### SQL Server引用dll

## **在**[SQL](http://www.51testing.com/html/45/javascript:;" \t "http://www.51testing.com/html/45/_self)**Server中引用dll分为两个步骤**

1.创建一个dll文件。

2.把dll文件放进SQL Server的程序集中。然后定义一个Function，就可以通过该Function来引用dll中的函数。

## **创建一个dll文件**

1. 点击“文件”->“新建”->“项目”->类库，把命名空间改成ClassLibrary2，添加如下代码。

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Security.Cryptography;

using System.IO;

// 注意：密钥和向量必须为8位，否则加密解密都不成功

namespace ClassLibrary2

{

public partial class DES

{

public static string Encrypt(string sourceString, string key, string iv)

{

try

{

byte[] btKey = Encoding.UTF8.GetBytes(key);

byte[] btIV = Encoding.UTF8.GetBytes(iv);

DESCryptoServiceProvider des = new DESCryptoServiceProvider();

using (MemoryStream ms = new MemoryStream())

{

byte[] inData = Encoding.UTF8.GetBytes(sourceString);

try

{

using (CryptoStream cs = new CryptoStream(ms, des.CreateEncryptor(btKey, btIV), CryptoStreamMode.Write))

{

cs.Write(inData, 0, inData.Length);

cs.FlushFinalBlock();

}

return Convert.ToBase64String(ms.ToArray());

}

catch

{

return sourceString;

}

}

}

catch { }

return "DES加密出错";

}

//调用： string recordString = Encrypt("金胖子死了", "20111219", "12345678");

// 结果：recordString= "xQ969nexy964SXhkTuekUQ=="

public static string Decrypt(string encryptedString, string key, string iv)

{

byte[] btKey = Encoding.UTF8.GetBytes(key);

byte[] btIV = Encoding.UTF8.GetBytes(iv);

DESCryptoServiceProvider des = new DESCryptoServiceProvider();

using (MemoryStream ms = new MemoryStream())

{

byte[] inData = Convert.FromBase64String(encryptedString);

try

{

using (CryptoStream cs = new CryptoStream(ms, des.CreateDecryptor(btKey, btIV), CryptoStreamMode.Write))

{

cs.Write(inData, 0, inData.Length);

cs.FlushFinalBlock();

}

return Encoding.UTF8.GetString(ms.ToArray());

}

catch

{

return encryptedString;

}

}

}

}

}

2.右击“项目”，点击”生成“或者直接按F6

3.右击“项目”，点击“在文件资源管理系统中打开文件夹”，点击bin->debug,然后就会看到一个ClassLibrary2.dll文件。

## **在SQL中引用dll文件**

1、数据库设置--------启用clr enabled配置选项

exec sp\_configure 'show advanced options', '1';

go

reconfigure;

go

exec sp\_configure 'clr enabled', '1'

go

reconfigure;

exec sp\_configure 'show advanced options', '1';

go

1. 引入dll文件

使用SQL语句 create assembly TestDll from 'C:\Users\Documents\Visual Studio 2012\Projects\StringHelp\StringHelp\obj\Debug\ClassLibrary2.dll'

　　其中TestDll是你在[数据库](http://www.51testing.com/html/45/javascript:;" \t "http://www.51testing.com/html/45/_self)中为这个程序集起的名字，‘’里面是dll的路径。

　　注意，如果你的dll文件中，引用了其他dll文件，那么必须在引入dll之前，先引用其程序集。如你的dll文件使用到了system.web dll文件，那么必须在引用TestDll之前以同样的方法引用system.web dll文件。

1. 创建一个Function，使用该dll文件

使用如下SQL语句

CREATE FUNCTION dbo.DESEncrypt( --该函数名字

@InputString1 as nvarchar(500),

@InputString2 as nvarchar(500),

@InputString3 as nvarchar(500)

)

RETURNS nvarchar(200) --返回类型

AS EXTERNAL NAME TestDll.[ClassLibrary2.DES].Encrypt

TestDll是指你程序集中dll的名称。

ClassLibrary2是指dll文件中那个类的命名空间

DES是指dll文件中那个类的类名

Encrypt是指dll文件中那个被调用的静态方法

最后，我们执行sql：

SELECT [graduation\_project].[dbo].[DESDecrypt]('金胖子死了', '20111219','12345678');

(1 行受影响)