عنوان: **تعیین اعتبار یک GUID در دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۱/۰۷ ۱۹:۰۳:۴۳
آدرس: www.dotnettips.info
برچسب‌ها: Security

GUID یا Globally unique identifier یک عدد صحیح 128 بیتی است (بنابراین 2 به توان 128 حالت را می‌توان برای آن درنظر گرفت). از لحاظ آماری تولید دو GUID یکسان تقریبا صفر می‌باشد. به همین جهت از آن با اطمینان می‌توان به عنوان یک شناسه منحصربفرد استفاده کرد. برای مثال اگر به لینک‌های دانلود فایل‌ها از سایت مایکروسافت دقت کنید، این نوع GUID ها را به وفور می‌توانید ملاحظه نمائید. یا زمانیکه قرار است فایلی را که بر روی سرور آپلود شده، ذخیره نمائیم، می‌توان نام آن‌را یک GUID درنظر گرفت بدون اینکه نگران باشیم آیا فایل آپلود شده بر روی یکی از فایل‌های موجود overwrite می‌شود یا خیر. یا مثلا استفاده از آن در سناریوی بازیابی کلمه عبور در یک سایت. هنگامیکه کاربری درخواست بازیابی کلمه عبور فراموش شده خود را داد، یک GUID برای آن تولید کرده و به او ایمیل می‌زنیم و در آخر آن‌را در کوئری استرینگی دریافت کرده و با مقدار موجود در دیتابیس مقایسه می‌کنیم. مطمئن هستیم که این عبارت قابل حدس زدن نیست و همچنین یکتا است.

برای تولید GUID ها در دات نت می‌توان مانند مثال زیر عمل کرد و خروجی‌های دلخواهی را با فرمت‌های مختلفی دریافت کرد:

```
System.Guid.NewGuid().ToString() = 81276701-9dd7-42e9-b128-81c762a172ff
System.Guid.NewGuid().ToString("N") = 489ecfc61ee7403988efe8546806c6a2
System.Guid.NewGuid().ToString("D") = 119201d9-84d9-4126-b93f-be6576eedbfd
System.Guid.NewGuid().ToString("B") = {fd508d4b-cbaf-4f1c-894c-810169b1d20c}
System.Guid.NewGuid().ToString("P") = (eee1fe00-7e63-4632-a290-516bfc457f42)
```

تمام این‌ها خیلی هم خوب! اما همان سناریوی مشخص ساختن یک فایل با GUID و یا بازیابی کلمه عبور فراموش شده را درنظر بگیرید. یکی از اصول امنیتی مهم، تعیین اعتبار ورودی کاربر است. چگونه باید یک GUID را به صورت مؤثری تعیین اعتبار کرد و مطمئن شد که کاربر از این راه قصد تزریق اس کیوال را ندارد؟
دو روش برای انجام اینکار وجود دارد
الف) عبارت دریافت شده را به new Guid پاس کنیم. اگر ورودی غیرمعتبر باشد، یک exception تولید خواهد شد.
ب) استفاده از regular expressions جهت بررسی الگوی عبارت وارد شده

پیاده سازی این دو را در کلاس زیر می‌توان ملاحظه نمود:

```
using System;
using System.Text.RegularExpressions;

namespace sample
{
  /// <summary>
  /// بررسی اعتبار یک گوئید
  /// </summary>
  public static class CValidGUID
  {
      /// <summary>
      /// بررسی تعیین اعتبار ورودی
      /// </summary>
      /// <param name="guidString">ورودی</param>
      /// <returns></returns>
      public static bool IsGuid(this string guidString)
      {
          if (string.IsNullOrEmpty(guidString)) return false;

          bool bResult;
          try
          {
              Guid g = new Guid(guidString);
              bResult = true;
          }
          catch
```

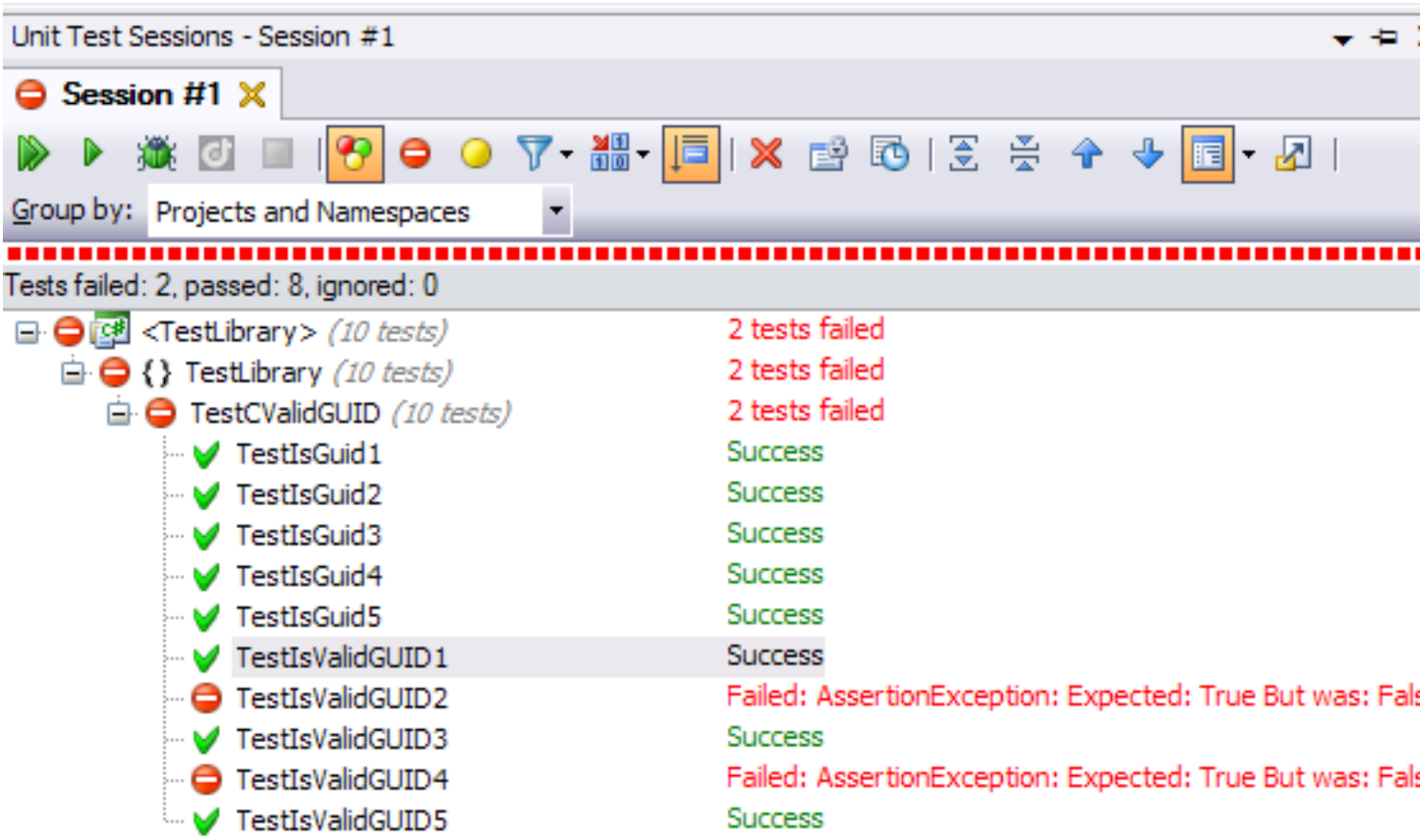
```
            {
                bResult = false;
            }

            return bResult;
        }

        /// <summary>
        /// بررسی تعیین اعتبار ورودی
        /// </summary>
        /// <param name="input">ورودی</param>
        /// <returns></returns>
        public static bool IsValidGUID(this string input)
        {
            return !string.IsNullOrEmpty(input) &&
                new Regex(@"^(\{{0,1}([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){12}\}{0,1})$").IsMatch(input);
        }
    }

}
```

سؤال: آیا متدهای فوق (extension methods) درست کار می‌کنند و واقعا نیاز ما را برآورده خواهند ساخت؟ به همین منظور، آزمایش واحد آن‌ها را نیز تهیه خواهیم کرد:

```
using NUnit.Framework;
using sample;

namespace TestLibrary
{
  [TestFixture]
  public class TestCValidGUID
  {

      /*************************************************************************/
      [Test]
      public void TestIsGuid1()
      {
          Assert.IsTrue("81276701-9dd7-42e9-b128-81c762a172ff".IsGuid());
      }

      [Test]
      public void TestIsGuid2()
      {
          Assert.IsTrue("489ecfc61ee7403988efe8546806c6a2".IsGuid());
      }

      [Test]
      public void TestIsGuid3()
      {
          Assert.IsTrue("{fd508d4b-cbaf-4f1c-894c-810169b1d20c}".IsGuid());
      }

      [Test]
      public void TestIsGuid4()
      {
          Assert.IsTrue("(eee1fe00-7e63-4632-a290-516bfc457f42)".IsGuid());
      }

      [Test]
      public void TestIsGuid5()
      {
          Assert.IsFalse("81276701;9dd7;42e9-b128-81c762a172ff".IsGuid());
      }


      /*************************************************************************/
      [Test]
      public void TestIsValidGUID1()
      {
          Assert.IsTrue("81276701-9dd7-42e9-b128-81c762a172ff".IsValidGUID());
      }

      [Test]
      public void TestIsValidGUID2()
      {
```

```
            Assert.IsTrue("489ecfc61ee7403988efe8546806c6a2".IsValidGUID());
        }

        [Test]
        public void TestIsValidGUID3()
        {
            Assert.IsTrue("{fd508d4b-cbaf-4f1c-894c-810169b1d20c}".IsValidGUID());
        }

        [Test]
        public void TestIsValidGUID4()
        {
            Assert.IsTrue("(eee1fe00-7e63-4632-a290-516bfc457f42)".IsValidGUID());
        }

        [Test]
        public void TestIsValidGUID5()
        {
            Assert.IsFalse("81276701;9dd7;42e9-b128-81c762a172ff".IsValidGUID());
        }
    }

}
```

نتیجه این آزمایش به صورت زیر است:



همانطور که ملاحظه می‌کنید حالت دوم یعنی استفاده از عبارات باقاعده دو حالت را نمی‌تواند بررسی کند (مطابق الگوی بکار گرفته شده که البته قابل اصلاح است)، اما روش معمولی استفاده از new Guid ، تمام فرمت‌های تولید شده توسط دات نت را پوشش می‌دهد.