原文地址：<https://www.cnblogs.com/siyi/p/5009032.html>

//动态加载程序集

Assembly ass = Assembly.LoadFile(@"F:\Cheng\_20140819\VS2012\_WorkSpace\Assembly\Assembly\TestClassLibary\bin\Debug\TestClassLibary.dll");

//利用GetTypes获取所有类型 public internal 都能拿到

Type[] ts = ass.GetTypes();

foreach (Type item in ts)

{

Console.WriteLine(item.Name);

}

//只获取公开的类型

//获取导出的类型

Type[] publicTypes = ass.GetExportedTypes();

foreach (Type item in publicTypes)

{

Console.WriteLine("--->>"+item.Name);

}

//获取特定的Type，类型需要完全限定名称 ： 记得带上命名空间哟~  
Type personType = ass.GetType("TestClassLibary.Person");

//根据特定的Type 来创建对象，无参数构造函数  
object personObj = Activator.CreateInstance(personType);

//获取特定的方法  
MethodInfo method = personType.GetMethod("Add");

//调用，参数1 代表获取的类型的对象 ，参数2  代表调用的方法的参数列表  
object obj= method.Invoke(personObj, new object[] { 2, 3 });

//获取参数的类型

ParameterInfo[] args = methodInfo.GetParameters();

Args[0].ParameterType

//获取继承接口的类型

Type[] types = type.GetInterfaces()

//方法设置泛型

methodInfo.MakeGenericMethod(type);

//--------------------------------调用属性---------------------

Assembly ass = Assembly.LoadFile (@"F:\Cheng\_20140819\VS2012\_WorkSpace\Assembly\Assembly\TestClassLibary\bin\Debug\TestClassLibary.dll");

Type t = ass.GetType ("TestClassLibary.Person");

object obj = Activator.CreateInstance (t);

PropertyInfo pifo = t.GetProperty ("Name");

pifo.SetValue (obj, "cheng");

MethodInfo mifo = t.GetMethod ("WriteName");

mifo.Invoke (obj, null);

//--------------------------------利用特定的构造函数

Assembly ass = Assembly.LoadFile [(@"F:\Cheng\_20140819\VS2012\_WorkSpace\20151129\_Heima\_Day11\_Assembly\Assembly\TestClassLibary\bin\Debug\TestClassLibary.dll");](mailto:(@\"F:/Cheng_20140819/VS2012_WorkSpace/20151129_Heima_Day11_Assembly/Assembly/TestClassLibary/bin/Debug/TestClassLibary.dll\");)

Type personType = ass.GetType ("TestClassLibary.Person");

//这里使用 ctor    GetConstructor()  其参数列表就 是指定的构造函数的列表

ConstructorInfo ctor = personType.GetConstructor (new Type[] { typeof (string), typeof (int), typeof (string) });

object o = ctor.Invoke (new object[] { "ccc", 20, "CCC@sina.cn" });

Console.WriteLine (o.ToString ());

MethodInfo mInfo = personType.GetMethod ("WriteName");

mInfo.Invoke (o, null);