Url Encode Base 64

```
#region License
// Copyright (c) 2007 James Newton-King
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated documentation
// files (the “Software”), to deal in the Software without
// restriction, including without limitation the rights to use,
// copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the
// Software is furnished to do so, subject to the following
// conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
// OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
// NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
// WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
// OTHER DEALINGS IN THE SOFTWARE.
```

```
#endregion

using System;
using System.Collections.Generic;
using System.Text;
using System.Collections.Specialized;
using System.Web;

namespace Newtonsoft.Utilities.Web
{
public static class UrlUtils
{
public static string GetSchemeHostPort(Uri uri)
{
if (uri == null)
throw new ArgumentNullException(“uri”);

return uri.Scheme + “://” + uri.Host + “:” + uri.Port + “/”;
}

public static string UrlEncodeBase64(string base64Data)
{
return new string(UrlEncodeBase64(base64Data.ToCharArray()));
}

public static char[] UrlEncodeBase64(char[] base64Data)
{
for (int i = 0; i < base64Data.Length; i++) { switch
(base64Data[i]) { case '+': base64Data[i] = '@'; break; case
'/': base64Data[i] = '$'; break; } } return base64Data; }
public static string UrlDecodeBase64(string base64Data) {
return new string(UrlDecodeBase64(base64Data.ToCharArray()));
} public static char[] UrlDecodeBase64(char[] base64Data) {
for (int i = 0; i < base64Data.Length; i++) { switch
(base64Data[i]) { case '@': base64Data[i] = '+'; break; case
'$': base64Data[i] = '/'; break; } } return base64Data; } } }
[/csharp]
```