

حتما برای شما هم پیش آمده‌است که در پروژه‌ای نیاز داشتید تا رشته‌ای تصادفی را تولید کنید. کد تصادفی میتواند کاربردهایی چون تولید رمز، تولید شناسه، تولید url، تولید کد فعال سازی و مواردی از این قبیل را داشته باشد. احتمالا برای ساخت کد یا رشته تصادفی، اولین چیزی که به ذهن شما می‌رسد، استفاده از کلاس random می‌باشد. اما روش‌های خلاقانه و جالب زیادی وجود دارند که برای این کار استفاده می‌شوند. در اینجا می‌خواهیم تعدادی از آنها را با هم بررسی کنیم.

روش‌های تولید اعداد یا رشته تصادفی:

1- معمول‌ترین روش تولید یک کد شش رقمی با استفاده از کلاس random

```
[TestMethod]
public void TestRandomClass()
{
    var code = new Random().Next(100000, 999999);
    Assert.IsTrue(code.ToString().Length == 6);
}
```

2- تولید با استفاده کلاس Random و Enumerable

```
[TestMethod]
public void TestRandomWithEnumerable()
{
    var chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    var random = new Random();
    var result = new string(
        Enumerable.Repeat(chars, 6)
        .Select(s => s[random.Next(s.Length)])
        .ToArray());
    Assert.IsTrue(result.Length == 6);
}
```

البته بدیهی هست که در قسمت chars می‌توانید هر نوع کاراکتری را قرار دهید و کد نهایی بر آن مبنا تولید می‌شود. مثلا می‌توانید فقط اعداد را مشخص کنید و در این حالت رشته‌ی خروجی فقط شامل رقم خواهد بود. اگر خواستید رشته‌ی طولانی‌تری را تولید کنید، کافی ست طول مورد نیاز را با عدد 6 در کد بالا جایگزین کنید. مثلا برای تولید رمز عبور از لیست زیر می‌تونید استفاده کنید:

```
var chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz~!@#%&^*";
```

3- تولید کد با استفاده از guid

```
Guid.NewGuid().ToString().Replace("-", string.Empty).Substring(0, 6);
```

یا

```
Guid.NewGuid().ToString("n").Substring(0, 6);
```

کد ToString("n") کاراکترهای غیر عددی را از رشته‌ی مورد نظر حذف میکند.

4 - تولید با استفاده از کلاس RNGCryptoServiceProvider

بعضی‌ها روش‌های ویژه را می‌پسندند. البته استفاده از این کلاس مزایا و معایب خودش را دارد. از نظر سرعت نسبت به کلاس random پایین‌تر هست، چون محاسبات بیشتری دارد.

```
public static string GetUniqueKey(int maxSize)
{
    char[] chars = new char[62];
    chars =
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890".ToCharArray();
    byte[] data = new byte[1];
    using (RNGCryptoServiceProvider crypto = new RNGCryptoServiceProvider())
    {
        crypto.GetNonZeroBytes(data);
        data = new byte[maxSize];
        crypto.GetNonZeroBytes(data);
    }
    StringBuilder result = new StringBuilder(maxSize);
    foreach (byte b in data)
    {
        result.Append(chars[b % (chars.Length)]);
    }
    return result.ToString();
}
```

5 - استفاده از متد Path.GetRandomFileName

کاربرد اصلی این متد در واقع تولید نام فایلی تصادفی است؛ ولی از آن برای تولید رشته هم استفاده میکنند. متد Path.GetRandomFileName در پشت صحنه از همان کلاس RNGCryptoServiceProvider برای تولید نام فایل استفاده میکند.

```
public string Get8CharacterRandomString()
{
    string path = Path.GetRandomFileName();
    path = path.Replace(".", ""); // Remove period.
    return path.Substring(0, 6); // Return 6 character string
}
```

6- تولید کد با استفاده از کلاس random و linq

```
var chars = "abcdefghijklmnopqrstuvwxyz123456789".ToArray();
string pw = Enumerable.Range(0, passwordLength)
    .Aggregate(
        new StringBuilder(),
        (sb, n) => sb.Append((chars[random.Next(chars.Length)])),
        sb => sb.ToString());
```

نتیجه

مطمئناً روش‌های زیادی برای تولید رشته تصادفی وجود دارند و البته همه شباهت‌هایی نیز دارند و در لایه‌های پایین‌تر، دارای اصولی مشترک هستند. موارد بالا فقط روش‌های متفاوت تولید کد نهایی را نشان می‌دهند که شما بسته به نیاز خود میتوانید از آنها استفاده کنید.

شما چه روش‌هایی را برای این کار می‌شناسید؟

نظرات خوانندگان

نویسنده: احمد رجبی
تاریخ: ۱۴:۵ ۱۳۹۴/۰۳/۱۱

[یک نکته:](#)

There are several common mistakes I've seen developers make. The first is to assume that Random is thread-safe and is ok to be used concurrently from multiple threads

...

نویسنده: وحید نصیری
تاریخ: ۱۶:۲۳ ۱۳۹۴/۰۴/۲۴

یک نکته‌ی تکمیلی

[کتابخانه‌ی رمزنگاری اطلاعات Inferno](#) به همراه یک تولید کننده‌ی اعداد تصادفی [thread safe](#) است.