



This repository Search

[Explore](#) [Features](#) [Enterprise](#) [Blog](#)[Sign up](#)[Sign in](#)

pvginkel / ProgressEncode

Star 3

Fork 1

branch: master

ProgressEncode / ProgressEncode / Progress.cs



Fetching contributors...



80 lines (66 sloc) 3.669 kb

[Raw](#)[Blame](#)[History](#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace ProgressEncode
6 {
7     public static class Progress
8     {
9         private static readonly ushort[] _lookup =
10         {
11             0, 49345, 49537, 320, 49921, 960, 640, 49729, 50689, 1728, 1920, 51009, 1280, 50625, 50305, 1088, 52225, 32
12             2560, 51905, 52097, 2880, 51457, 2496, 2176, 51265, 55297, 6336, 6528, 55617, 6912, 56257, 55937, 6720, 768
13             56385, 5120, 54465, 54657, 5440, 55041, 6080, 5760, 54849, 53761, 4800, 4992, 54081, 4352, 53697, 53377, 41
14             62081, 12864, 13824, 63169, 63361, 14144, 62721, 13760, 13440, 62529, 15360, 64705, 64897, 15680, 65281, 16
15             14592, 63937, 63617, 14400, 10240, 59585, 59777, 10560, 60161, 11200, 10880, 59969, 60929, 11968, 12160, 61
16             9600, 58689, 9984, 59329, 59009, 9792, 8704, 58049, 58241, 9024, 57601, 8640, 8320, 57409, 40961, 24768, 24
17             42689, 42881, 26432, 42241, 26048, 25728, 42049, 27648, 44225, 44417, 27968, 44801, 28608, 28288, 44609, 43
18             26688, 30720, 47297, 47489, 31040, 47873, 31680, 31360, 47681, 48641, 32448, 32640, 48961, 32000, 48577, 48
19             47041, 46721, 30272, 29184, 45761, 45953, 29504, 45313, 29120, 28800, 45121, 20480, 37057, 37249, 20800, 37
20             38721, 21760, 38337, 38017, 21568, 39937, 23744, 23936, 40257, 24320, 40897, 40577, 24128, 23040, 39617, 39
21             18624, 18816, 35137, 19200, 35777, 35457, 19008, 19968, 36545, 36737, 20288, 36097, 19904, 19584, 35905, 17
22             34369, 33281, 17088, 17280, 33601, 16640, 33217, 32897, 16448
23         };
24
25         public static string Encode(byte[] input)
26         {
27             if (input == null)
28                 throw new ArgumentNullException("input");
29
30             byte[] scratch = new byte[16];
31
32             ushort hash = 17;
33
34             for (int i = 0; i < 5; i++)
35             {
36                 for (int j = 0; j < input.Length; j++)
37                 {
38                     scratch[15 - (j % 16)] ^= input[j];
39                 }
40
41                 for (int j = 0; j < 16; j += 2)
42                 {
43                     hash = Hash(scratch, hash);
44
45                     scratch[j] = (byte)(hash & 0xff);
46                     scratch[j + 1] = (byte)((hash >> 8) & 0xff);
47                 }
48             }
49
50             byte[] target = new byte[16];
51
52             for (int i = 0; i < 16; i++)
53             {
54                 byte lower = (byte)(scratch[i] & 0x7f);
55
56                 if (lower >= 'A' && lower <= 'Z' || lower >= 'a' && lower <= 'z')
57                     target[i] = lower;
```

```
58         else
59             target[i] = (byte)((scratch[i] >> 4) + 0x61);
60     }
61
62     return Encoding.ASCII.GetString(target);
63 }
64
65 private static ushort Hash(byte[] scratch, ushort hash)
66 {
67     for (int i = 15; i >= 0; i--)
68     {
69         hash = (ushort)(
70             hash >> 8 ^
71             _lookup[hash & 0xff] ^
72             _lookup[scratch[i]]
73         );
74     }
75
76     return hash;
77 }
78 }
79 }
```