

Geração de Chave para o Web Service

Chave para o Unity:

-hmy|:djCle@|6&ydd;/)~/cs,'w5P9s" `h_g+PM}IFU-#~T-\$M6DU%3E[T@^kw

(lembrando que, no C#, as aspas deverão estar como \", pois tem uma aspa nessa chave – posição 32).

Passos para gerar a chave para se logar no web service:

1) Crie uma lista de inteiros:

```
List<byte> lsCaracteres = new List<byte>();
```

2) Obtenha os códigos ASCII de cada caractere:

```
lsCaracteres = Encoding.ASCII.GetBytes(strChave).ToList();
```

3) Depois de pegar os caracteres, pegue a data e hora atual, dividindo-as e some com os valores da lista feita anteriormente:

```
string strData = DateTime.Now.ToString("yyyyMMddHHmmss");
```

```
List<byte> lsData = Encoding.ASCII.GetBytes(strData);
```

```
for (int intLetra = 0; intLetra < lsData.Length; intLetra++)
{
    byte byteLetra = lsCaracteres[intLetra];
    byteLetra += lsData[intLetra];
    lsCaracteres[intLetra] = byteLetra;
}
```

4) Agora faça a mesma coisa com a senha, só que em vez de iniciar na posição 0 é necessário iniciar na posição 26, finalizando com um caractere número 127:

```
List<byte> lsSenha = Encoding.ASCII.GetBytes(strSenha);
```

```
for (int intLetra = 0; intLetra < lsData.Length; intLetra++)
{
    byte byteLetra = lsCaracteres[26 + intLetra];
    byteLetra += lsSenha[intLetra];
    lsCaracteres[26 + intLetra] = byteLetra;
}
```

```
byte byteLetra = lsCaracteres[26 + intLetra];
byteLetra += 127;
lsCaracteres[26 + intLetra] = byteLetra;
```

5) Agora é necessário transformar esses números decimais em hexadecimais, colocando um 0 antes dos números que não tiverem:

```
string strChaveSimples = "";
```

```
foreach (int intCodASC in lsCaracteres)
{
    strChaveSimples += String.Format("{0:X3}", intCodASC);
}
```

5) Agora é necessário transformar essa string gerada em base 64:

```
byte[] byteB64 = Encoding.UTF8.GetBytes(strChaveSimples);
string strB64 = Convert.ToBase64String(byteB64);
```

6) Agora é a parte inusitada. É necessário agora comprimir com Gzip a string:

```
byte[] byteZip = Encoding.UTF8.GetBytes(strB64);
```

```
byte[] byteRes;
```

```
using (MemoryStream msEntrada = new MemoryStream(byteZip))
```

```
{
```

```
    using (MemoryStream msSaida = new MemoryStream())
```

```
    {
```

```
        using (GzipStream gzsZip = new GzipStream(msSaida, CompressionMode.Compress))
```

```
        {
```

```
            byte[] byteCompactar = new byte[4096];
```

```
            int intValor;
```

```
            while ((intValor = msEntrada.Read(byteCompactar, 0, byteCompactar.Length)) != 0)
```

```
            {
```

```
                gzsZip.Write(byteCompactar, 0, intValor);
```

```
            }
```

```
        }
```

```
        byteRes = msSaida.ToArray();
```

```
    }
```

```
}
```

7) Agora sim você tem a chave gerada pelo Gzip, então vamos converter para uma linguagem que a URL vai entender:

```
string strRes = Encoding.UTF8.GetBytes(byteRes);
```

```
string strParamChave = HttpContext.Current.Server.UrlEncode(strRes);
```