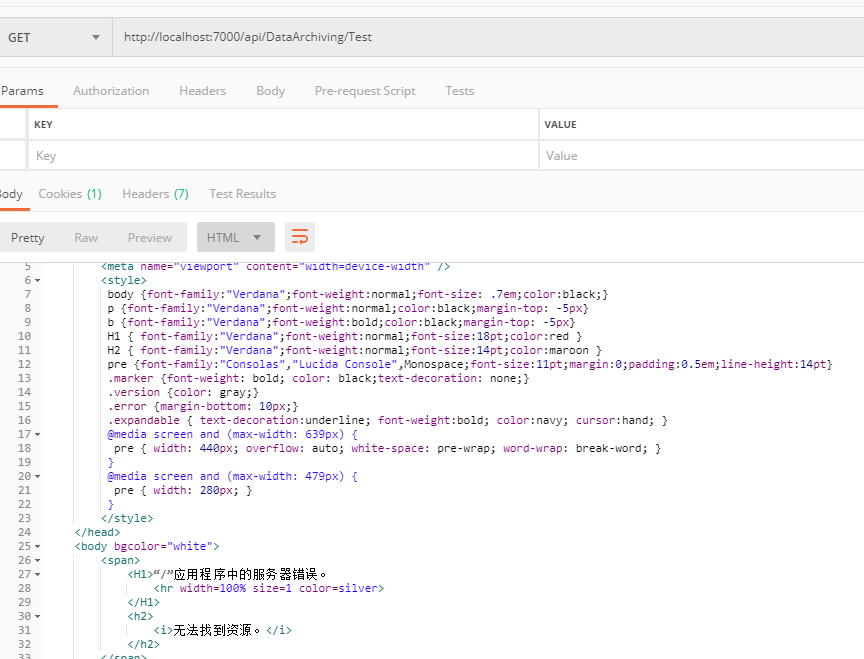
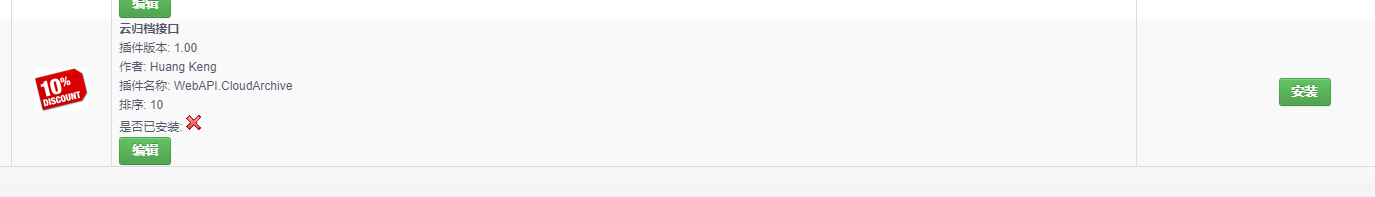
能力开放平台——正式动工

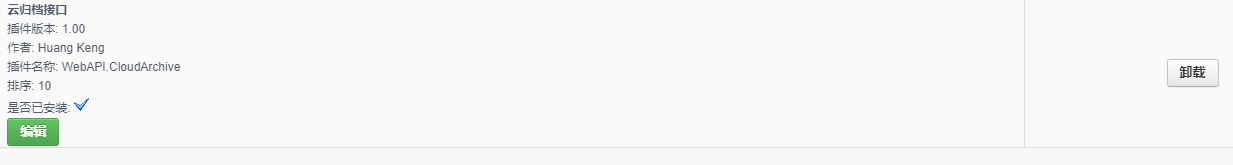
1.

接口无法找到资源，

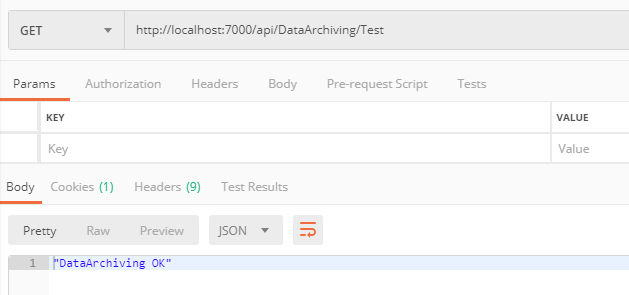


安装插件：





接口通了，开森。



2.

安装代码检查器。

除了eWorld.Entity之外，都给加上。

3.

添加第一个方法：

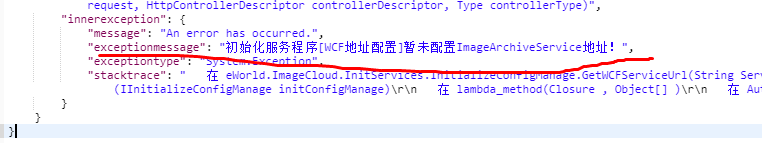
"error:System.NullReferenceException: 未将对象引用设置到对象的实例。\r\n 在 eWorld.Plugin.WebAPI.CloudArchive.Controllers.PublicController.Hua()"

应该是云归档的wcf没有配置地址。感觉。

4.

调用crm的wcfClient没有问题，ImageArchive就有问题了。

查了错误，是没有配置。



直接去源代码里面找：

/// <summary>

/// 获取WCF地址，如CRMService

/// </summary>

/// <param name="ServiceName">服务名称</param>

/// <param name="ReturnCompleteWCFUrl">是否返回完整的WCF地址，false仅返回IP</param>

/// <returns></returns>

public string GetWCFServiceUrl(string ServiceName, bool ReturnCompleteWCFUrl = true)

{

if (\_cacheWCFConnectSetting == null || \_cacheWCFConnectSetting.Count <= 0)

\_cacheWCFConnectSetting = InvokeApiGetWCFConnectString();

var currentWCFItem = \_cacheWCFConnectSetting.FirstOrDefault(s => s.Name == ServiceName);

if (currentWCFItem == null || string.IsNullOrEmpty(currentWCFItem.Url))

没有配置，是这个报错

throw new Exception(string.Format("初始化服务程序[WCF地址配置]暂未配置{0}地址！", ServiceName));

Uri hostUri = new Uri(currentWCFItem.Url);

if (!ReturnCompleteWCFUrl)

return hostUri.ToString();

else

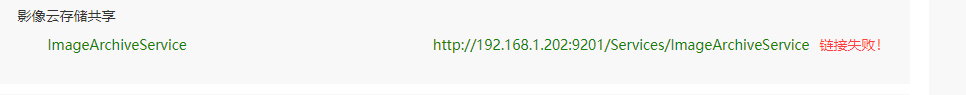
{

Uri retUri = new Uri(hostUri, string.Format("/Services/{0}.svc", ServiceName));

return retUri.ToString();

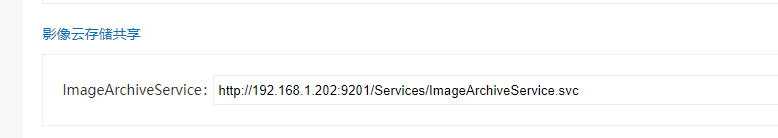
}

}



应该是这个的缘故，我改下配置。

加了svc



去202获取这个记录数据的xml文件，对的啊，有值的。

http://192.168.1.202:9201/Services/ImageArchiveService.svc

//初始化配置工具、WCF客户端调用

builder.RegisterType<InitializeConfigManage>().As<IInitializeConfigManage>().WithParameters(new List<NamedParameter>()

{

new NamedParameter("systemCode",1100),

new NamedParameter("initServiceSystemKey","bac11f3bd919f9fb")

}).SingleInstance();

知道原因了：

public enum SystemCodeEnum

{

门户 = 10,

公共服务 = 20,

远程医疗 = 100,

网络医院 = 200,

影像存储 = 300,

医学教育 = 400,

典型病例 = 500,

影像质控 = 600,

科研协同 = 700,

智能诊断 = 800,

影像大数据 = 900,

客户关系管理 = 1100,

运营管理 = 1200,

机构管理 = 1210,

在线监控 = 1300,

数据挖掘 = 1400,

能力开放 = 9100

}

这里是根据code的编码获取的，我这里给的是1100，所以我要换成300才行。

用军哥的1100获取的结果：

{

"data": [

{

"Name": "CommonService",

"Url": "http://192.168.1.202:9002/Services/CommonService.svc"

},

{

"Name": "MediaService",

"Url": "http://192.168.1.202:9002/Services/MediaService.svc"

},

{

"Name": "CRMService",

"Url": "http://192.168.1.202:9101/Services/CRMService.svc"

},

{

"Name": "PaymentService",

"Url": "http://192.168.1.202:9101/Services/PaymentService.svc"

}

],

"code": 0,

"msg": "操作成功"

}

用云归档的300获取的结果：

{

"data": [

{

"Name": "CommonService",

"Url": "http://192.168.1.202:9002/Services/CommonService.svc"

},

{

"Name": "MediaService",

"Url": "http://192.168.1.202:9002/Services/MediaService.svc"

},

{

"Name": "CRMService",

"Url": "http://192.168.1.202:9101/Services/CRMService.svc"

},

{

"Name": "ImageArchiveService",

"Url": "http://192.168.1.202:9201/Services/ImageArchiveService.svc"

}

],

"code": 0,

"msg": "操作成功"

}

开始报新的错：



两个方法重名了：

"exceptionmessage": "An exception was thrown while invoking the constructor 'Void .ctor(eWorld.ImageCloud.InitServices.IInitializeConfigManage)' on type 'eWorldWCFClient`1'. ---> 同一个协定中不能存在两个名称相同的操作，类型为 eWorld.IDCAS.WCFServerContract.ServerContract.IImageArchiveService 的方法 GetPatientExamDocumentList 和 GetPatientExamDocumentList 违反了此规则。可以通过更改方法名称或使用 OperationContractAttribute 的 Name 属性更改其中一个操作的名称。 (See inner exception for details.)",

暂时在方法后面加一个WithShort

/// <summary>

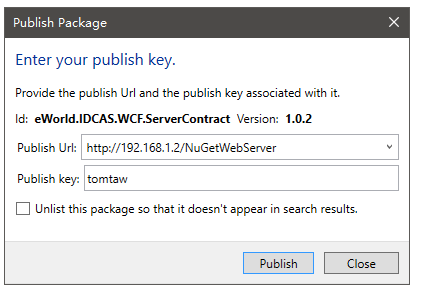
/// 获取文档信息(使用短链)

/// </summary>

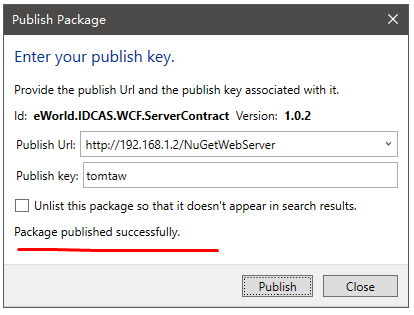
/// <param name="query">请求参数</param>

/// <returns>检查文档列表信息</returns>

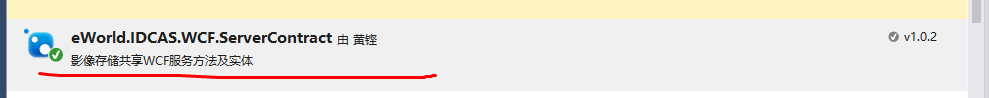
BaseListReturnResult<DocumentViewDTO> GetPatientExamDocumentListWithShort (DocumentQuery query);



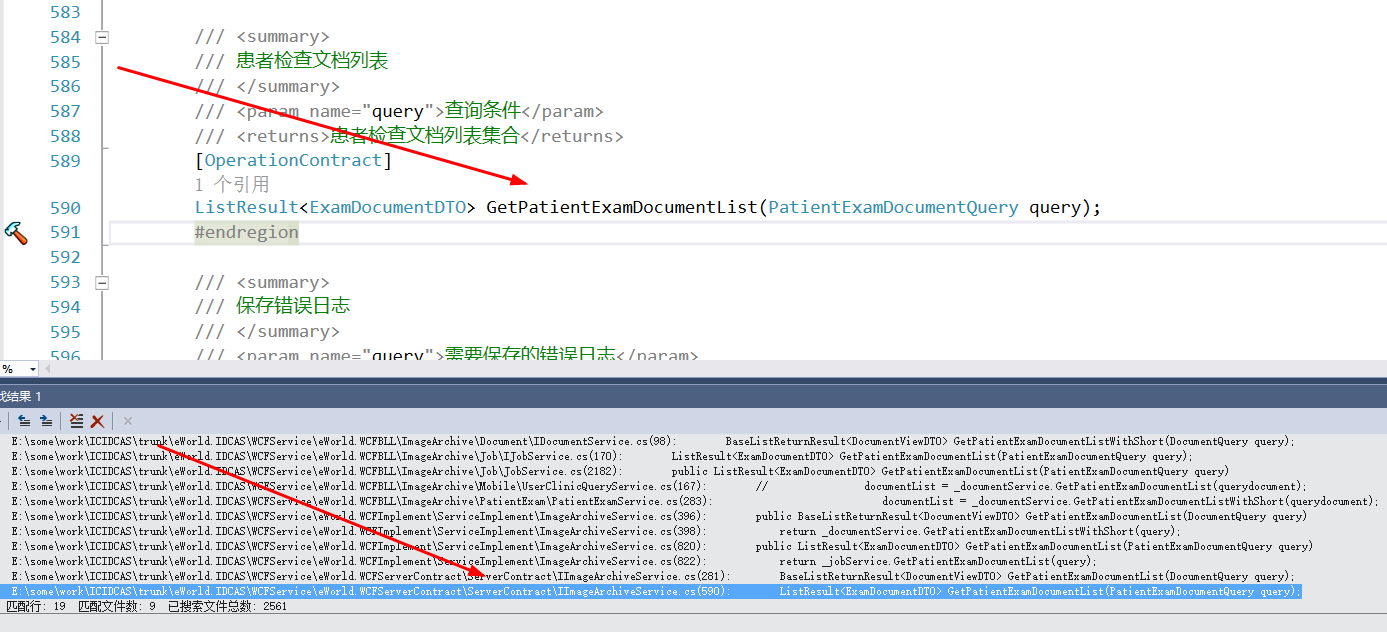
应该是成功了：



好了

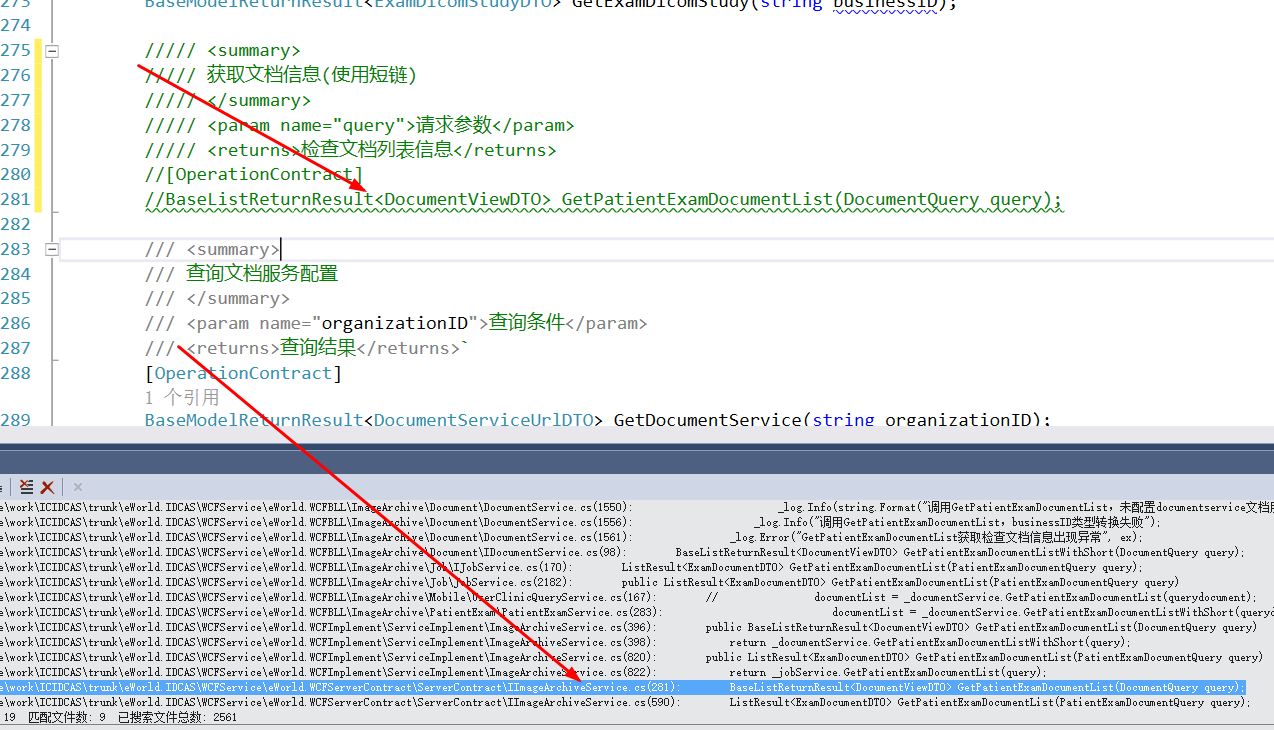


还有重名的，在ImageArchiveService中。先注释掉了。

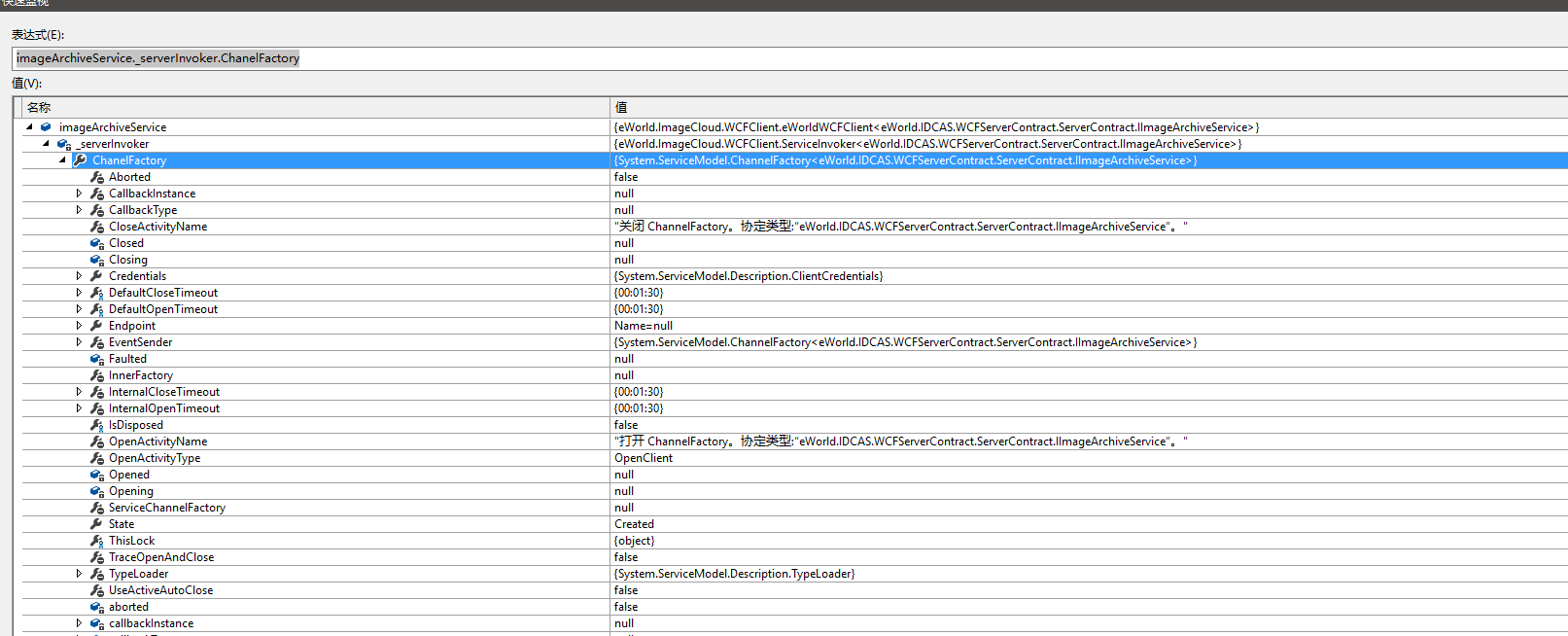


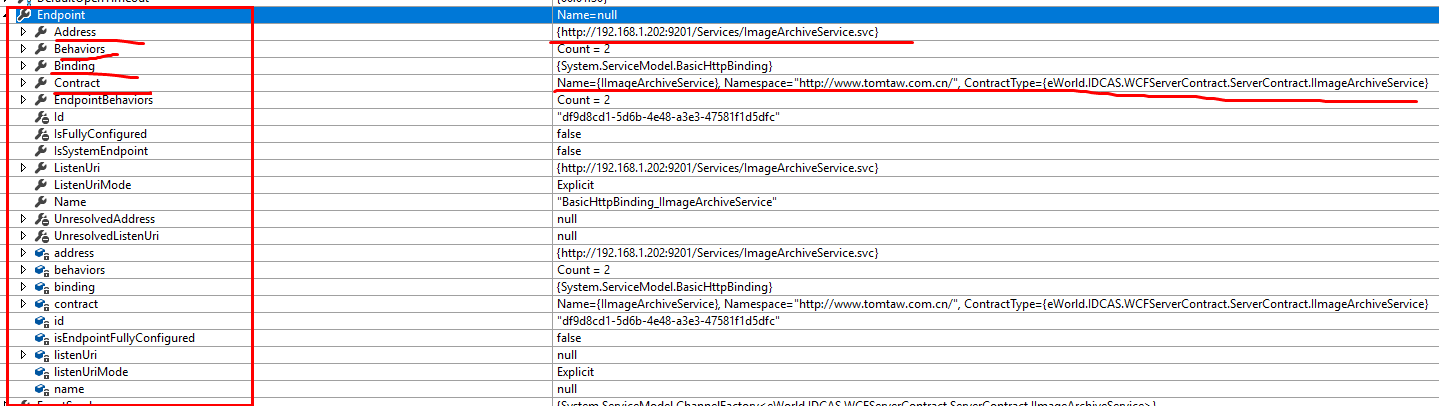
一共两个，一个是调用之前的DocumentService中的GetPatientExamDocumentList(DocumentQuery query)方法，（获取文档信息(使用短链)）

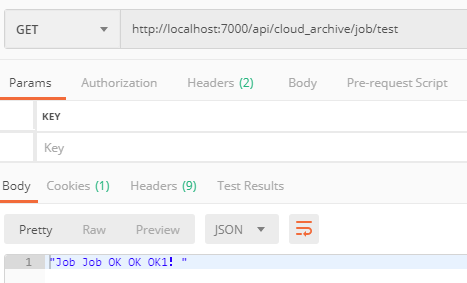
还有一个是调用JobService中的GetPatientExamDocumentList(PatientExamDocumentQuery query)方法。



来了来了，开森







Good Job！

调用的时候发生报错：



修改一下地址：

<http://localhost:9301/Services/ImageArchiveService.svc>

<http://192.168.1.202:9201/Services/ImageArchiveService.svc>

地址变成了本地，但是报错啊：

None of the constructors found with

'Autofac.Core.Activators.Reflection.DefaultConstructorFinder'

on type 'eWorld.IDCAS.WCF.BLL.Common.Configuration.DataDicInfoService'

can be invoked with the available services and parameters:Cannot resolve parameter 'eWorld.Core.Data.ICommonRepository`1[eWorld.Entity.Domain.Common.Configuration.DataDicInfo] datadicinfoRepository'

of constructor

'Void .ctor(eWorld.Core.Data.ICommonRepository`1[eWorld.Entity.Domain.Common.Configuration.DataDicInfo]

使用可用的服务和参数可以调用在

'eWorld.IDCAS.WCF.BLL.Common.Configuration.DataDicInfoService'类型中使用'Autofac.Core.Activators.Reflection.DefaultConstructorFinder'找到的构造函数：

无法解析参数

'eWorld.Core.Data.ICommonRepository`1 [eWorld.Entity.Domain.Common.Configuration.DataDicInfo] datadicinfoRepository'的构造函数

'Void.ctor（eWorld.Core.Data.ICommonRepository .1

[eWorld.Entity.Domain.Common.Configuration.DataDicInfo]]

"None of the constructors found with 'Autofac.Core.Activators.Reflection.DefaultConstructorFinder' on type 'eWorld.IDCAS.WCF.BLL.Common.Configuration.DataDicInfoService' can be invoked with the available services and parameters:\r\nCannot resolve parameter 'eWorld.Core.Data.ICommonRepository`1[eWorld.Entity.Domain.Common.Configuration.DataDicInfo] datadicinfoRepository' of constructor 'Void .ctor(eWorld.Core.Data.ICommonRepository`1[eWorld.Entity.Domain.Common.Configuration.DataDicInfo])'."

修改了这个：



5.

升级到Cloud 1.4后的报错：

几个方法：

（1）

/// <summary>

/// 获取患者平台检查列表

/// </summary>

/// <param name="query">查询条件</param>

/// <returns>获取患者平台检查列表的结果</returns>

public PageResult<PatientRelatedExamListDTO> GePatientPlatformExamList(ArchiveRecordQuery query)

（2）

#region Job触发器（JobTriggerMst）

/// <summary>

/// 根据条件查询Job触发器列表信息

/// </summary>

/// <param name="query">查询条件</param>

/// <returns>查询结果</returns>

PageResult<JobTriggerInfo> GetJobTriggerList(JobTriggerQuery query);

（3）

契约当中有些没加标签

#region Job触发器（JobTriggerMst）

/// <summary>

/// 根据条件查询Job触发器列表信息

/// </summary>

/// <param name="query">查询条件</param>

/// <returns>查询结果</returns>

PageResult<JobTriggerInfo> GetJobTriggerList(JobTriggerQuery query);

/// <summary>

/// 查询Job触发器及Job定义列表信息

/// </summary>

/// <param name="serviceUID">采集服务唯一号</param>

/// <param name="jobUID">作业唯一号</param>

/// <param name="groupUID">分组唯一号</param>

/// <returns>查询结果</returns>

ListResult<JobTriggerWithJobDefineInfo> GetJobTriggerWithJobDefineList(Guid? serviceUID, Guid? jobUID, Guid? groupUID);

/// <summary>

/// 保存Job触发器信息

/// </summary>

/// <param name="jobTriggerInfo">需要保存的Job触发器信息</param>

/// <returns>保存结果</returns>

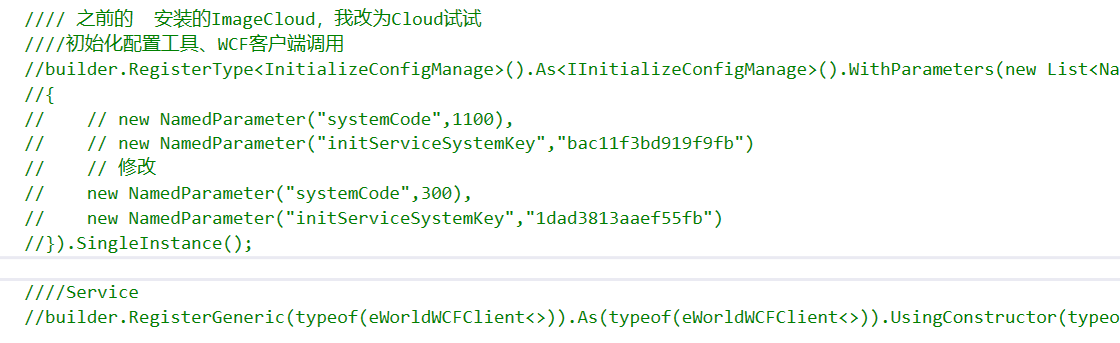
DataResult<Guid> SaveJobTrigger(JobTriggerInfo jobTriggerInfo);

"找不到方法:

“eWorld.Cloud.Result.ListResult`1<eWorld.IDCAS.WCF.ServerContract.Models.DTO.Info.DocumentViewDTO> eWorld.IDCAS.WCFServerContract.ServerContract.IImageArchiveService.GetExamDocumentList(eWorld.IDCAS.WCF.ServerContract.Models.DTO.Query.DocumentQuery)”。"

（4）

把注入，之前的ImageCloud改成Cloud试试。

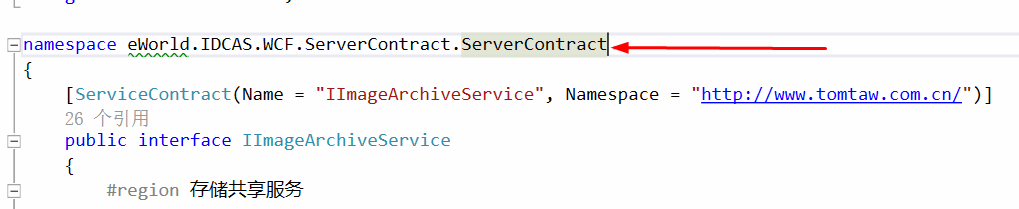


删除了，换成Cloud



修改了数据库的配置。

（5）



{"未能从程序集

“eWorld.IDCAS.WCF.ServerContract, Version=1.0.2.6694, Culture=neutral, PublicKeyToken=null”

中加载类型“eWorld.IDCAS.WCF.ServerContract.IImageArchiveService”。":"eWorld.IDCAS.WCF.ServerContract.IImageArchiveService"}

（6）

crm中的wcf



（7）

注入改回去了：

//// 之前的 安装的ImageCloud，我改为Cloud试试

////初始化配置工具、WCF客户端调用

//builder.RegisterType<InitializeConfigManage>().As<IInitializeConfigManage>().WithParameters(new List<NamedParameter>()

//{

// // new NamedParameter("systemCode",1100),

// // new NamedParameter("initServiceSystemKey","bac11f3bd919f9fb")

// // 修改

// new NamedParameter("systemCode",300),

// new NamedParameter("initServiceSystemKey","1dad3813aaef55fb")

//}).SingleInstance();

////Service

//builder.RegisterGeneric(typeof(eWorldWCFClient<>)).As(typeof(eWorldWCFClient<>)).UsingConstructor(typeof(IInitializeConfigManage)).InstancePerHttpRequest();

//要删除

//初始化配置工具、WCF客户端调用

builder.RegisterType<InitializeConfigManage>().As<IInitializeConfigManage>().WithParameters(new List<NamedParameter>()

{

//new NamedParameter("systemCode",1100),

//new NamedParameter("initServiceSystemKey","bac11f3bd919f9fb")

//修改

new NamedParameter("systemCode",300),

new NamedParameter("initServiceSystemKey","1dad3813aaef55fb")

}).SingleInstance();

//Service

builder.RegisterGeneric(typeof(eWorldWCFClient<>)).As(typeof(eWorldWCFClient<>)).UsingConstructor(typeof(IInitializeConfigManage)).InstancePerHttpRequest();

// builder.RegisterType<ImageArchiveService>().As<IImageArchiveService>()

//.WithParameter(ResolvedParameter.ForNamed<ICacheManager>("nop\_cache\_static"))

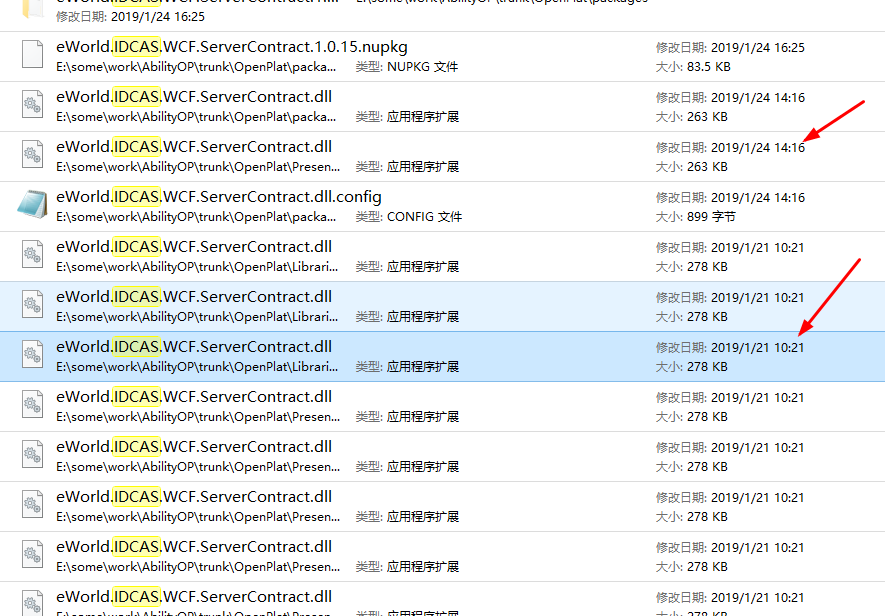
//.InstancePerHttpRequest();

////要删除

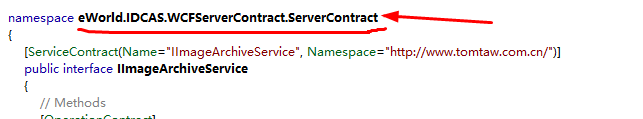
（8）

把重复的，之前的dll去除试试。

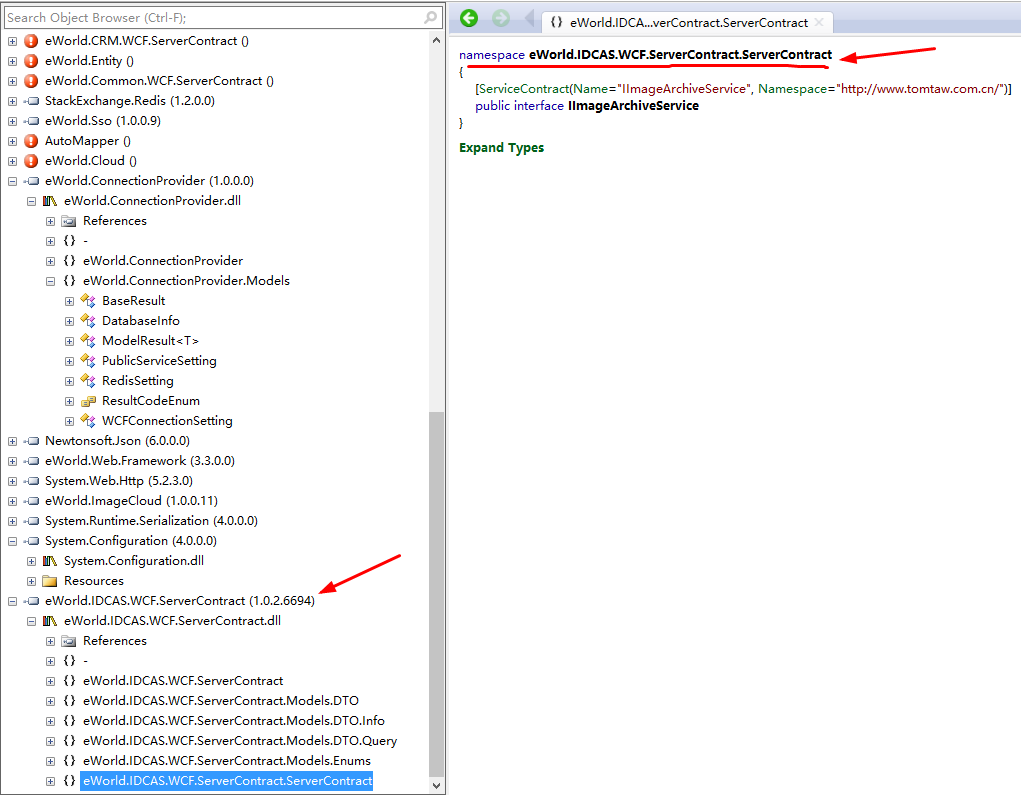
发现删除了，但编译之后，重新生成的还是之前版本的契约dll



反编译之后：



新的，多一个点的。



但为什么版本号1.02.6694是一样的？

（9）

搞定了，就是陈旧的dll错乱的问题，把eWorld.Core里面的引用去除后，再把所有的dll删除后，就好了。

控制器的注入依旧用了 ImageCloud

public ThirdPartyController(

ImageCloud.WCFClient.eWorldWCFClient<IImageArchiveService> imageArchiveService,

ImageCloud.InitServices.IInitializeConfigManage initializeConfigManage,

ImageCloud.WCFClient.eWorldWCFClient<ICRMService> crmClient)

{

this.\_imageArchiveService = imageArchiveService;

this.\_initializeConfigManage = initializeConfigManage;

this.\_crmClient = crmClient;

}

但暂时切换到本地的9301调用，因为可以调试，而且数据库也正常。

测试wawa

<http://localhost:7000/api/cloud_archive/ThirdParty/wa_wa>

get请求，函数体：

[HttpGet]

public object wa\_wa()

{

string wcf = null;

try

{

DocumentQuery param = new DocumentQuery()

{

BusinessID = "16b7e171-6e3d-45a7-90d0-925d6907275d"

};

var result = this.\_imageArchiveService.Invoke(i => i.GetExamDocumentList(param));

return result;

//return null;

//var customer = this.CurrentCustomerGuid;

//return "customer" + customer + "Public OK OK OK1！";

}

catch (Exception e)

{

return $@"wcf:{wcf}\r\nerror:{e}";

}

}

返回结果：

{

"data": [

{

"fileuid": "94440b6d-1a75-4bc3-b36b-9bb235313241",

"businessid": "16b7e171-6e3d-45a7-90d0-925d6907275d",

"mimetype": "image/jpeg",

"businesstype": "Exam",

"uniqueid": null,

"classcode": "Exam",

"typecode": "ExamResult",

"formatcode": null,

"filesha": "dd2e8ca104a4ce60f29ac9841465c0eaecaacc01",

"filesize": 1165368,

"filecreatetime": "2018-12-18T09:25:52",

"originalfilename": "b0f33923-10e4-4b3a-849f-19c1eb1c4212.jpg",

"filedownloadurl": "http://192.168.1.202:9200/DocumentService/Download/Document/?fileUid=94440b6d-1a75-4bc3-b36b-9bb235313241&token=d44b4ee73edc072798df8f3cfe65578f",

"dicomdownloadlist": null,

"fileviewurl": "http://192.168.1.202:9200/DocumentService/Viewer/?businessID=16b7e171-6e3d-45a7-90d0-925d6907275d&businessType=Exam&classCode=Exam&typeCode=ExamResult&fileUid=94440b6d-1a75-4bc3-b36b-9bb235313241&organizationID=47000326-5&print=&download=&businessView=false&token=635c910899c69b7ae23612780630f9bd&reportimgpost=&Anonymous=",

"filebusinessviewurl": "http://192.168.1.202:9200/DocumentService/Viewer/?businessID=16b7e171-6e3d-45a7-90d0-925d6907275d&businessType=Exam&classCode=Exam&typeCode=ExamResult&fileUid=94440b6d-1a75-4bc3-b36b-9bb235313241&organizationID=47000326-5&print=&download=&businessView=true&token=635c910899c69b7ae23612780630f9bd&reportimgpost=&Anonymous=",

"eworldviewerurl": null,

"filepath": null,

"filerelativepath": null,

"modality": null

},

{

"fileuid": "7f4d657d-3126-479f-b510-7f5aa88f33c5",

"businessid": "16b7e171-6e3d-45a7-90d0-925d6907275d",

"mimetype": "application/dicom",

"businesstype": "Exam",

"uniqueid": null,

"classcode": "Exam",

"typecode": "ExamImage",

"formatcode": "DICOMDIR",

"filesha": "089138ae459ec17085984c1c908867fc663a4fcb",

"filesize": 7902304,

"filecreatetime": "2018-12-18T09:25:53",

"originalfilename": "75ae6a62-cc8a-43c5-b2f8-944030112160.zip",

"filedownloadurl": "http://192.168.1.202:9200/DocumentService/Download/Document/?fileUid=7f4d657d-3126-479f-b510-7f5aa88f33c5&token=9e35d2f8a9d86016da669a957f3263d6",

"dicomdownloadlist": null,

"fileviewurl": "http://192.168.1.202:9200/DocumentService/Viewer/?businessID=16b7e171-6e3d-45a7-90d0-925d6907275d&businessType=Exam&classCode=Exam&typeCode=ExamImage&fileUid=7f4d657d-3126-479f-b510-7f5aa88f33c5&organizationID=47000326-5&print=&download=&businessView=false&token=9836f40e0224e9b06e4667b57016c58f&reportimgpost=&Anonymous=",

"filebusinessviewurl": "http://192.168.1.202:9200/DocumentService/Viewer/?businessID=16b7e171-6e3d-45a7-90d0-925d6907275d&businessType=Exam&classCode=Exam&typeCode=ExamImage&fileUid=7f4d657d-3126-479f-b510-7f5aa88f33c5&organizationID=47000326-5&print=&download=&businessView=true&token=9836f40e0224e9b06e4667b57016c58f&reportimgpost=&Anonymous=",

"eworldviewerurl": null,

"filepath": null,

"filerelativepath": null,

"modality": null

}

],

"code": 0,

"msg": "OK"

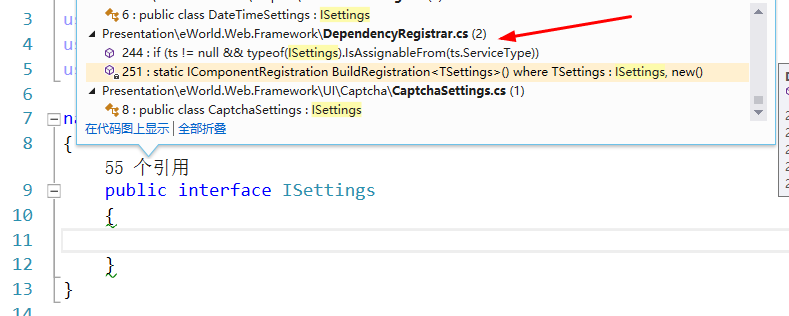
}

居然直接把地址拼接好了，不用我再做data的转换？

（10）

6.

配置表的批量注入：



不知道是不是就是这个：

public IEnumerable<IComponentRegistration> RegistrationsFor(

Service service,

Func<Service, IEnumerable<IComponentRegistration>> registrations)

{

var ts = service as TypedService;

if (ts != null && typeof(ISettings).IsAssignableFrom(ts.ServiceType))

{

var buildMethod = BuildMethod.MakeGenericMethod(ts.ServiceType);

yield return (IComponentRegistration)buildMethod.Invoke(null, null);

}

}

static IComponentRegistration BuildRegistration<TSettings>() where TSettings : ISettings, new()

{

return RegistrationBuilder

.ForDelegate((c, p) =>

{

var currentWebSiteId = c.Resolve<IWebSiteContext>().CurrentWebSite.Id;

//uncomment the code below if you want load settings per WebSite only when you have two WebSites installed.

//var currentWebSiteId = c.Resolve<IWebSiteService>().GetAllWebSites().Count > 1

// c.Resolve<IWebSiteContext>().CurrentWebSite.Id : 0;

//although it's better to connect to your database and execute the following SQL:

//DELETE FROM [Setting] WHERE [WebSiteId] > 0

return c.Resolve<ISettingService>().LoadSetting<TSettings>(currentWebSiteId);

})

.InstancePerHttpRequest()

.CreateRegistration();

}

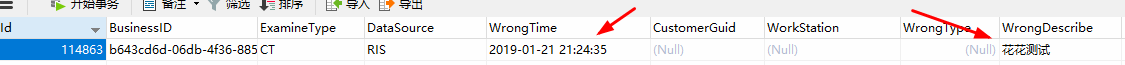
7.

所有的log暂时注释

// \_log.Debug("Endpoint地址为代理模式，将转换为：" + endpoint);

8.

测试SaveLogError方法成功。



9.

把参数验证移植到云归档里面，而不是放在能力开放平台。

if (triggerUID == Guid.Empty)

{

result.code = -1;

result.msg = "TriggerUID不能为空";

\_log.Info("查询任务触发的数据配置列表失败，错误信息：TriggerUID不能为空");

return result;

}

10.

{"DicomInfo":{"OrganizationID":null,"DICOMIP":"192.168.1.9","DICOMPort":109,"DICOMAE":"IA9","DICOMCMovePort":0},"StudyInstanceUIDList":["1.2.826.0.1.3680043.2.1505.97676.20160101093257.0"],"PatientID":"GY92260","AccessionNumber":"SD-53259","Anonymize":false}

1.2.826.0.1.3680043.2.1505.97676.20160101093257.0

{"RequestFileList":[],"ReportFileList":[{"DownloadUrl":"http://192.168.2.47:8787/DocumentService/Download/Document?FileUID=140fac5e-6ba2-4c2a-85dd-c57659c38148&token=2b5b070f839b35329a8b1f505a309b5c","TempFileName":"140fac5e-6ba2-4c2a-85dd-c57659c38148.Jpg","ImageFormatCode":null,"FilmFormatCode":null}],"ImageFileList":[],"FilmFileList":[],"BusinessTime":"2018/11/24 9:32:57"}

修改解决方案

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "eWorld.Entity", "..\..\..\Data\trunk\entity\eWorld.Entity\eWorld.Entity.csproj", "{BD066F4B-E1AB-423E-BC3D-B32688CE6EF9}"

11.

预留，job控制器中的注册和获取检查列表，两个接口没做，入参实体变了，而且nuget上没发新包。

12.

13.

14