接口文档和逻辑整理

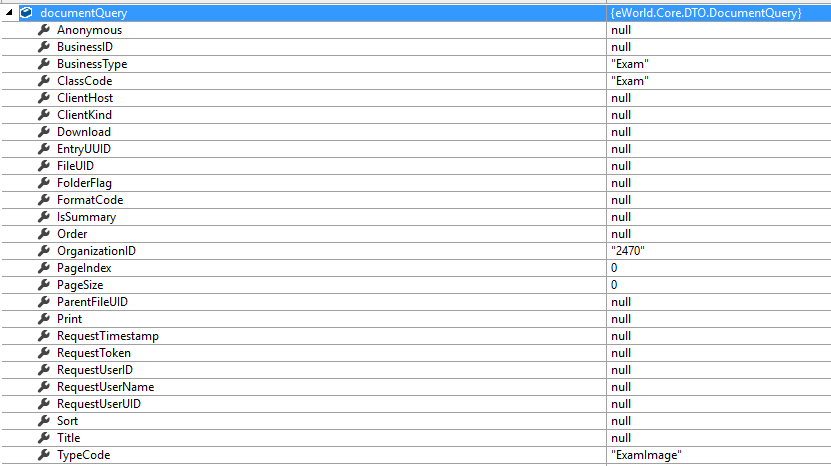
1.

获取影像浏览地址

http://ip:port/v2/public/get\_exam\_imageurl

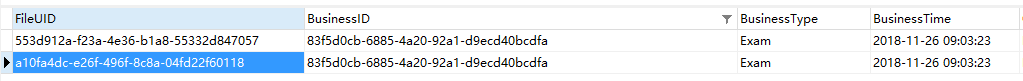
进入到组织查询-文档的方法：

var docmentInfo = QueryDocument(docmentQuery).FirstOrDefault();



最终是按上传时间排序

query = query.OrderBy(d => d.UploadTime);





但前面是取第一个，

var docmentInfo = QueryDocument(docmentQuery).FirstOrDefault();

而且传递参数的时候，

var docmentQuery = new DocumentQuery

{

BusinessID = query.ObservationUID,

OrganizationID = query.OrganizationID,

BusinessType = businessType,

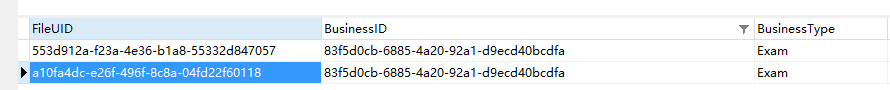
ClassCode = classCode,

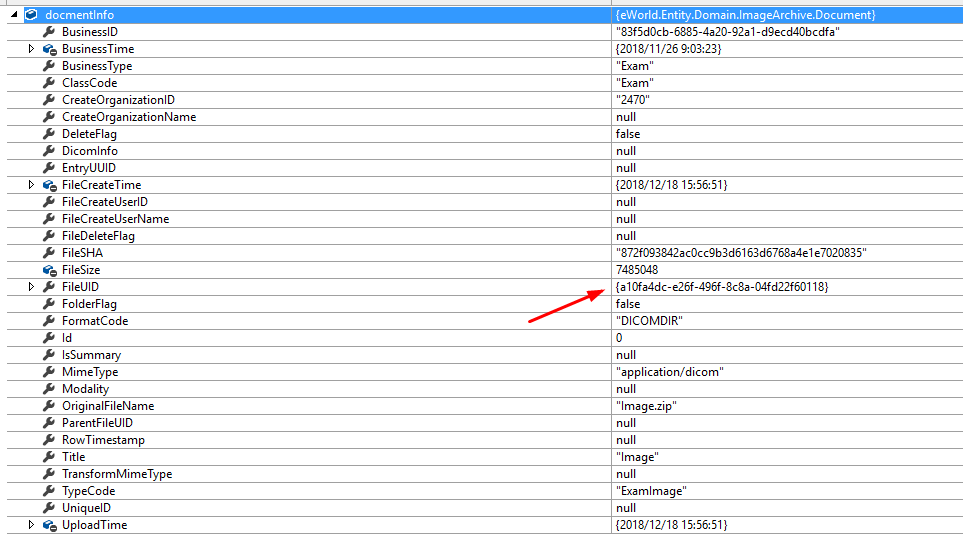
TypeCode = typeCode

};

这里没有传递很多的，可以精确到某条记录的查询参数，所以如果数据多条的话，如果只每次传ObservationUID，因为每条数据对应的都是该ObservationUID，每次拿到的都是最新的那条数据，所以数据是错的。

ObservationUID对应了两条数据：





感觉这里的FirstOrDefault()可以改成Tolist，获取所有的。

var docmentInfo = QueryDocument(docmentQuery).FirstOrDefault();

if (docmentInfo == null)

{

modelResult.msg = "检查未获取到影像";

modelResult.code = ResultCodeEnum.Fail;

return modelResult;

}

var url = endpoint.TrimEnd('?').TrimEnd('/') + "/Viewer/?" + "businessID=" + query.ObservationUID +

"&businessType=" + businessType +

"&classCode=" + classCode +

"&typeCode=" + typeCode +

"&formatCode=" + docmentInfo.FormatCode +

"&entryUUID=" + docmentInfo.EntryUUID +

"&uniqueID=" + docmentInfo.UniqueID +

"&title=" + docmentInfo.Title +

"&organizationID=" + query.OrganizationID +

"&token=" + CommonUtil.CreateWCFToken(docmentInfo.FileUID.ToString());

最后在Web api层对url地址进行重组：

result.data = CommonUtil.TransLocalhostForClient(resultdata.Result, ConfigSettings.GetIDCASSettings().Url.TrimEnd('/'));

result.code = ReturnInfoEnum.Success;

过来的时候是localhost，在这里去获取配置文件中的地址，也就是当前架设的web页面的地址（9200，Document，icc，idcas这些在一起的地址），也就是访问当前影像的地址。

到时候我要配置在能力开放平台的，因为最后部署，我，还有运维是知道所部署的医院，对应的web地址是多少的。

接口调用：

入参：

{

"AccessionNumber":"SD-53243",

"ServiceSectID":"DX",

"OrganizationID":"2470",

"ObservationUID":"83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa"

}

返回结果：

{

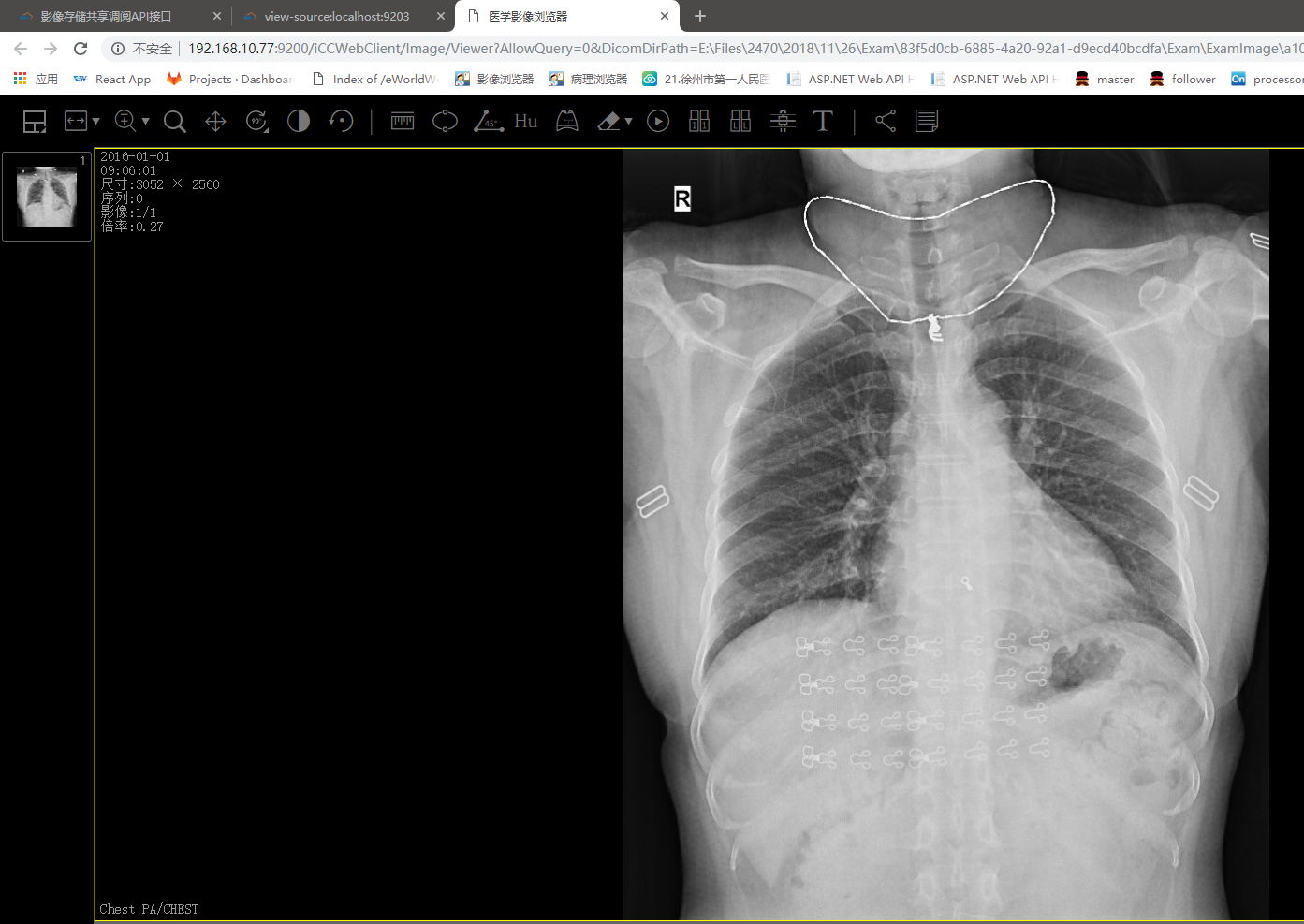
"code": 0,

"msg": "成功",

"data": "http://192.168.10.77:9200/DocumentService/Viewer/?businessID=83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa&businessType=Exam&classCode=Exam&typeCode=ExamImage&formatCode=DICOMDIR&entryUUID=&uniqueID=&title=Image&organizationID=2470&token=220880ee12461c846b0e0cba3774bddb"

}

最终可以浏览的影像：



尝试，将Document的获取换成了集合，发现，其实最终拿到的也是一个数据：

var docmentInfos = QueryDocument(docmentQuery).ToList();



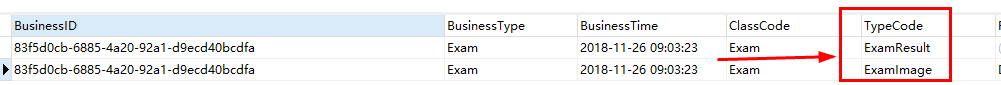
继续查看，发现TypeCode是不一样，因而猜想，难道是和ObservationResult那种，同一个ObservationUID，对应的是多条数据，通过其中一个字段去区分是哪个结果。

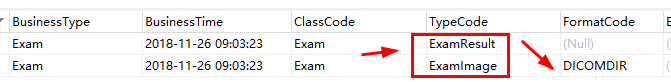
/// <summary>

/// 文档小类代码。详见DocumentService设计文档。

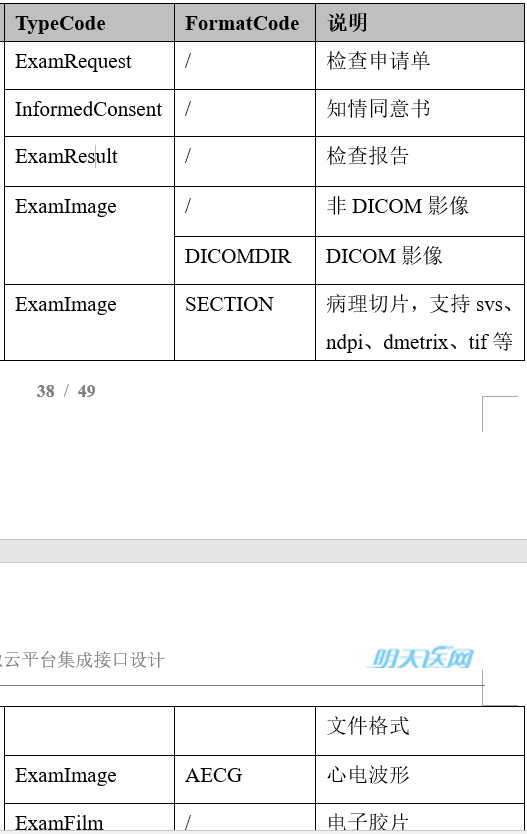
/// </summary>

public string TypeCode { get; set; }





原来一个是检查报告，一个是影像



所以这里方法的目的是通过各种参数确定了唯一性的那个检查结果：

var docmentQuery = new DocumentQuery

{

BusinessID = query.ObservationUID,

OrganizationID = query.OrganizationID,

BusinessType = businessType,

ClassCode = classCode,

TypeCode = typeCode

};

var docmentInfo = QueryDocument(docmentQuery).FirstOrDefault();

因为当前是 获取单次检查影像文件浏览地址 ，所以必然是选中了和影像相关的参数去查询。

根据不同需求，也有看到ToList()的：

var documentList = QueryDocument(query).ToList();

2.

获取检查文档信息

WebApi层，根据是否启用短链接，跳入不同的两个wcf方法。

if (ConfigDefine.UseShortDocumentViewerUrl)

{

resultdata = \_imageArchiveClient.Invoke(t => t.DocumentList(query.ToQuery()));

}

else

{

resultdata = \_imageArchiveClient.Invoke(t => t.GetExamDocumentList(query.ToQuery()));

}

进入了第二个方法：

获取了内网地址

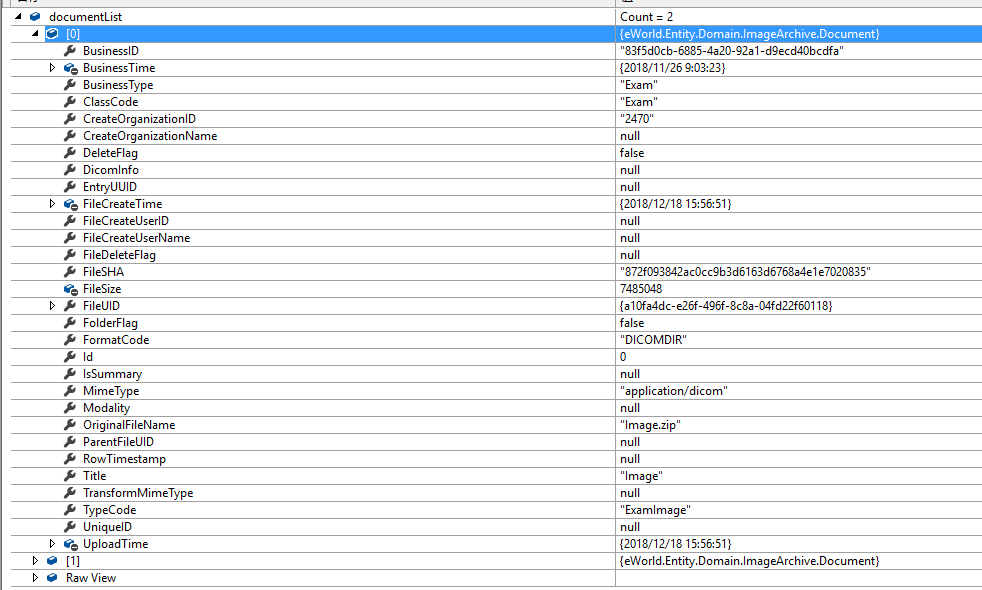


endpoint = endpoint.TrimEnd('?').TrimEnd('/');

这里就是ToList()，获取所有满足条件的配置了：

var documentList = QueryDocument(query).ToList();

和上面第一当中同一个ObservationUID，也就是BusinessID，对应Document中有两组数据。



将Doucument表中的每条数据进行拼接，获取想要的结果集，进行返回：

if (documentList.Count > 0)

{

List<DocumentViewDTO> resultList = null;

#region 数据处理

resultList = documentList.Select(d => new DocumentViewDTO

{

FileUID = d.FileUID,

MimeType = d.MimeType,

BusinessID = d.BusinessID,

BusinessType = d.BusinessType,

ClassCode = d.ClassCode,

TypeCode = d.TypeCode,

FormatCode = d.FormatCode,

FileSHA = d.FileSHA,

FileSize = d.FileSize,

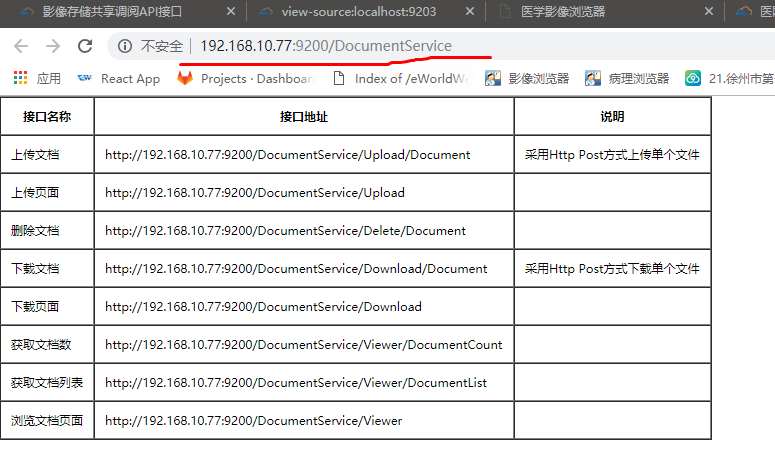
UniqueID = d.UniqueID,

FileCreateTime = d.FileCreateTime,

OriginalFileName = d.OriginalFileName,

这些通过endpoint去拼接的，都是拼凑出张丹那边，DocumentService的Get访问地址。

通过地址（和云归档配在一个Web端口，9200）去一览：



FileDownloadUrl = endpoint + "/Download/Document/?fileUid=" + d.FileUID +

"&token=" + CommonUtil.CreateWCFToken(d.FileUID.ToString()),

FileViewUrl = endpoint + "/Viewer/?businessID=" + d.BusinessID +

"&businessType=" + d.BusinessType + "&classCode=" + d.ClassCode +

"&typeCode=" + d.TypeCode + "&fileUid=" + d.FileUID +

"&organizationID=" + (string.IsNullOrWhiteSpace(query.OrganizationID) ? d.CreateOrganizationID : query.OrganizationID) +

"&print=" + query.Print + "&download=" + query.Download + "&businessView=false&token=" +

CommonUtil.CreateWCFToken(d.BusinessID + d.BusinessType + d.ClassCode + d.TypeCode +

d.FileUID + "00") + "&reportimgpost=" + query.reportImgPost + "&Anonymous=" + query.Anonymous,

FileBusinessViewUrl = endpoint + "/Viewer/?businessID=" + d.BusinessID +

"&businessType=" + d.BusinessType + "&classCode=" + d.ClassCode +

"&typeCode=" + d.TypeCode + "&fileUid=" + d.FileUID +

"&organizationID=" + (string.IsNullOrWhiteSpace(query.OrganizationID) ? d.CreateOrganizationID : query.OrganizationID) +

"&print=" + query.Print + "&download=" + query.Download +

"&businessView=true&token=" +

CommonUtil.CreateWCFToken(d.BusinessID + d.BusinessType + d.ClassCode +

d.TypeCode + d.FileUID + "00") + "&reportimgpost=" + query.reportImgPost + "&Anonymous=" + query.Anonymous

}).ToList();

这上面就是对表里的数据进行拼接，获取最终想要的结果

#endregion

#region withDicomDownload或者withDicomdirPath

if (query.WithDicomDownload || query.WithDicomdirPath)

{

var webclient = new WebClientEx();

webclient.Headers.Add("Content-Type", "application/x-www-form-urlencoded");

foreach (var d in resultList)

{

if (d.FormatCode == "DICOMDIR")

{

if (!string.IsNullOrWhiteSpace(endpoint))

{

try

{

var getDICOMDownloadListUrl = endpoint +

"/Viewer/GetDICOMDownloadList?fileUid=" + d.FileUID + "&token=" +

CommonUtil.CreateWCFToken(d.FileUID.ToString()) +

"&withDicomDownload=" + query.WithDicomDownload;

\_log.Info(string.Format("DICOMDownloadListUrl={0}", getDICOMDownloadListUrl));

var data = webclient.UploadData(getDICOMDownloadListUrl, "POST", new byte[0]);

var dateStr = HttpUtility.UrlDecode(Encoding.ASCII.GetString(data));

var dicomdirDownload = JsonHelper.DeserializeFromJsonString<DicomdirDownloadInfo>(dateStr);

d.DICOMDownloadList = dicomdirDownload.DcmList;

}

catch (Exception e)

{

\_log.Error("生成dicom文件下载地址失败，源dicom文件不存在", e);

continue;

}

}

}

}

}

#endregion

#region WithEWorldViewerUrl

if (query.WithEWorldViewerUrl == true)

{

var dicomServiceConfig = \_serviceConfigService.GetServiceConfigList(new ServiceConfigQuery()

{

ServiceType = ServiceTypeEnum.DICOM,

CurrentFlag = true

}).ResultList.FirstOrDefault();

if (dicomServiceConfig == null)

{

\_log.Warn("没有查询到默认的DICOM Service Config。");

}

else

{

foreach (var item in resultList)

{

if (item.TypeCode == "ExamImage")

{

var dicomStudy = \_dicomStudyRepository.Table.Where(d => d.FileUID == item.FileUID).ToList();

var studyInstanceUID = "";

var accessionNumber = "";

var patientID = "";

var modality = "";

foreach (var ds in dicomStudy)

{

studyInstanceUID += ds.StudyInstanceUID + ";";

accessionNumber = ds.AccessionNumber;

patientID = ds.PatientID;

modality = ds.ModalityInStudy;

}

studyInstanceUID = studyInstanceUID.TrimEnd(';');

var serviceUrl = (dicomServiceConfig.UseTLS == true ? dicomServiceConfig.EndpointTLS : dicomServiceConfig.Endpoint).Split(':');

var ip = serviceUrl[0];

var port = serviceUrl.Length > 1 ? serviceUrl[1] : "80";

item.EWorldViewerUrl = string.Format("eworldview:FileSrc=Network,IP={0},Port={1},AE={2},StudyInstanceUID={3},PatientID={4},AccessionNumber={5},Modality={6}",

ip,

port,

dicomServiceConfig.ServiceID,

studyInstanceUID,

accessionNumber,

patientID,

modality

);

}

}

}

}

上面的参数，默认只有这两个是True

/// <summary>

/// 文档记录查询

/// </summary>

public class DocumentQuery:BaseQuery

{

public DocumentQuery()

{

WithDicomDownload = true;

WithDicomdirPath = true;

}

/// <summary>

/// 是否返回DICOM原始文件地址

/// </summary>

public bool WithDicomDownload { get; set; }

/// <summary>

/// 是否返回DICOMDIR地址

/// </summary>

public bool WithDicomdirPath { get; set; }

其余的，默认都是false，也可以通过参数去传递，是否要返回结果：

/// <summary>

/// 是否返回DICOMDIR地址

/// </summary>

public bool WithDicomdirPath { get; set; }

/// <summary>

/// 是否返回EWorldViewer地址

/// </summary>

public bool WithEWorldViewerUrl { get; set; }

/// <summary>

/// 是否返回文件存放路径

/// </summary>

public bool WithFilePath { get; set; }

/// <summary>

/// 是否返回文件存放相对路径

/// </summary>

public bool withFileRelativePath { get; set; }

最后在Web Api层面进行数据的重组，因为地址都是内网，要改成当前的网址。看配置在哪家医院，就是哪家医院的DocumentService的web地址（9200）。

这里面的优化，就是先把地址全部提取出来。

list.ForEach(p =>

{

p.FileBusinessViewUrl = CommonUtil.TransLocalhostForClient(p.FileBusinessViewUrl, ConfigDefine.IDCASUrl.TrimEnd('/'));

p.FileDownloadUrl = CommonUtil.TransLocalhostForClient(p.FileDownloadUrl, ConfigDefine.IDCASUrl.TrimEnd('/'));

p.FileViewUrl = CommonUtil.TransLocalhostForClient(p.FileViewUrl, ConfigDefine.IDCASUrl.TrimEnd('/'));

if (p.DICOMDownloadList != null && p.DICOMDownloadList.Count > 0)

{

var newdicomdownloadlist = new List<string>();

foreach (var item in p.DICOMDownloadList)

{

newdicomdownloadlist.Add(CommonUtil.TransLocalhostForClient(item, ConfigDefine.IDCASUrl.TrimEnd('/')));

}

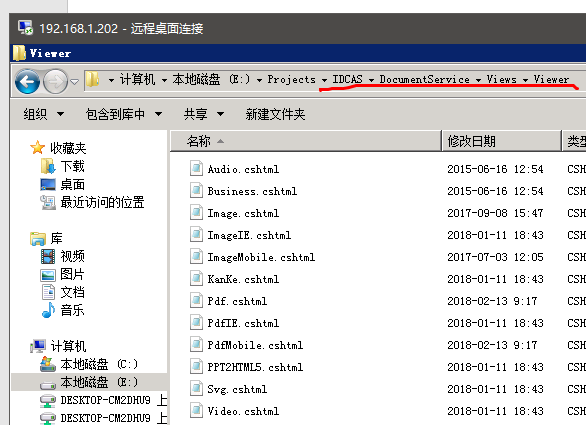
p.DICOMDownloadList = newdicomdownloadlist;

}

});

如果对上面的几个bool类型的入参进行赋值，就会对结果集的空字段（跟Dicom有关了）进行地址拼接，再赋值。

是在DocumentService文件夹下，Viewer控制器中。



foreach (var d in resultList)

{

if (d.FormatCode == "DICOMDIR")

{

if (!string.IsNullOrWhiteSpace(endpoint))

{

try

{

var getDICOMDownloadListUrl = endpoint +

"/Viewer/GetDICOMDownloadList?fileUid=" + d.FileUID + "&token=" +

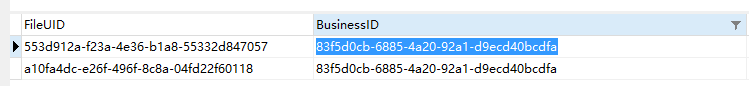
CommonUtil.CreateWCFToken(d.FileUID.ToString()) +

"&withDicomDownload=" + query.WithDicomDownload;

\_log.Info(string.Format("DICOMDownloadListUrl={0}", getDICOMDownloadListUrl));

获取地址

<http://localhost/DocumentService/Viewer/GetDICOMDownloadList?fileUid=a10fa4dc-e26f-496f-8c8a-04fd22f60118&token=220880ee12461c846b0e0cba3774bddb&withDicomDownload=True>



var data = webclient.UploadData(getDICOMDownloadListUrl, "POST", new byte[0]);

var dateStr = HttpUtility.UrlDecode(Encoding.ASCII.GetString(data));

var dicomdirDownload = JsonHelper.DeserializeFromJsonString<DicomdirDownloadInfo>(dateStr);

d.DICOMDownloadList = dicomdirDownload.DcmList;

}

结果报错了！因为我本地没有架设DocumentView的服务（web页面）



catch (Exception e)

{

\_log.Error("生成dicom文件下载地址失败，源dicom文件不存在", e);

continue;

}

}

}

}

但如果是77上面，因为配置了DocumentService，就有结果：

{

"BusinessID": "83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa",

"WithDicomDownload": true,

"WithDicomdirPath": true,

"WithEWorldViewerUrl": true,

"WithFilePath": true

}

返回结果：

{

"code": 0,

"msg": "成功",

"data": {

"DataList": [

{

"FileUID": "a10fa4dc-e26f-496f-8c8a-04fd22f60118",

"BusinessID": "83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa",

"BusinessType": "Exam",

"BusinessTime": null,

"UniqueID": null,

"ClassCode": "Exam",

"TypeCode": "ExamImage",

"FormatCode": "DICOMDIR",

"MimeType": "application/dicom",

"FileSHA": "872f093842ac0cc9b3d6163d6768a4e1e7020835",

"FileSize": 7485048,

"Title": null,

"UploadTime": null,

"FileCreateTime": "2018-12-18T15:56:51",

"OriginalFileName": "Image.zip",

"TransformMimeType": null,

"DicomInfo": null,

"FileDownloadUrl": "http://192.168.10.77:9200/DocumentService/Download/Document/?fileUid=a10fa4dc-e26f-496f-8c8a-04fd22f60118&token=220880ee12461c846b0e0cba3774bddb",

"FileViewUrl": "http://192.168.10.77:9200/DocumentService/Viewer/?businessID=83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa&businessType=Exam&classCode=Exam&typeCode=ExamImage&fileUid=a10fa4dc-e26f-496f-8c8a-04fd22f60118&organizationID=2470&print=&download=&businessView=false&token=54bb0b0081a3754056b768a4d553cdde&reportimgpost=&Anonymous=",

"FileBusinessViewUrl": "http://192.168.10.77:9200/DocumentService/Viewer/?businessID=83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa&businessType=Exam&classCode=Exam&typeCode=ExamImage&fileUid=a10fa4dc-e26f-496f-8c8a-04fd22f60118&organizationID=2470&print=&download=&businessView=true&token=54bb0b0081a3754056b768a4d553cdde&reportimgpost=&Anonymous=",

"EWorldViewerUrl": "eworldview:FileSrc=Network,IP=192.168.2.35,Port=1004,AE=PACS,StudyInstanceUID=1.2.826.0.1.3680043.2.1505.97648.20160101090323.0,PatientID=SD-53243,AccessionNumber=GY92233,Modality=DX",

"DICOMDownloadList": null,

"FilePath": "E:\\Files\\2470\\2018\\11\\26\\Exam\\83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa\\Exam\\ExamImage\\a10fa4dc-e26f-496f-8c8a-04fd22f60118\\DICOMDIR",

"FileRelativePath": null,

"Modality": null

},

{

"FileUID": "553d912a-f23a-4e36-b1a8-55332d847057",

"BusinessID": "83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa",

"BusinessType": "Exam",

"BusinessTime": null,

"UniqueID": null,

"ClassCode": "Exam",

"TypeCode": "ExamResult",

"FormatCode": null,

"MimeType": "image/jpeg",

"FileSHA": "b9c8800036b839bac6340c83414bc7c1af681f78",

"FileSize": 861273,

"Title": null,

"UploadTime": null,

"FileCreateTime": "2019-01-02T13:57:14",

"OriginalFileName": null,

"TransformMimeType": null,

"DicomInfo": null,

"FileDownloadUrl": "http://192.168.10.77:9200/DocumentService/Download/Document/?fileUid=553d912a-f23a-4e36-b1a8-55332d847057&token=e588c0764fe15f9cea0dad0fd09e6191",

"FileViewUrl": "http://192.168.10.77:9200/DocumentService/Viewer/?businessID=83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa&businessType=Exam&classCode=Exam&typeCode=ExamResult&fileUid=553d912a-f23a-4e36-b1a8-55332d847057&organizationID=2470&print=&download=&businessView=false&token=947bce27e28680348da04bb6940493a2&reportimgpost=&Anonymous=",

"FileBusinessViewUrl": "http://192.168.10.77:9200/DocumentService/Viewer/?businessID=83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa&businessType=Exam&classCode=Exam&typeCode=ExamResult&fileUid=553d912a-f23a-4e36-b1a8-55332d847057&organizationID=2470&print=&download=&businessView=true&token=947bce27e28680348da04bb6940493a2&reportimgpost=&Anonymous=",

"EWorldViewerUrl": null,

"DICOMDownloadList": null,

"FilePath": "http://idea.test/oos-nm2.ctyunapi.cn/ZBQ/2470/2018/11/26/Exam/83f5d0cb-6885-4a20-92a1-d9ecd40bcdfa/Exam/ExamResult/553d912a-f23a-4e36-b1a8-55332d847057.Jpg",

"FileRelativePath": null,

"Modality": null

}

],

"PageInfo": {

"PageIndex": 1,

"PageSize": 10,

"TotalCount": 2,

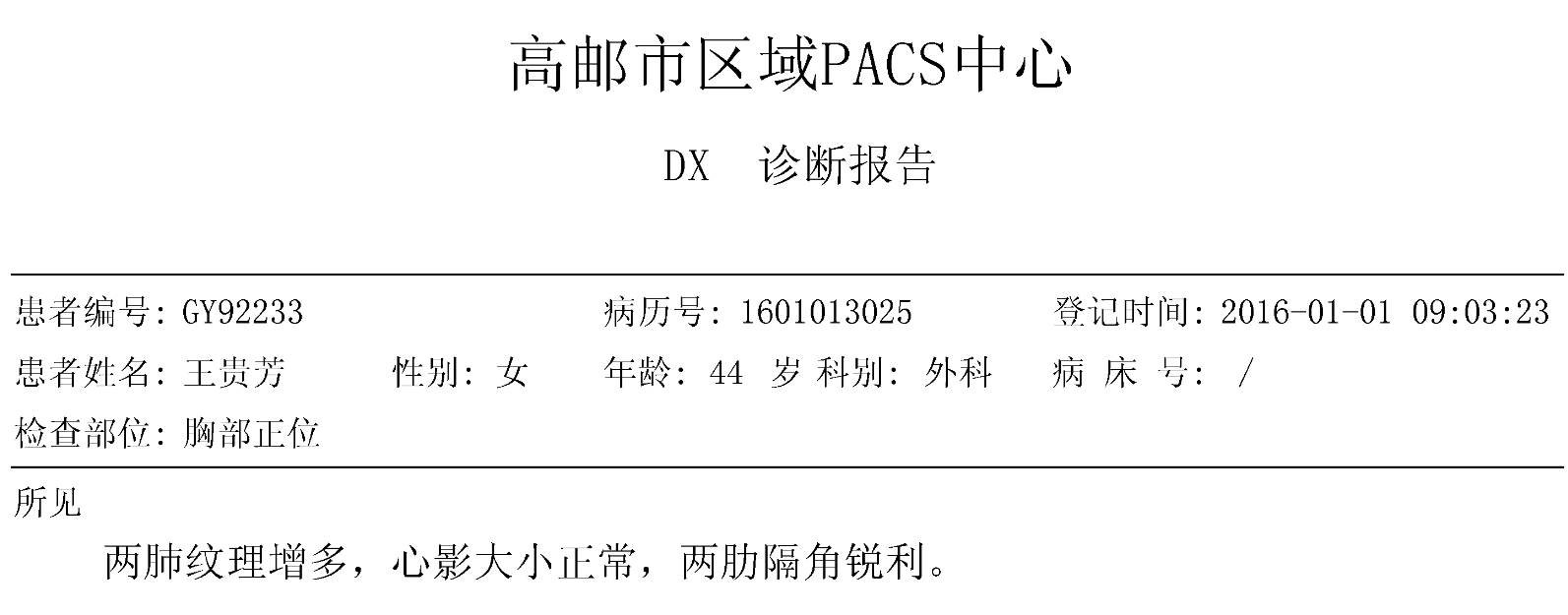
"TotalPages": 1

}

}

}

返回的地址打开，就是类似这种报告：



3.

4.

5.