33, 0x14, 0x15, 0x56 };
24                 byte[] inputByteArray = Encoding.UTF8.GetBytes(text);
25
26                 MemoryStream memoryS = new MemoryStream();
27                 cStream = new CryptoStream(memoryS, RijndaelAlg.CreateEncryptor  ↗
28                     (key, IV), CryptoStreamMode.Write);
29                 cStream.Write(inputByteArray, 0, inputByteArray.Length);
30                 cStream.FlushFinalBlock();
31                 return Convert.ToBase64String(memoryS.ToArray());
32             }
33             catch (Exception e)
34             {
35                 string y = e.Message.ToString();
36                 return null;
37             }
38         }
39         finally
40         {
41             if (cStream != null)
42                 cStream.Close();
43         }
44     }
45
46     public static string Decrypt(String text)
47     {
48         Rijndael RijndaelAlg = Rijndael.Create();
49         CryptoStream cStream = null;
```

```

...e\New Version\Authorization\Authorization\Model\Common.cs 2
50         byte[] key = { 0x02, 0x1A, 0x03, 0x04C, 0x05, 0x06, 0xAB, 0x08, 0x09, 0xEF, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };
51         byte[] IV = { 0x01, 0x11, 0x10, 0x99, 0x19, 0x06, 0x07, 0x28, 0x09, 0x10, 0x11, 0x12, 0x33, 0x14, 0x15, 0x56 };
52         byte[] inputByteArray = Convert.FromBase64String(text);
53         MemoryStream memoryS = new MemoryStream();
54         cStream = new CryptoStream(memoryS, RijndaelAlg.CreateDecryptor(key, IV), CryptoStreamMode.Write);
55         cStream.Write(inputByteArray, 0, inputByteArray.Length);
56         cStream.FlushFinalBlock();
57         Encoding encode = Encoding.UTF8;
58         return encode.GetString(memoryS.ToArray());
59     }
60     catch (Exception e)
61     {
62         string y = e.Message.ToString();
63         return null;
64     }
65     finally
66     {
67         if (cStream != null)
68             cStream.Close();
69     }
70 }
71
72 public static string GenerateToken()
73 {
74     using (var cryptoProvider = new RNGCryptoServiceProvider())
75     {
76         var secretKeyByteArray = new byte[32]; //256 bit
77         cryptoProvider.GetBytes(secretKeyByteArray);
78         return Convert.ToBase64String(secretKeyByteArray);
79     }
80 }
81 }
82 }
83

```