使用CSharpCompilation实现动态编译

// 添加要引用的程序集

List<MetadataReference> refs = new List<MetadataReference>() {

MetadataReference.CreateFromFile (typeof (object).Assembly.Location),

MetadataReference.CreateFromFile (typeof (List<int>).Assembly.Location),

MetadataReference.CreateFromFile (typeof (ASCIIEncoding).Assembly.Location),

MetadataReference.CreateFromFile (typeof (JsonConvert).Assembly.Location),

MetadataReference.CreateFromFile (typeof (IEManageSystemCMSModule).Assembly.Location),

MetadataReference.CreateFromFile (typeof (Entity).Assembly.Location),

};

// 生成C#编译

var cSharpCompilation = CSharpCompilation

.Create(Guid.NewGuid().ToString() + ".dll")

.WithOptions(new CSharpCompilationOptions(

Microsoft.CodeAnalysis.OutputKind.DynamicallyLinkedLibrary,

usings: null,

optimizationLevel: OptimizationLevel.Debug, // TODO

checkOverflow: false, // TODO

allowUnsafe: true, // TODO

platform: Platform.AnyCpu,

warningLevel: 4,

xmlReferenceResolver: null

))

.AddReferences(refs);

MemoryStream ms = new MemoryStream();

// 要编译的源码

string code = $@"

using System;

using System.Collections.Generic;

using System.Text;

public class ClassA {{

}}

"

cSharpCompilation

// 添加语法树

.AddSyntaxTrees(CSharpSyntaxTree.ParseText(code))

// 生成程序集流

// 这里将流导入到内存流中

// 如果要生成dll文件，使用 Emit(dllPath)

.Emit(ms);