云归档动工

**数据采集**

1.

注册患者信息

四种入参：患者信息，就诊信息，检查信息，检查结果信息（对应四张表）

因为Visit（病人就诊记录）表中有PatientIndex中的PatientMasterID（病人主索引ID），而ObservationRequest表中有Visit表中的ObservationUID（观察唯一序号），而ObservationResult（观察结果表）中又有ObservationRequest表中的ObservationUID（关联观察请求序列号），所以这四张表的存在关系，就是1234如下排列。

（1）PatientIndex：病人主索引信息

（2）Visit：病人就诊记录

（3）ObservationRequest：观察请求记录

（4）ObservationResult：观察结果表

第五种为最后插入数据的表。

（5）LogClinicDataOperation临床数据操作日志

namespace eWorld.Core.DTO

{

/// <summary>

/// 注册检查信息请求参数

/// </summary>

public class DataArchivingQuery

{

/// <summary>

/// 患者信息

/// </summary>

public PatientRegisterQuery Patient { get; set; }

/// <summary>

/// 就诊信息

/// </summary>

public PatientVisitQuery Visit { get; set; }

/// <summary>

/// 检查信息

/// </summary>

public ObservationRequestAddQuery ObservationRequest { get; set; }

/// <summary>

/// 检查结果信息

/// </summary>

public List<ObservationResultInfo> ObservationResultList { get; set; }

}

}

最终注册的时候，只调用了 PatientIndex：病人主索引信息 和 Visit：病人就诊记录

/// <summary>

/// 患者注册

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

private ModelResult<string> PatientRegister(PatientRegisterQuery query, PatientVisitQuery visitquery)

{

这一串的注册，感觉应该是同生共死，用事务。

一连串，还是对四个表的插入工作。但是第一个，PatientIndex：病人主索引信息表，涉及了insert或者update的操作。

而且，这里涉及后一个方法的参数用到前一个刚生成的参数。

var patientregister = PatientRegister(query.Patient, query.Visit);

if (patientregister.code == ResultCodeEnum.Success)//患者信息注册成功方可写入检查信息

{

\_log.Info("患者信息注册成功");

string patientmasterid = patientregister.Result;

query.ObservationRequest.PatientMasterID = patientmasterid;

var uploadobservationrequest = UploadObservationRequest(query.ObservationRequest);

if (uploadobservationrequest.code == ResultCodeEnum.Success)

{

\_log.Info("患者检查信息注册成功");

query.Visit.PatientMasterID = patientmasterid;

query.Visit.VisitUID = uploadobservationrequest.Result.VisitUID.Value;

var uploadpatientvisit = UploadPatientVisit(query.Visit);

if (uploadpatientvisit.code == ResultCodeEnum.Success)

{

\_log.Info("患者就诊信息注册成功" );

}

else

{

result.msg = "检查信息上传失败，患者就诊信息注册失败";

\_log.Info("检查信息上传失败,患者就诊信息注册失败，：" + uploadpatientvisit.msg);

}

observationuid = uploadobservationrequest.Result.ObservationUID.Value;

foreach (var observationresult in query.ObservationResultList)

{

observationresult.ObservationUID = observationuid;

}

var uploadobservationresult = UploadObservationResult(query.ObservationResultList);

if (uploadobservationresult.code == ResultCodeEnum.Success)

{

\_log.Info("患者纠缠结果信息注册成功");

try

{

var log = new LogClinicDataOperation();

log.BusinessType = "Exam";

log.BusinessUID = observationuid;

log.ClassCode = "Exam";

log.ClientApplication = "";

log.ClientHostName = "";

log.ClientKind = 1;

log.CreateDate = DateTime.Now;

log.OperationDesc = "采集检查信息入库";

log.OperationType = 0;//新增

log.TypeCode = "Exam";

log.UserKind = 2;

\_logClinicDataOperationRepository.Insert(log);

}

catch (Exception e)

{

\_log.Error("采集检查信息入库，保存采集日志日志出现异常", e);

}

result.code = ResultCodeEnum.Success;

result.msg = "OK";

result.Result = observationuid;

}

else

{

//检查结果信息入库失败，删除检查信息、就诊信息

\_log.Info("采集检查信息入库,检查结果信息保存失败，：" + uploadobservationresult.msg);

result.msg = "检查信息上传失败，检查结果信息保存失败";

}

}

else

{

\_log.Info("采集检查信息入库,检查信息保存失败，：" + uploadobservationrequest.msg);

result.msg = "检查信息上传失败，检查信息保存失败";

}

}

else

{

result.msg = "检查信息上传失败，患者信息注册失败";

\_log.Info("检查信息上传失败,患者信息注册失败，：" + patientregister.msg);

}

最后还有第五个方法，采集检查信息入库。

涉及LogClinicDataOperation临床数据操作日志 表

\_log.Info("患者检查结果信息注册成功");

try

{

var log = new LogClinicDataOperation();

log.BusinessType = "Exam";

log.BusinessUID = observationuid;

log.ClassCode = "Exam";

log.ClientApplication = "";

log.ClientHostName = "";

log.ClientKind = 1;

log.CreateDate = DateTime.Now;

log.OperationDesc = "采集检查信息入库";

log.OperationType = 0;//新增

log.TypeCode = "Exam";

log.UserKind = 2;

\_logClinicDataOperationRepository.Insert(log);

}

数据采集接口：

var returnModel = \_imageArchiveClient.Invoke(i => i.SaveExamDataInfo(query));

第三方用的：

var saveRes = \_imageArchiveClient.Invoke(t => t.SavePatientInfo(param));

用数据采集的，现在是把就诊信息也放在了里面，我要把它隔离出来。

现在是五张表，

拆分之后：

（1）操作主索引表

（2）操作就诊信息，检查信息，检查结果信息，临床数据操作日志暂时没提。

对应的，曹兵那边也会拆成两张表。

改造后，需要第一步完成，才能去操作第二步。

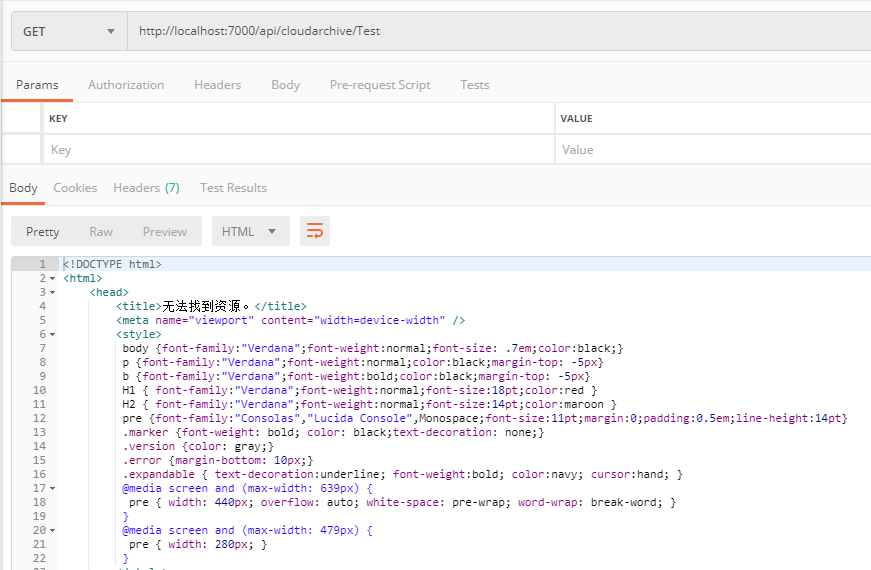
这两个接口是给采集服务用的，我要做的，患者信息注册和数据采集的这个接口，是有两个接口，一个是检查数据的，一个是患者信息的。只要走这两个接口，就可以把检查信息给传上来了。

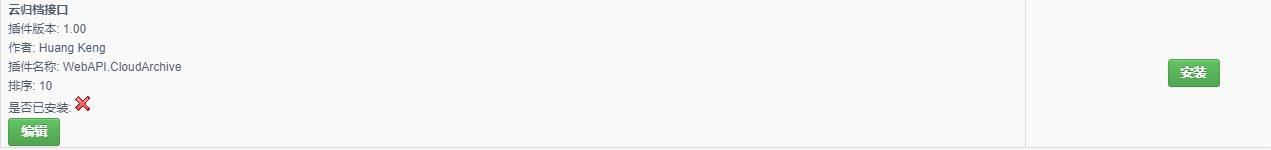
以前的采集服务的接口，也得重新整一遍，包括心电的，很老的，没优化过的，给喻永红用的，都要重新整。

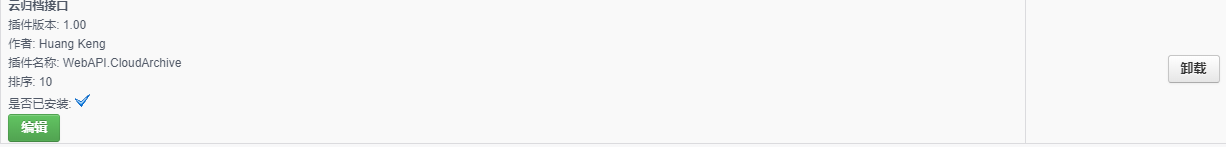
我需要先搞清楚流程。

2.

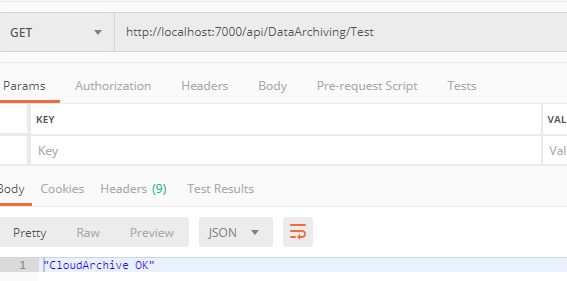
未安装插件之前







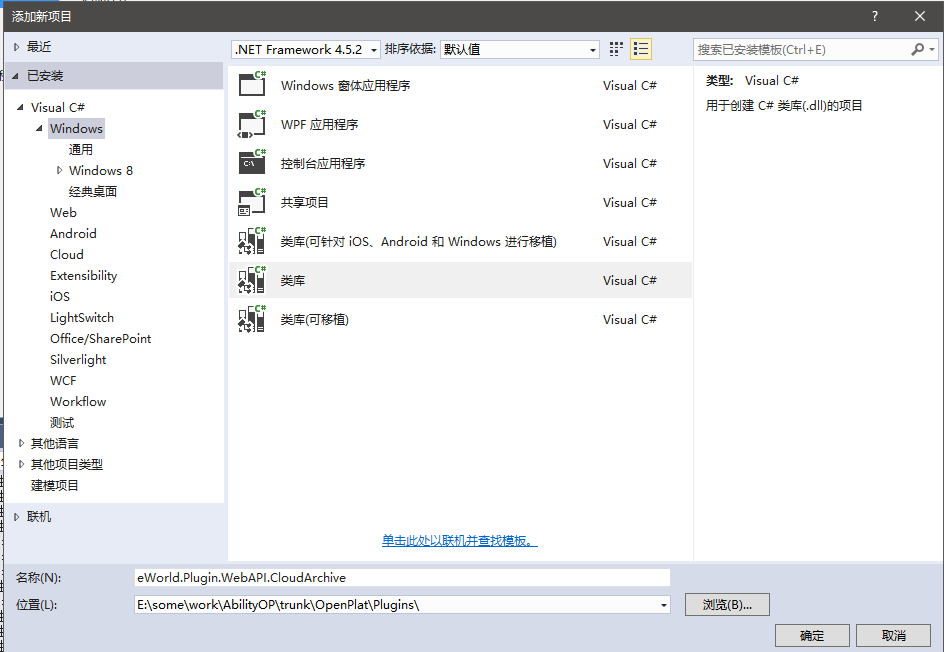
安装之后ok了。



3.

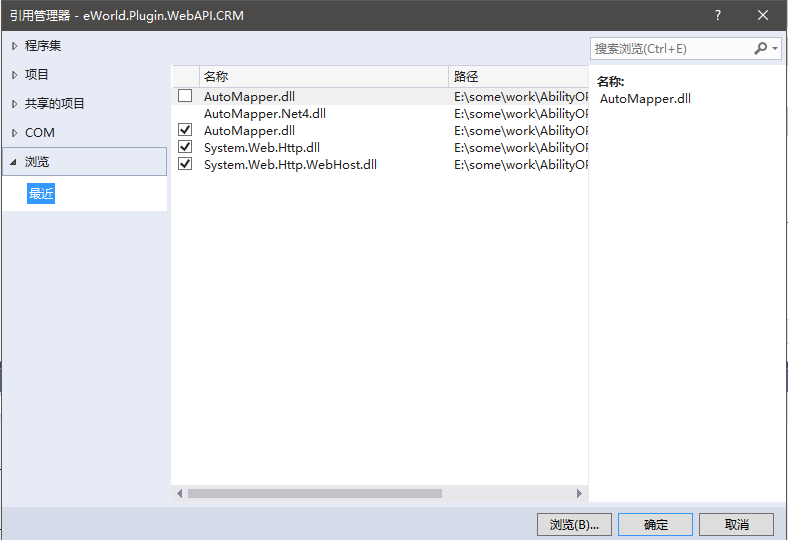
再走一次

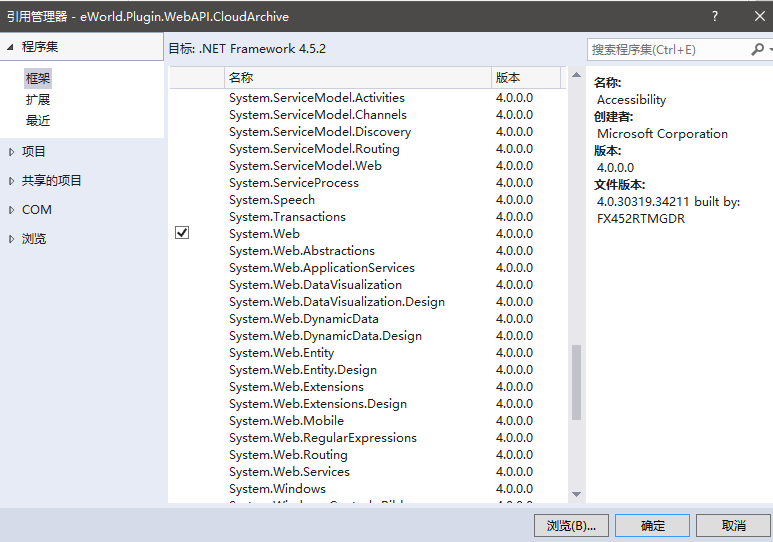
（1）

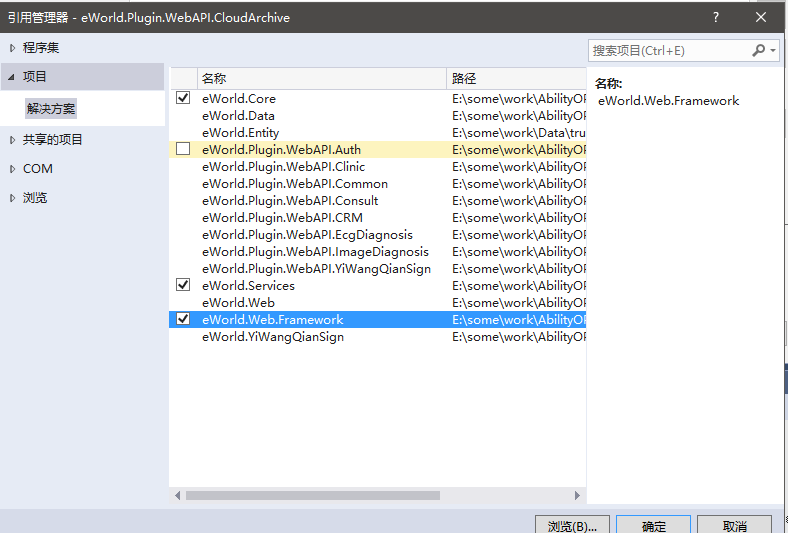


（2）

添加引用

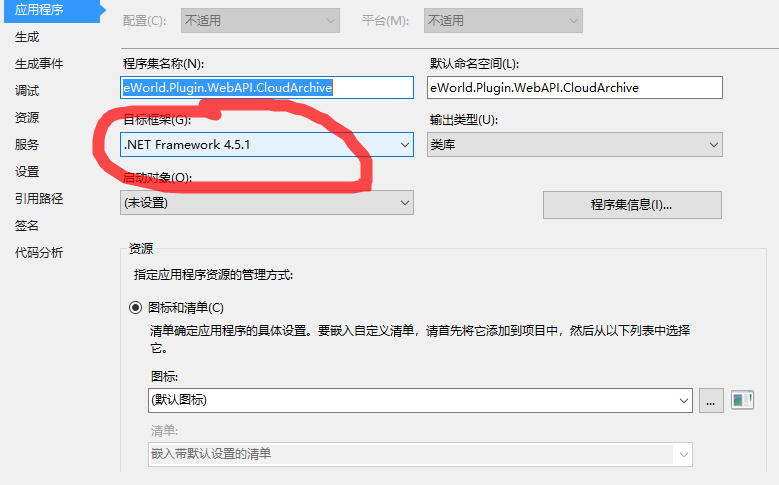




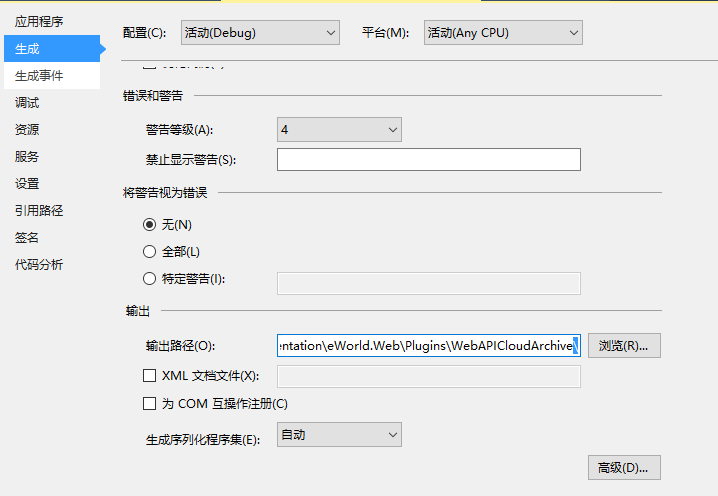


（3）

4.5.1



生成：



（4）

4.

上传文件：

var saveExamRes = \_imageArchiveClient.Invoke(t => t.UploadDocument(info));

第三方接口和Foreign控制器

有一个字节流的参数

/// <summary>

/// 文件的字节流。不加入Token验证。

/// </summary>

public byte[] FileSize { get; set; }

5.

获取患者检查dicomstudy信息

/// <summary>

/// 获取患者检查dicomstudy信息

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

[Route("v2/dataarchiving/get\_exam\_dicomstudy")]

[HttpPost]

public ModelReturnResult<PatientExamDicomStudyDTO> GetExamDicomStudy([FromBody]PatientExamDicomStudyQuery query)

{

WCF：

var returnModel = \_imageArchiveClient.Invoke(i => i.GetExamDicomStudy(query.ToQuery()));

核心代码：

先是在Document（文档记录）中查询该BusinessID对应的结果，如果存在，再去DicomStudy（影像的检查信息）表中根据FileUID关联查询到所有的病人信息。

var documentInfo = \_repository.Table.Where(d => d.BusinessID == query.BusinessID && d.TypeCode == "ExamImage" && d.DeleteFlag == false).FirstOrDefault();

if (documentInfo != null)

{

var dicomStudy = \_dicomStudyRepository.Table.Where(d => d.FileUID == documentInfo.FileUID).FirstOrDefault();

if (dicomStudy != null)

{

model.AccessionNumber = dicomStudy.AccessionNumber;

model.ModalityInStudy = dicomStudy.ModalityInStudy;

model.PatientID = dicomStudy.PatientID;

model.PatientName = dicomStudy.PatientName;

model.StudyDateTime = dicomStudy.StudyDateTime;

model.StudyID = dicomStudy.StudyID;

model.StudyInstanceUID = dicomStudy.StudyInstanceUID;

}

}

Guid businessid = new Guid();

if (Guid.TryParse(query.BusinessID, out businessid))

{

此外，在上面已经走了两个表之后，还要去ObservationRequest（观察请求记录）中查询该ObservationUID（也是和businessid关联）获取该观察结果，然后根据其中的PatientID，PatientMasterID，PIDAssigningAuthority，三个条件又一次关联Document（文档记录）表，去获取病人信息，进行组装。

var exam = \_observationRequestRepository.Table.Where(i => i.ObservationUID == businessid && i.DeleteFlag == false).FirstOrDefault();

这里，一旦ObservationRequest（观察请求记录）中没有数据，则直接跳到下面的“获取检查信息失败”的结果。

if (exam != null)

{

model.OrganizationID = exam.OrganizationID;

model.OrganizationName = exam.OrganizationName;

var patientInfo = \_repository.GetTable<PatientIndex>().Where(p => p.PatientID == exam.PatientID && p.PatientMasterID == exam.PatientMasterID && p.PIDAssigningAuthority == exam.PIDAssigningAuthority).FirstOrDefault();

if (patientInfo != null)

{

model.Name = patientInfo.Name;

}

modelResult.code = ResultCodeEnum.Success;

modelResult.msg = "OK";

modelResult.Result = model;

}

else

{

modelResult.msg = "根据业务唯一号获取检查信息失败";

}

}

入参：

{

"BusinessID":"00007b20-70b7-40f0-9913-13880fffafcd"

}

返回：

{

"code": 0,

"msg": "成功",

"data": {

"PatientName": null,

"Name": "赵治学",

"PatientID": null,

"StudyDateTime": null,

"AccessionNumber": null,

"ModalityInStudy": null,

"StudyID": null,

"StudyInstanceUID": null,

"OrganizationID": "123308244720006231",

"OrganizationName": "音坑乡卫生院"

}

}

6.

更新检查影像状态

var returnModel = \_imageArchiveClient.Invoke(i => i.ModifyExamImageState(query.ToQuery()));

针对 观察请求记录：ObservationRequest 表的操作。

var obs = \_observationRequestRepository.Table.Where(t => t.ObservationUID == query.ObservationUID).FirstOrDefault();

if (obs == null)

{

throw new Exception("查找不到ObservationRequest记录，ObservationUID=" + query.ObservationUID.ToString());

}

double diskSpace = 0;

var observationUID = query.ObservationUID.ToString();

var documentList = \_documentRepository.Table.Where(t => t.BusinessID == observationUID && t.DeleteFlag == false).ToList();

if (documentList != null && documentList.Count > 0)

{

foreach (var document in documentList)

{

if (document.FileSize != null)

{

diskSpace += document.FileSize.Value;

}

}

}

if (diskSpace > 0)

{

这里计算消耗磁盘空间：DiskSpace（消耗磁盘空间（Mb））

obs.DiskSpace = IntHelpert.Round(diskSpace, 1048576.00);

}

obs.IsSummary = false;

obs.HasImage = query.HasImage;

obs.ImageLocation = (byte)query.ImageLoaction;

obs.LastUpdateDate = DateTime.Now;

\_observationRequestRepository.Update(obs);

Document（文档记录）表的字段 BusinessID（业务唯一号） 和 ObservationRequest（观察请求记录）的ObservationUID（观察唯一序号）进行关联，相等去查询。

从Document表中获取该业务号对应的文件大小总和，修改到ObservationRequest查询出来的记录，然后update ObservationRequest的当前entity。

总结：就是对ObservationRequest表的update操作。

还是参数的校验，bll逻辑的各种判定，然后进行 crud 操作。

7.更新检查胶片数量

/// <summary>

/// 更新检查胶片数量

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

[Route("v2/dataarchiving/update\_exam\_filmcount")]

[HttpPost]

public BaseReturnResult UpdateExamFilmCount([FromBody]ModifyExamFimlCountQuery query)

WCF：

var returnModel = \_imageArchiveClient.Invoke(i => i.ModifyExamFilmCount(query.ToQuery()));

依旧是对于ObservationRequest（观察请求记录）的upadate，先查ObservationRequest（观察请求记录）表的当前匹配条件的entity，然后查出Document（文档记录）表中的磁盘消耗，最终都update，体现到ObservationRequest（观察请求记录）表的记录中去。

var obs = \_observationRequestRepository.Table.Where(t => t.ObservationUID == query.ObservationUID).FirstOrDefault();

if (obs == null)

{

throw new Exception("查找不到ObservationRequest记录，ObservationUID=" + query.ObservationUID.ToString());

}

double diskSpace = 0;

var observationUID = query.ObservationUID.ToString();

var documentList = \_documentRepository.Table.Where(t => t.BusinessID == observationUID && t.DeleteFlag == false).ToList();

if (documentList != null && documentList.Count > 0)

{

foreach (var document in documentList)

{

if (document.FileSize != null)

{

diskSpace += document.FileSize.Value;

}

}

}

if (diskSpace > 0)

{

磁盘消耗每次操作都要计算啊。就是每次update操作顺便查，顺便记录进去。

obs.DiskSpace = IntHelpert.Round(diskSpace, 1048576.00);

}

obs.IsSummary = false;

obs.FilmCount = query.FilmCount;

obs.LastUpdateDate = DateTime.Now;

\_observationRequestRepository.Update(obs);

总结：这两张表也想着连起来查询，但一对多，还是单一，然后查询第二种获取所有记录，合适的。

测试是否运行到了这个表。



8.

9.

10.

11.

12.

一些：

##### 患者信息注册：输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **描述** | **必选** | **值域说明** |
| -PatientID | 病人号(DES加密) | 必选 |  |
| -PIDAssigningAuthority | 病人号分配机构 | 必选 | 例如：17302.UIS 机构ID+“.”+检查科室类型 |
| -PatientMasterID | 病人主索引 | 可选 | 若是已经实现主索引，需要提供主索引值 |
| -Name | 病人姓名(DES加密) | 必选 |  |
| -NameSpell | 病人姓名拼音(DES加密) | 可选 |  |
| -MotherName | 母亲姓名(DES加密) | 可选 |  |
| -Sex | 性别(DES加密) | 必选 | 男、女、未知、未说明 |
| -BirthDate | 出生日期 | 必选 | yyyy-MM-dd HH:mm:ss |
| -BirthPlace | 出生地 | 可选 |  |
| -Nation | 民族 | 可选 |  |
| -Citizenship | 国籍 | 可选 |  |
| -MaritalStatus | 婚姻状况 | 可选 | 未婚、已婚、丧偶、离婚 |
| -IDCardNo | 身份证号(DES加密) | 必选 |  |
| -HealthCardNO | 健康卡号(DES加密) | 可选 |  |
| -ContactPhoneNO | 联系电话(DES加密) | 必选 |  |
| -HomePhoneNO | 家庭电话(DES加密) | 可选 |  |
| -BusinessPhoneNO | 工作电话(DES加密) | 可选 |  |
| -Email | 电子邮箱地址 | 可选 |  |
| -AddressProvince | 地址-省份 | 可选 |  |
| -AddressCity | 地址-城市 | 可选 |  |
| -AddressDistrict | 地址-区县 | 可选 |  |
| -AddressStreet | 地址-街道、乡镇 | 可选 |  |
| -AddressRoad | 地址-街、路、村 | 可选 |  |
| -AddressDetail | 地址-详细地址 | 可选 |  |
| -Postalcode | 邮政编码 | 可选 |  |
| -Occupation | 职业 | 可选 |  |
| -WorkUnit | 工作单位(DES加密) | 可选 |  |
| -InsuranceType | 医保类别 | 可选 |  |
| -InsuranceID | 医保号(DES加密) | 可选 |  |
| -OrganizationID | 组织机构ID(DES加密) | 必选 |  |
| **-**MedRecNO | 病历号(DES加密) | 可选 |  |
| -OutPatientNO | 门诊号(DES加密) | 可选 |  |
| **-**InPatientNO | 住院号(DES加密) | 可选 |  |
| **-**InHospitalFlag | 在院状态1在院其他不在院 | 可选 |  |

患者检查申请信息上传：

###### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **描述** | **必选** | **值域说明** |
| -ObservationUID | 检查业务唯一号 | 可选 |  |
| -PatientID | 病人号(DES加密) | 必选 |  |
| -PIDAssigningAuthority | PatientID分配机构 | 必选 |  |
| -PatientMasterID | 关联病人主索引 | 必选 |  |
| -PatientClass | 病人类型， | 必选 | 门诊、急诊、住院、体检 |
| -OrganizationID | 该就诊所发生的组织机构(DES加密) | 必选 |  |
| -OrganizationName | 组织机构名 | 必选 |  |
| -ClinicInfoType | 临床信息分类 | 可选 | Exam：检查  Lab:检验 |
| -PlacerOrderNO | 申请号 (DES加密) | 可选 |  |
| -PlacerAssigningAuthority | 申请号分配机构名 | 可选 |  |
| -PlacerOrderDetailNO | 申请单明细号，多个以,分隔 | 可选 |  |
| -FillerOrderNO | 检查分配的号码，如检查的检查号、检验的检验号。(DES加密) | 可选 |  |
| -FillerAssigningAuthority | FillerOrderNO号码的分配机构名 | 可选 |  |
| -FillerPatientID | 检查分配病人号 | 可选 |  |
| -AccessionNumber | 检查号，用于DICOM访问。(DES加密) | 必选 |  |
| -ServiceID | 观察服务ID，如果是通过医嘱而来的，则与医嘱代码一致，如检查为检查项目ID、检验为检验主题ID | 必选 |  |
| -ServiceCodeScheme | 观察项目编码体系 | 可选 |  |
| -ServiceText | 观察服务内容(DES加密) | 可选 |  |
| -ServiceSectID | 观察服务类型ID | 必选 |  |
| -ProcedureID | 观察过程ID，如检查的检查部位ID，检验的检验项目ID | 可选 |  |
| -ProcedureName | 观察过程名称，如检查的检查部位名称，检验的检验项目名称 | 必选 |  |
| -ProviderID | 观察申请者ID | 可选 |  |
| -ProviderName | 观察申请者姓名 | 可选 |  |
| -ProviderPhone | 观察申请者电话 | 可选 |  |
| -RequestDeptID | 观察申请科室ID | 可选 |  |
| -RequestDeptName | 观察申请科室名称 | 可选 |  |
| -RequestedDate | 观察请求时间，申请时间 | 可选 |  |
| -ResultOrganizationID | 出结果的机构ID | 可选 |  |
| -ResultServiceCenterUID | 出结果的服务中心UID | 可选 |  |
| -Reason | 申请观察目的 | 可选 |  |
| -Attention | 申请观察注意事项 | 可选 |  |
| -Symptom | 症状体征 | 可选 |  |
| -AdverseReaction | 过敏及不良反应 | 可选 |  |
| -ClinicDiagnosis | 临床诊断 | 可选 |  |
| -RelevantClinicalInfo | 其他相关的临床信息 | 可选 |  |
| -FastingFlag | 是否空腹 | 可选 | 0非空腹，1空腹 |
| -TransportationMode | 病人运输方式 | 可选 | CART:车或担架、PORT:床边、WALK:行走、WHLC:轮椅 |
| -ResultStatus | 检查结果状态名称 | 可选 | 附录1.5.2 |
| -ResultStatusCode | 检查结果状态代码 | 可选 | 附录1.5.2 |
| -RegTime | 登记时间 | 必选 |  |
| -RegisterID | 检查登记员ID | 可选 |  |
| -RegisterName | 检查登记员姓名 | 可选 |  |
| -ObservationDeptID | 检查执行科室ID | 可选 |  |
| -ObservationDeptName | 检查执行科室名称 | 可选 |  |
| -ObservationDate | 检查时间 | 必选 |  |
| -ObservationEndDate | 检查结束时间 | 可选 |  |
| -ObservationLocation | 检查发生的场所 | 可选 | 如检查房间 |
| -ObservationEquipmentID | 检查设备的ID | 可选 |  |
| ObservationEquipment | 检查设备名称 | 可选 |  |
| -ObservationMethod | 保存该观察的所有观察方法 | 可选 |  |
| -StudyInstanceUID | 检查实例UID | 可选 | 多个以,分割开。 |
| -TechnicianID | 检查技师ID | 可选 |  |
| -TechnicianName | 检查技师名称 | 可选 |  |
| -ResultPrincipalID | 结果主要负责人ID，如报告审核医生。同ObservationResult表中相应字段值 | 可选 |  |
| -ResultPrincipalName | 结果主要负责人姓名，如报告审核医生。同ObservationResult表中相应字段值 | 可选 |  |
| -ResultReviseID | 结果修订者ID | 可选 |  |
| -ResultReviseName | 结果修订者 | 可选 |  |
| -ResultAssistantID | 结果助理ID，如报告医生。同ObservationResult表中相应字段值 | 可选 |  |
| -ResultAssistantName | 结果助理姓名，如报告医生。同ObservationResult表中相应字段值 | 可选 |  |
| -ResultAssistantPhone | 报告医生电话 | 可选 |  |
| -PreliminaryDate | 初步结果时间，如检查的初步报告时间 | 可选 |  |
| -AuditDate | 初步结果时间，如检查的初步报告时间 | 可选 |  |
| -ReviseDate | 结果修订时间 | 可选 |  |
| -ResultDate | 最后的结果时间，报告状态时为报告时间，审核状态时为审核时间，修订时为修改时间。 | 可选 |  |
| -AbnormalFlags | 异常标志，如放射结果的阴阳性 | 可选 |  |
| -CriticalValue | 危急值 | 可选 |  |
| -ResultPrintCount | 是否已打印 | 可选 |  |
| -InfectionName | 传染病 | 可选 |  |
| -Charges | 检查费用，单位为元 | 可选 |  |
| -Payments | 病人实际支付费用，单位为元 | 可选 |  |
| -PaymentsFlag | 付款标志，0未付款(未收费)、1已付款(已收费) | 可选 |  |
| -FilmCount | 胶片数 | 可选 |  |
| -FilmNeed | 是否需要胶片，0不需要，1需要 | 可选 |  |
| -HasImage | 是否有影像 | 可选 |  |
| -ImageLocation | 影像位置，0:影像在医院内、1:影像在区域中心 | 可选 |  |
| -DataSource | 检查科室代码 | 可选 | 见附录1.5.3 |
| -DiskSpace | 消耗磁盘空间（Mb） | 可选 |  |
| -SourceOrganizationID | 检测机构ID | 可选 |  |
| -SourceOrganizationName | 检测机构名字 | 可选 |  |
| -AdmitDeptID | 住院入院科室ID，门诊就诊科室ID | 可选 |  |
| -AdmitDeptName | 住院入院科室名称，门诊就诊科室名称 | 可选 |  |
| -PointOfCare | 住院时为病区，非住院时时同就诊科室 | 可选 |  |
| -PointOfCareID | 住院时为病区ID，非住院时时同就诊科室ID | 可选 |  |
| -Room | 住院时为病房，非住院时为诊间 | 可选 |  |
| -Bed | 病床 | 可选 |  |
| -CurPatCondition | 当前病情状态。 | 可选 | 1危重 2急症 3疑难 |
| **-**MedRecNO | 病历号(DES加密) | 可选 |  |
| -OutPatientNO | 门诊号(DES加密) | 可选 |  |
| **-**InPatientNO | 住院号(DES加密) | 可选 |  |
| **-**CardType | 就诊卡类型 | 可选 |  |
| **-**CardNO | 就诊卡号(DES加密) | 可选 |  |
| **-**Age | 年龄 | 可选 |  |
| **-**AgeUnit | 年龄单位 | 可选 | 岁、月、周、天 |

数据采集第二版：

1.

保存病人信息WCF：

/// <summary>

/// 保存病人信息

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

public BaseResult SavePatientInfo(PatientIndexInfo patientIndexInfo)

{

return \_clinicManageService.SavePatientInfo(patientIndexInfo);

}

#endregion

值单独操作了病人主索引这一张表：

var item = \_patientIndexRepository.Table.FirstOrDefault(t => t.PatientID == patientIndexInfo.PatientID && t.PIDAssigningAuthority == patientIndexInfo.PIDAssigningAuthority);

if (item == null)

{

item = new PatientIndex();

mapExp = Mapper.CreateMap<PatientIndexInfo, PatientIndex>();

mapExp.ForMember(o => o.CreateDate, dto => dto.Ignore());

mapExp.ForMember(o => o.LastUpdateDate, dto => dto.Ignore());

item = Mapper.Map<PatientIndexInfo, PatientIndex>(patientIndexInfo, item);

item.CreateDate = DateTime.Now;

item.LastUpdateDate = DateTime.Now;

\_patientIndexRepository.Insert(item);

patientIndexInfo.CreateDate = item.CreateDate;

patientIndexInfo.LastUpdateDate = item.LastUpdateDate;

}

else

{

mapExp = Mapper.CreateMap<PatientIndexInfo, PatientIndex>();

mapExp.ForMember(o => o.CreateDate, dto => dto.Ignore());

mapExp.ForMember(o => o.LastUpdateDate, dto => dto.Ignore());

item = Mapper.Map<PatientIndexInfo, PatientIndex>(patientIndexInfo, item);

item.LastUpdateDate = DateTime.Now;

\_patientIndexRepository.Update(item);

patientIndexInfo.LastUpdateDate = item.LastUpdateDate;

}

患者注册走的WCF：

#region 患者检查归档（第三方接口）

/// <summary>

/// 患者注册

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

[OperationContract]

ModelResult<Guid> SaveExamDataInfo(DataArchivingQuery query);

\_patientManageService.GetPatientIndex(patient);

通过一个查询病人信息的方法进行插入数据，但是插入完成之后又查询了一次，来区别是否要插入还是update，不知道为什么

var item = \_patientIndexRepository.Table.FirstOrDefault(t => t.PatientID == patientIndexInfo.PatientID && t.PIDAssigningAuthority == patientIndexInfo.PIDAssigningAuthority);

if (item == null)

{

item = new PatientIndex();

mapExp = Mapper.CreateMap<PatientIndexInfo, PatientIndex>();

mapExp.ForMember(o => o.CreateDate, dto => dto.Ignore());

mapExp.ForMember(o => o.LastUpdateDate, dto => dto.Ignore());

item = Mapper.Map<PatientIndexInfo, PatientIndex>(patientIndexInfo, item);

item.CreateDate = DateTime.Now;

item.LastUpdateDate = DateTime.Now;

\_patientIndexRepository.Insert(item);

patientIndexInfo.CreateDate = item.CreateDate;

patientIndexInfo.LastUpdateDate = item.LastUpdateDate;

}

else

{

mapExp = Mapper.CreateMap<PatientIndexInfo, PatientIndex>();

mapExp.ForMember(o => o.CreateDate, dto => dto.Ignore());

mapExp.ForMember(o => o.LastUpdateDate, dto => dto.Ignore());

item = Mapper.Map<PatientIndexInfo, PatientIndex>(patientIndexInfo, item);

item.LastUpdateDate = DateTime.Now;

\_patientIndexRepository.Update(item);

patientIndexInfo.LastUpdateDate = item.LastUpdateDate;

}

拼接内容：

FillerAssigningAuthority（观察分配机构名）如果为空

就是OrganizationID+ DataSource+ ServiceSectID



//数据采集从影像中心采集由于该值不存在，获取检查记录时根据检查号匹配（检查号一致，检查号机构不一致）错误

if (string.IsNullOrWhiteSpace(query.ObservationRequest.FillerAssigningAuthority))

{

string assigningAuthority = query.ObservationRequest.OrganizationID;

if (!string.IsNullOrWhiteSpace(query.ObservationRequest.DataSource))

{

assigningAuthority += "." + query.ObservationRequest.DataSource;

}

if (!string.IsNullOrWhiteSpace(query.ObservationRequest.ServiceSectID))

{

assigningAuthority += "." + query.ObservationRequest.ServiceSectID;

}

query.ObservationRequest.FillerAssigningAuthority = assigningAuthority;

}

2.

3.

4.

5.

6.

检查数据共享：

1.

获取短链接参数

/// <summary>

/// 获取授权访问短链接参数

/// </summary>

/// <param name="query">条件</param>

/// <returns>查询结果</returns>

[Route("v2/public/get\_shorturl\_parameter")]

[HttpPost]

public ModelReturnResult<ShortUrlResult> GetShortUrlParameter([FromBody]GetShortUrlRequest query)

WCF服务：

var returnmodel = \_imageArchiveClient.Invoke(i => i.GetShortUrlParameter(query.ToQuery()));

需要去校验短链接是否过期

if (model.AccessMode.HasValue && model.AccessMode.Value == 2 && model.ExpireTime.Value < DateTime.Now)//时效性访问，判断是否过期

{

result.msg = "此短链已过期";

}

入参：

{

"accessId":"93183829734ccc4c"

}

返回：

{

"code": 0,

"msg": "OK",

"data": {

"Kind": 3,

"AccessMode": 0,

"PatientMasterID": "db2c153e-36f9-44df-bf68-cc679efa91e5",

"IDCardNO": null,

"ObservationUID": "3a8e1d8a-7dbf-4ff3-be59-d7cbe0db81ce",

"AccessionNumber": "618356",

"HospitalId": "2486",

"Modality": "CT",

"CustomParamContent": null

}

}

2.

获取检查列表

/// <summary>

/// 获取相关检查列表

/// </summary>

/// <param name="query">查询条件</param>

/// <returns></returns>

[Route("v2/public/get\_relatedexam\_list")]

[HttpPost]

public ModelReturnResult<GetRelatedExamListDTO> GetRelatedExamList([FromBody]GetRelatedExamQuery query)

入参：

/// <summary>

/// 相关检查列表请求参数

/// </summary>

public class GetRelatedExamQuery : BasePageQuery

{

/// <summary>

/// 用户唯一号

/// </summary>

public string PatientMasterID { get; set; }

/// <summary>

/// 检查号唯一号

/// </summary>

public string ObservationUID { get; set; }

}

参数：

2个：

用户唯一号，检查业务唯一号不能为空（query.ObservationUID) ，(query.PatientMasterID))

WCF：

var resultdata = \_imageArchiveClient.Invoke(t => t.GetRelatedExamList(relatedQuery));

接口通的，就是没整明白为什么这里要分页，这两个入参，难道会有很多的结果集？

病人+唯一的检查号，不是就表明是唯一的结果了？

入参：

{

"PatientMasterID":"8ab73c3f-0aa8-4e9c-95fa-3ae79d4522cd",

"ObservationUID":"28914756-9544-4d50-9ccf-8b766c3fb565"

}

结果：

{

"code": 0,

"msg": "ok",

"data": {

"DataList": [

{

"OrganizationName": "1E65408A9D9BCDFB94084E78D29F48416945CC3514768E55AF741E8891119D72C30E5EE78F38CE8F",

"OrganizationID": "557175DE00BFFB2B",

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

"Name": "95927258CF07D8329527472A9B9E2D5C",

"Sex": "CF067241960EED86",

"BirthDate": null,

"AccessionNumber": "C147E9671F0B9EBB5D879367D9B25847",

"ServiceSectID": "DX",

"PatientClass": "普通",

"ArchiveDate": "2018-12-18T10:01:09",

"FileSize": 7.9596719741821289,

"ProcedureName": "胸部正位",

"ResultDate": "2017-12-13T09:27:32",

"PatientID": "9E7BD662912232F1",

"PIDAssigningAuthority": "2470.RIS.DX",

"ServiceText": "普通摄片",

"InPatientNO": null,

"OutPatientNO": null,

"DataSource": "RIS",

"DataSourceName": "放射科",

"ObservationDate": "2018-11-26T09:03:23",

"ClinicInfoType": "Exam",

"PatientMasterID": "8ab73c3f-0aa8-4e9c-95fa-3ae79d4522cd",

"AuditDate": "2017-12-13T09:27:32",

"MedRecNO": "7FA22D5446120AD1CD708B9254A54C8E",

"VisitUID": "71ae1916-e4fa-448f-be30-8a82b4877e79",

"AbnormalFlags": "阳性",

"CriticalDesc": ""

}

],

"PageInfo": {

"PageIndex": 1,

"PageSize": 10,

"TotalCount": 1,

"TotalPages": 1

}

}

}

3.

获取检查详情

三个接口用到，wcf是同一个：

/// <summary>

/// 获取检查详情

/// </summary>

/// <returns></returns>

[Route("v2/public/get\_exam\_info")]

[HttpPost]

public ModelReturnResult<PatientExamDetailResultDTO> GetPatientExamInfo([FromBody]PatientExamDetailRequest query)

参数：

result.msg = "检查号，身份证号，检查业务唯一号不能同时为空";

必传：

query.hospitalId 机构解密

query.modality 检查类型

WCF：

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

4.

5.

6.

7.

8.

9.

**采集：**

1.

查询Job服务程序信息列表

/// <summary>

/// 根据条件查询Job服务程序列表信息

/// </summary>

/// <param name="query">查询条件</param>

/// <returns>查询结果</returns>

[Route("Job/GetJobServerList")]

[HttpPost]

public PageResult<JobServerInfo> GetJobServerList([FromBody]JobServerQuery query)

WCF服务：

var saveExamRes = \_imageArchiveClient.Invoke(t => t.GetJobServerList(query));

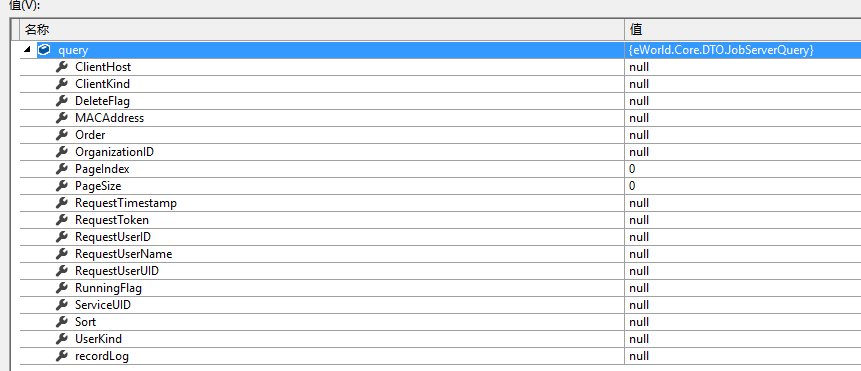
入参：

{

}

如果什么都不填，就报错。

一个大括号，这边检测的值就是：



结果：

{

"PageIndex": 1,

"TotalPage": 1,

"TotalCount": 7,

"ResultList": [

{

"ServiceUID": "300a0c7e-cccc-4451-9d15-c82c7ef092df",

"ServiceName": "杭州市西湖区中医院",

"MACAddress": "00:50:56:9E:96:20",

"IP": "192.168.2.75",

"HostName": "zhang-PC",

"RunningFlag": false,

"ControlCode": null,

"CreateDate": "2018-07-11T11:27:21",

"DeleteFlag": false,

"OrganizationID": "321084100000",

"UserName": "100008",

"Password": "3147CB05F050899E",

"Version": "1.0.3.2018063007\_beta",

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "48410be4-d898-4229-b092-abfdd98e37f3",

"ServiceName": "西安交通大学第一医院",

"MACAddress": "00:50:56:9E:96:20",

"IP": "192.168.10.77",

"HostName": "WIN-UD350RB0BNH",

"RunningFlag": false,

"ControlCode": null,

"CreateDate": "2018-02-28T11:03:00",

"DeleteFlag": false,

"OrganizationID": "17302",

"UserName": "100209",

"Password": "C7974E0248E6F71E46B07265A901A050",

"Version": null,

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "858298bd-a20e-49c4-942a-d7040e98dc18",

"ServiceName": "浙江大学医学院附属邵逸夫医院",

"MACAddress": null,

"IP": "192.168.10.77",

"HostName": null,

"RunningFlag": false,

"ControlCode": 0,

"CreateDate": "2018-12-20T15:52:24",

"DeleteFlag": false,

"OrganizationID": "17303",

"UserName": "100006",

"Password": null,

"Version": null,

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "ab67ddc8-a9d3-4a3b-a6d7-107d468264f1",

"ServiceName": "杭州市西湖区留下人民医院",

"MACAddress": "60:A4:4C:62:78:40",

"IP": "192.168.3.45",

"HostName": "WINDOWS-LD9T674",

"RunningFlag": true,

"ControlCode": null,

"CreateDate": "2017-11-02T14:45:46",

"DeleteFlag": false,

"OrganizationID": "2470",

"UserName": "100005",

"Password": "C7974E0248E6F71E46B07265A901A050",

"Version": "1.0.6.2018122001\_beta",

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "cbe9cdd4-20dd-41b2-97f4-e4d853b11e28",

"ServiceName": "嘉兴测试机构",

"MACAddress": "D0:17:C2:8C:49:D7",

"IP": "192.168.1.45",

"HostName": "WINDOWS-LD9T674",

"RunningFlag": false,

"ControlCode": 0,

"CreateDate": "2018-04-24T15:35:57",

"DeleteFlag": false,

"OrganizationID": "1",

"UserName": "100214",

"Password": "3147CB05F050899E",

"Version": null,

"IsNeedUpdate": false,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "cc66b42a-7e27-44ab-abe4-76cbae2c3b1e",

"ServiceName": "高邮市三垛中心卫生院",

"MACAddress": "1C:1B:0D:BD:B6:5F",

"IP": "192.168.3.45",

"HostName": "WINDOWS-LD9T674",

"RunningFlag": false,

"ControlCode": null,

"CreateDate": "2018-01-16T09:25:07",

"DeleteFlag": false,

"OrganizationID": "17030",

"UserName": "100052",

"Password": "3147CB05F050899E",

"Version": "1.0.6.2018122001\_beta",

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

},

{

"ServiceUID": "d1fd0d72-c8d0-4d61-a1c4-ea6d1f15fb85",

"ServiceName": null,

"MACAddress": "D0:17:C2:8C:49:D7",

"IP": "192.168.3.45",

"HostName": "WINDOWS-LD9T674",

"RunningFlag": false,

"ControlCode": null,

"CreateDate": "2018-11-28T12:01:10",

"DeleteFlag": false,

"OrganizationID": "123308244720006231",

"UserName": "100408",

"Password": "3147CB05F050899E",

"Version": "1.0.5.2018112801\_beta",

"IsNeedUpdate": null,

"RequestUserUID": null,

"RequestUserID": null,

"RequestUserName": null,

"RequestTimestamp": null,

"RequestToken": null,

"UserKind": null,

"ClientKind": null,

"recordLog": null,

"ClientHost": null

}

],

"code": 0,

"msg": "成功",

"info": null

}

单纯传页码和每页的数量，反而错误了。

{

"PageIndex":1,

"PageSize":7

}

{

"PageIndex": 0,

"TotalPage": 0,

"TotalCount": 0,

"ResultList": null,

"code": -1,

"msg": "必须指定排序字段!",

"info": null

}

这样三个参数就可以了。

{

"PageIndex":2,

"PageSize":3,

"Sort":"ServiceUID"

}

2.

3.

4.

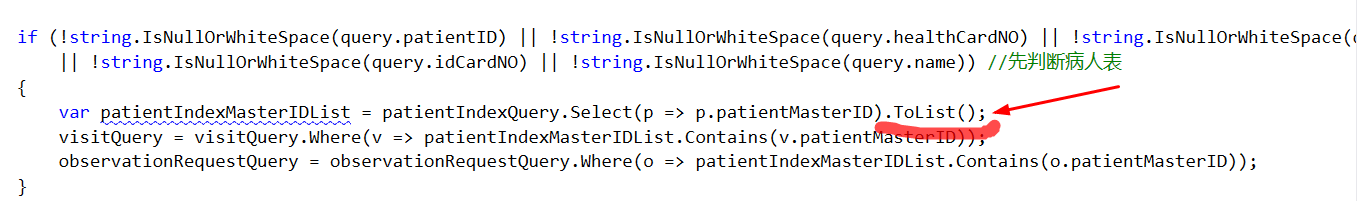
1.

没有dal层的，逻辑层和dal层完全混在一起，四百行代码的样子。

逻辑判断的同时对数据库进行操作：



一系列判断和查询之后，在这个方法将数据库断开连接，此时所有的数据加载到了内存中进行组装



多次连接，断开数据库，每次都是操作所有满足条件的数据。

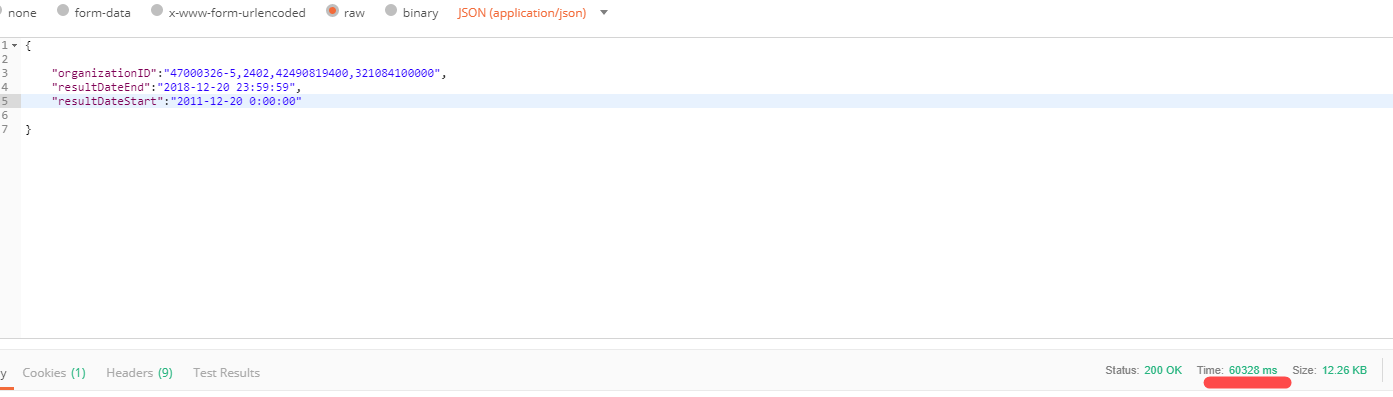


最后来的分页已经没有意义了，所有的符合条件的数据一开始就全部加载出来了

result = observationRequestQuery.Skip((query.PageIndex - 1) \* query.PageSize).Take(query.PageSize).ToList();

2.

优化中。



这样的时间，接口返回数据的时间炸了。

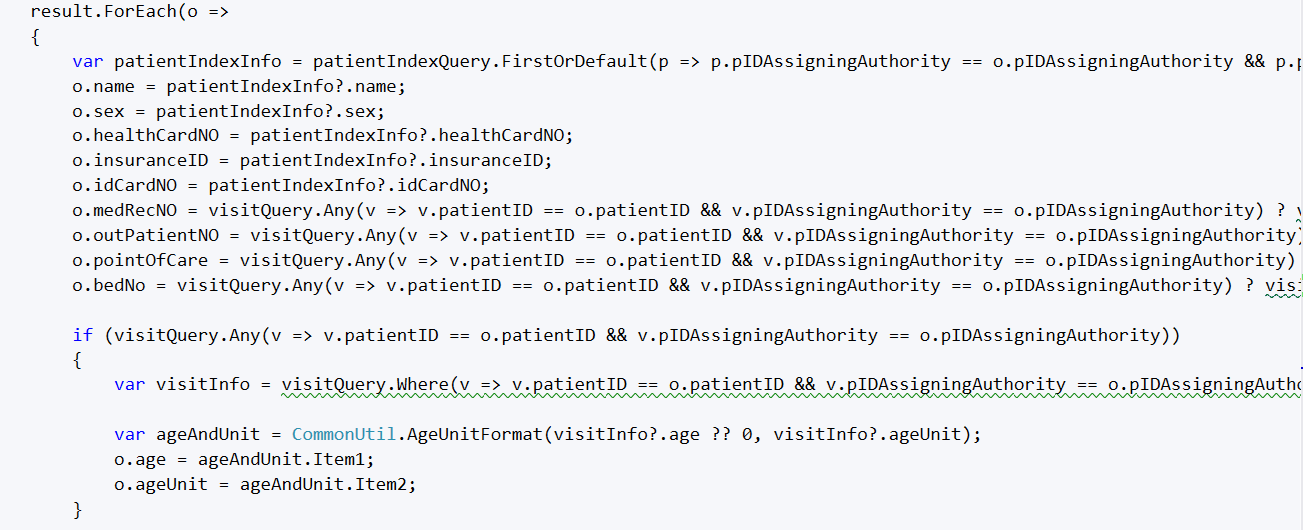
入参

"organizationID":"47000326-5,2402,42490819400,321084100000",

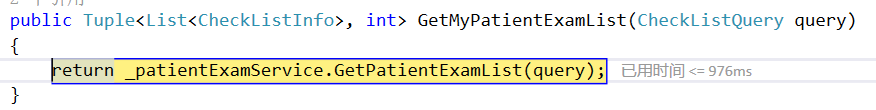
"resultDateEnd":"2018-12-20 23:59:59",

"resultDateStart":"2018-10-20 0:00:00"

基本诊断为分页后装配数据的地方，浪费了很多时间



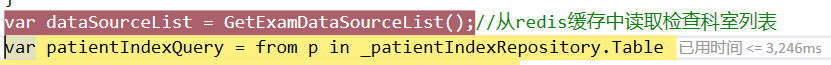
调用wcf：



或者



重新生成后，怎么从redis中读取这么慢。

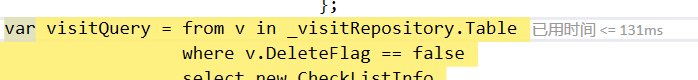


之后从redis中读取就很快了。

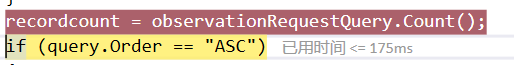


三个sql语句也是不少时间





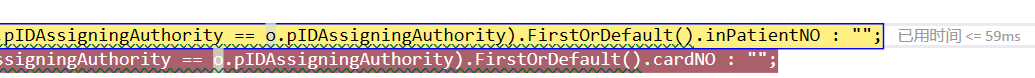
这个count，能去掉的话也很好



分页消耗：



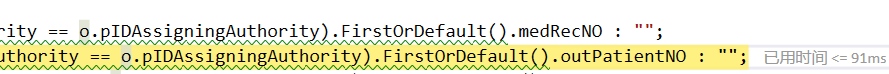
foreach里面，十次，那感觉也只需要，差不多2秒啊





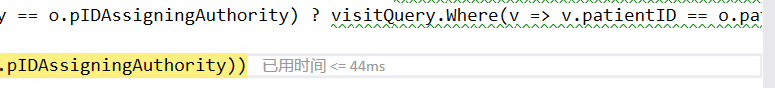
列举foreach里面大的一些个判定:

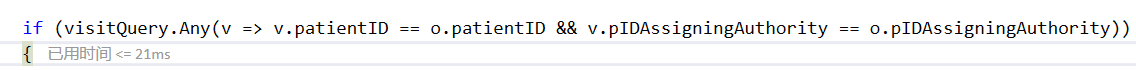
第一个21 ms

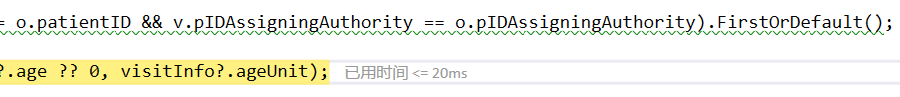




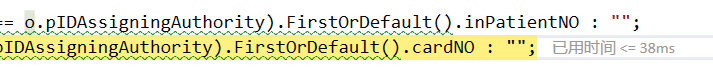


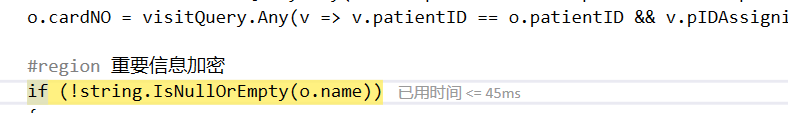






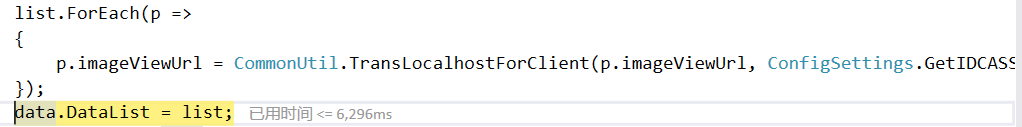




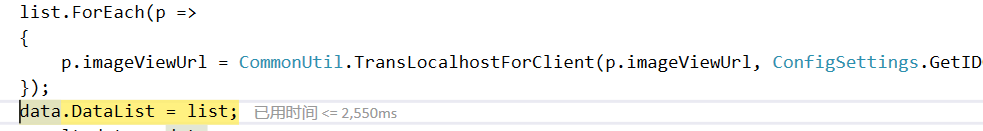


差不多370，倒也不是很慢啊，感觉。

WebApi层，这个又是怎么回事啊。



这玩意果然耗时间



总结下：

（1）wcf的调用，耗费在1秒

（2）count()方法的去除 200ms

（3）foreach中的优化，平均循环一次400ms的样子

（4）最后的DataList循环调用，三四千毫秒的样子

整体一个循环，1页，1条数据的情况，2秒不到一点，差不多吻合。



3.

测试了，和页码PageIndex无关，而跟单页数据量的多少有关

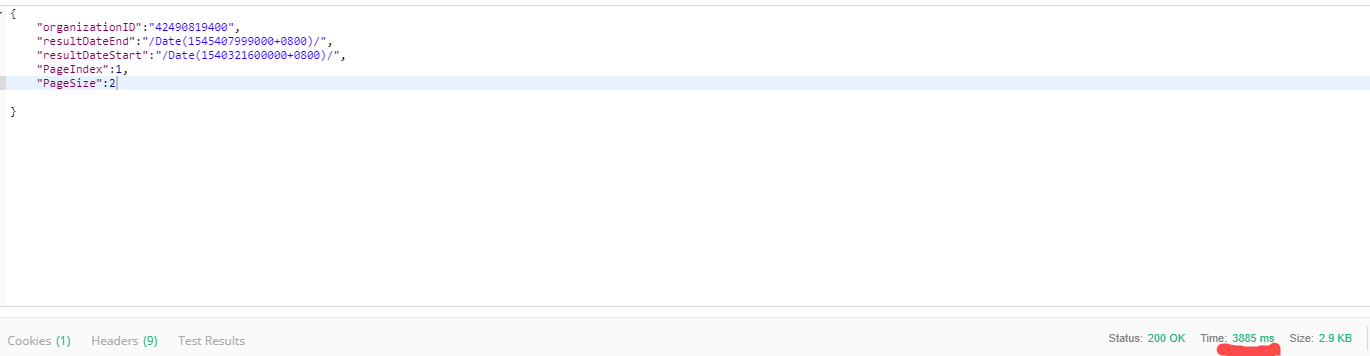


1824



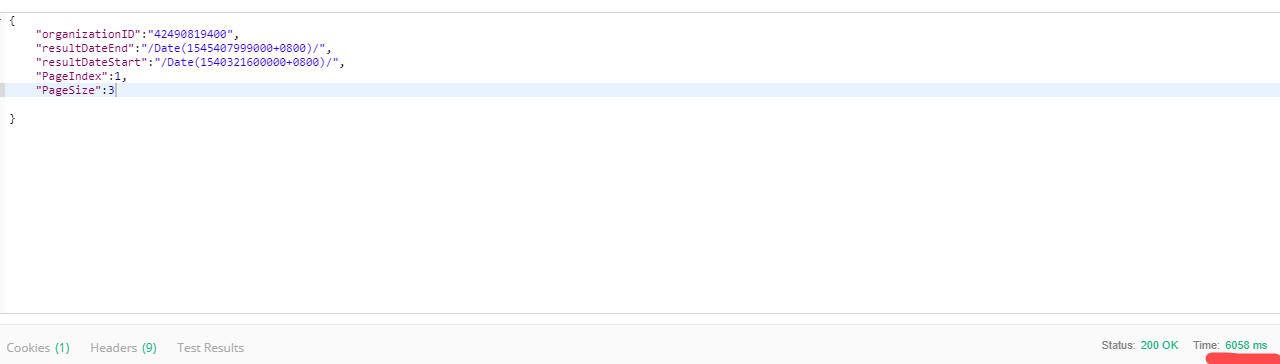
当为1，2

3885

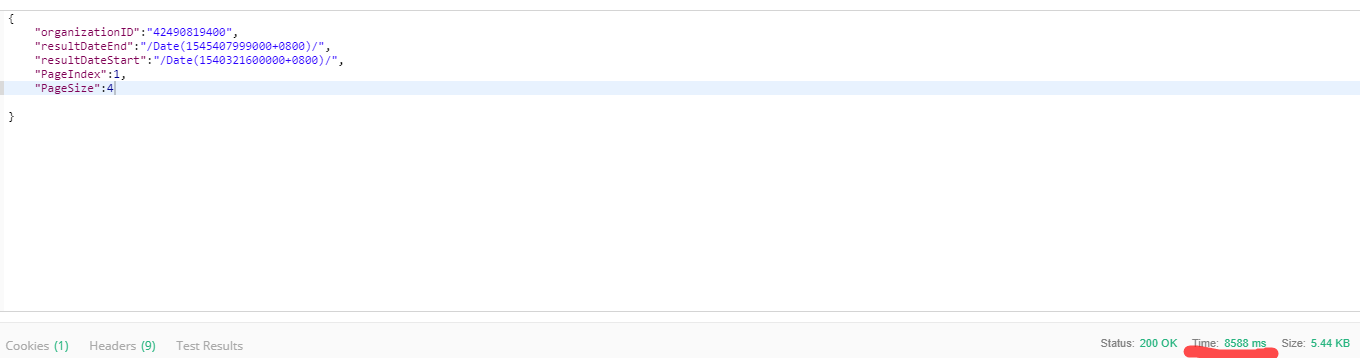


1，3

6058ms



1，4

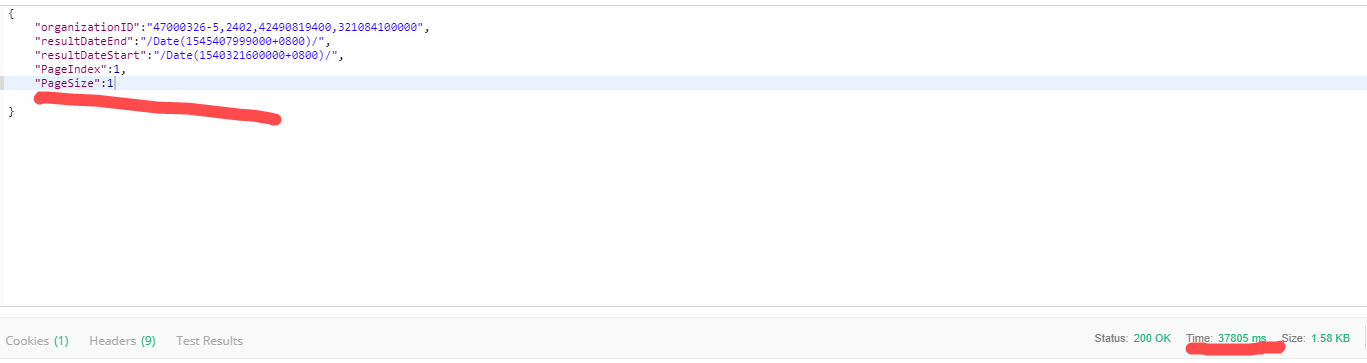


就是主要递增，而如果组织变成多个，直接变成最初的60s了，当同样每页10个数据。

如果1，1（一页一个）

37805ms，半分多钟了。

所以这个代码根据就不适合用多条件，这里的组织



8499

4.

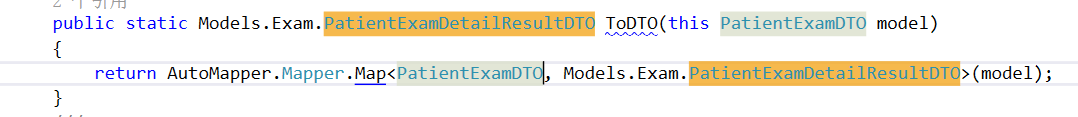
详情

web api中create map

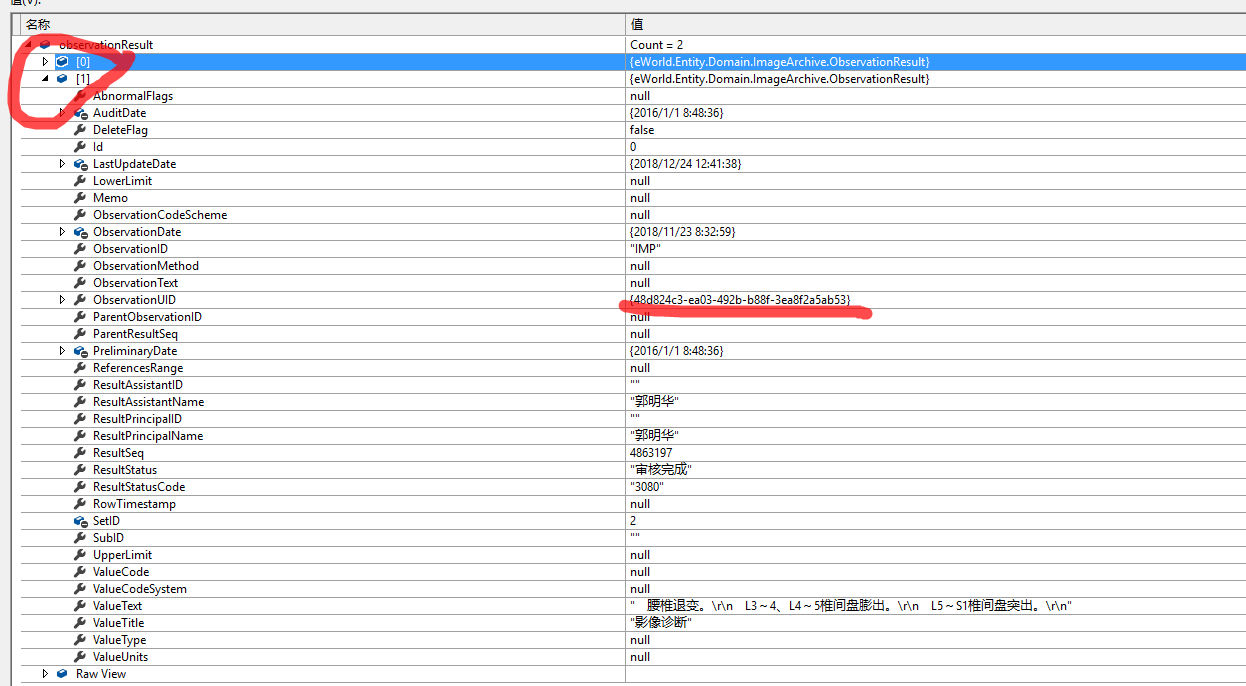


ToDTO里面实现映射。

data = resultdata.Result.ToDTO();

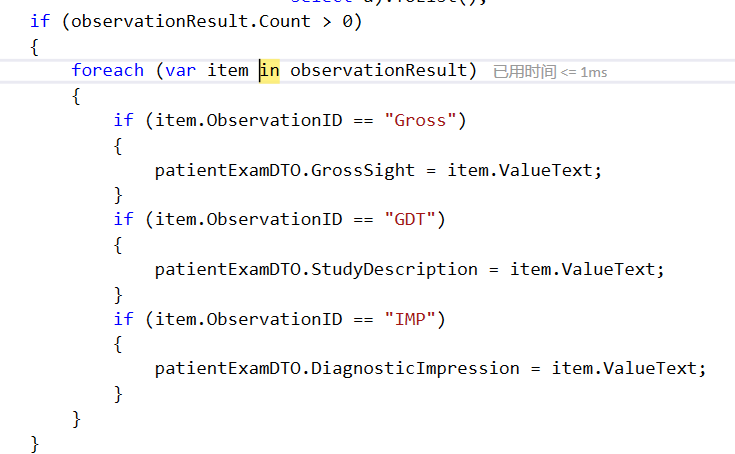


同一个观察ID，出现两条观察结果



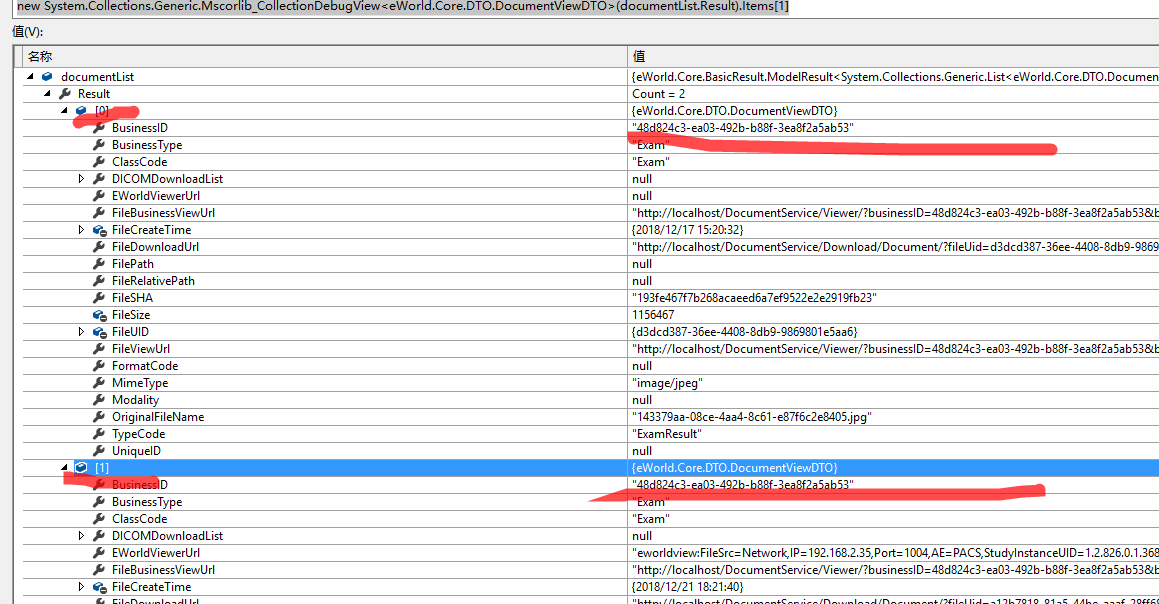
这里对于观察结果，for循环，获取的不是最后一个的结果么

不对，这里一个是影像所见，一个是诊断信息。

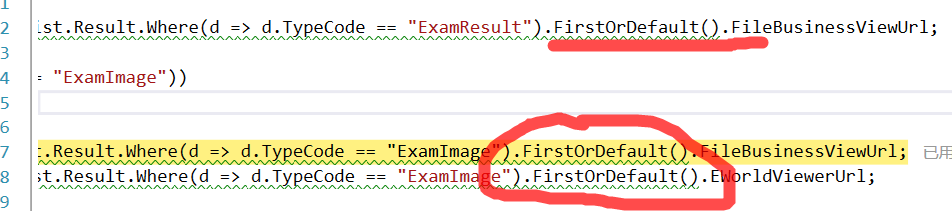


所以，正常的情况下，一个观察请求对应的，就是有两条观察结果的。

对应的Document表中也有两条数据：

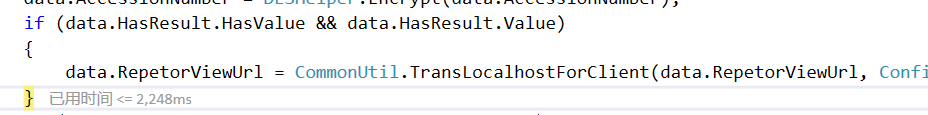


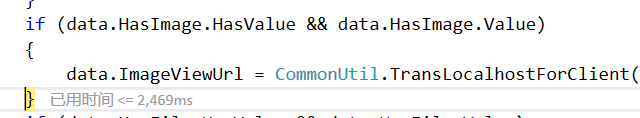
最终用的数据还是第一个



问题：

数据的dto之后





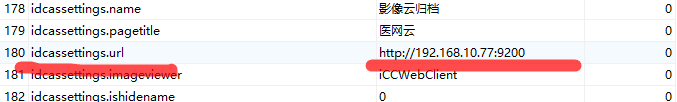
//获取相关检查的数量时，为什么要减1呢？

patientExamDTO.RelatedExamCount = relatedExamCount - 1;

涉及到病人主索引，观察请求，结果表，就诊visit表，然后去DataSourceSetup（数据来源对应检查类型）表中获取名称，再去document表中获取url地址，

"name":"95927258CF07D8329527472A9B9E2D5C",

5.



获取这个配置

6.

优化的问题在于获取配置数据上，要获取所有的，然后去获取当前的，现在参照协同，直接获取当前所需的，就ok了。

public static class ConfigDefine

{

private static readonly IDCASSettings \_idcasSettings = EngineContext.Current.Resolve<IDCASSettings>(); //影像存储共享设置

public static ILog \_log = LogManager.GetLogger(typeof(ConfigDefine));

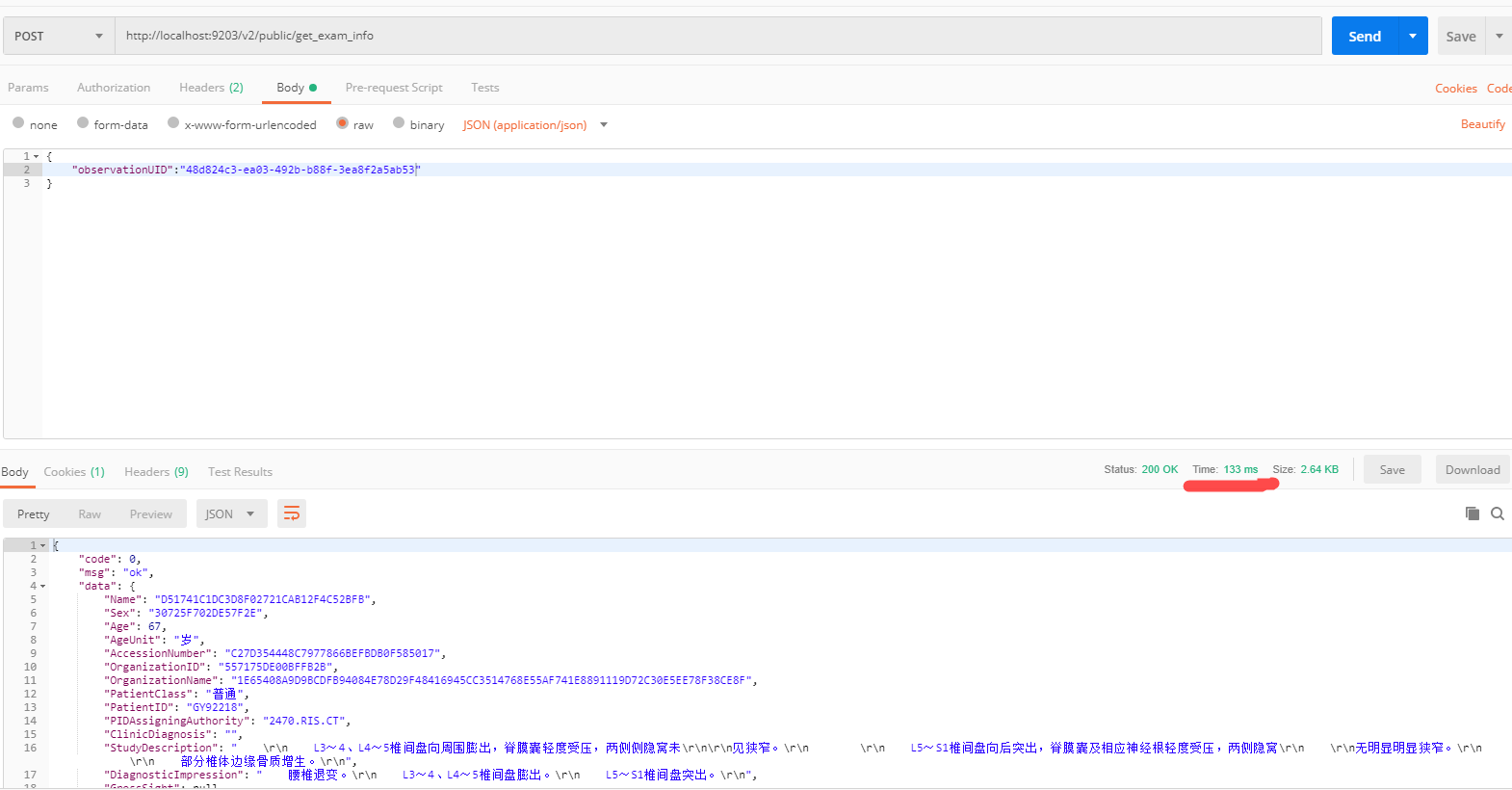
/// <summary>

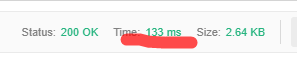
/// 网站地址

/// </summary>

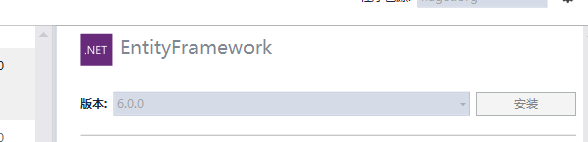
public static string Url => \_idcasSettings.Url;

瞬间ok了。





7.



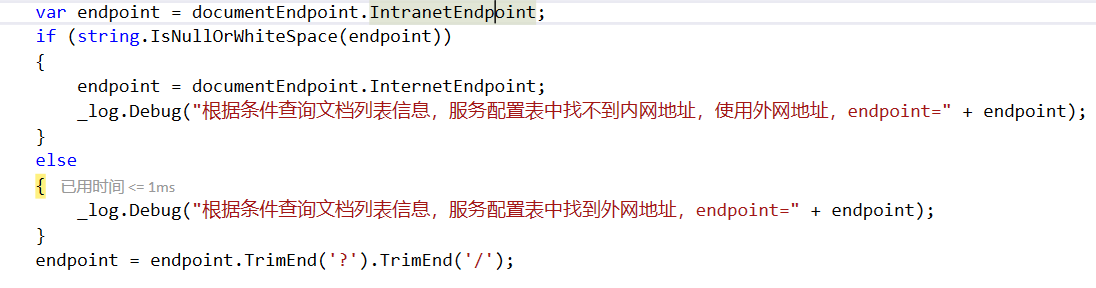
搞了半天，分支版本是曹兵漏传东西了。

202去发布

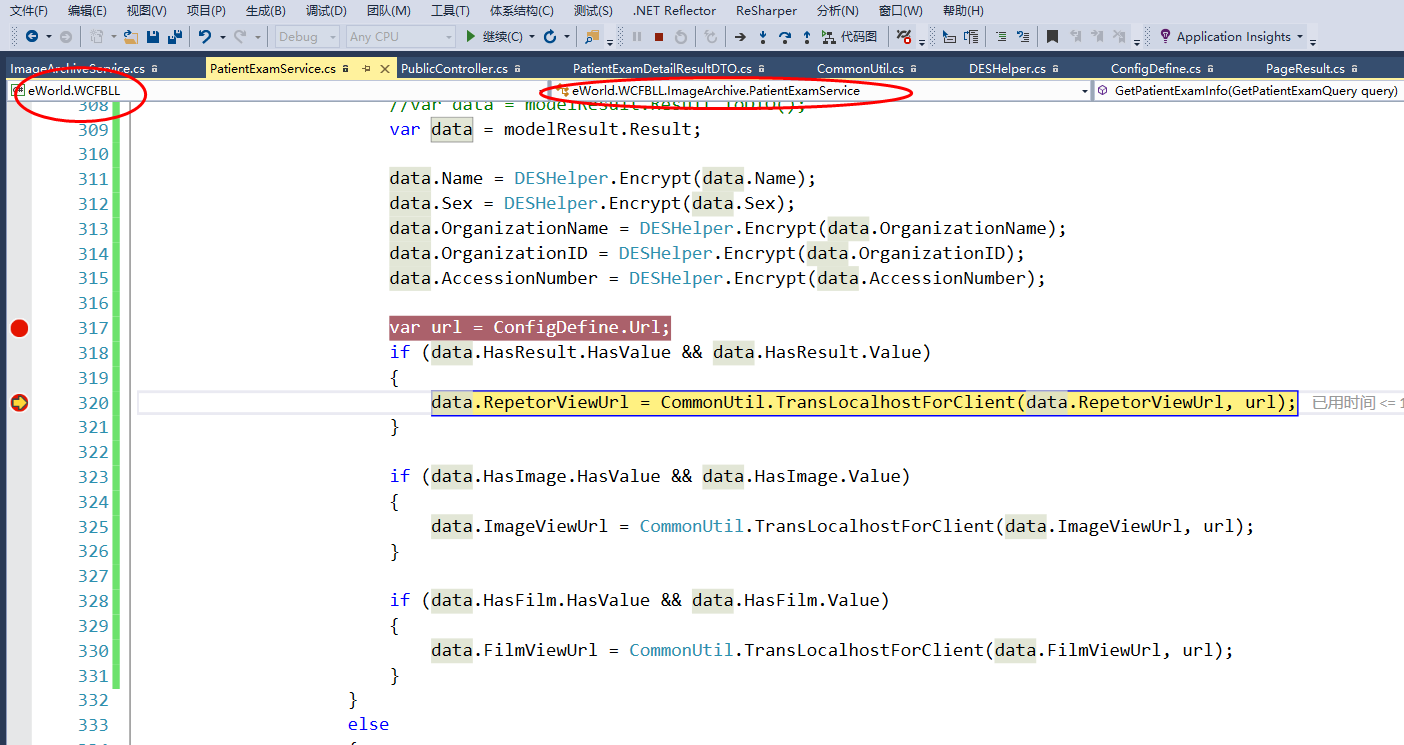
8.

问题1：

这里为什么优先使用内网地址？用外网不就直接拿到文档地址了。



问题2，把url结果处理放在bll层，可不可以。



放到了bll层

//更改（到时候删除）

//var data = modelResult.Result.ToDTO();

var data = modelResult.Result;

data.Name = DESHelper.Encrypt(data.Name);

data.Sex = DESHelper.Encrypt(data.Sex);

data.OrganizationName = DESHelper.Encrypt(data.OrganizationName);

data.OrganizationID = DESHelper.Encrypt(data.OrganizationID);

data.AccessionNumber = DESHelper.Encrypt(data.AccessionNumber);

var url = ConfigDefine.IDCASUrl;

//var url = ConfigDefine.Url;

//var idcasurl = ConfigDefine.IDCASUrl;

if (data.HasResult.HasValue && data.HasResult.Value)

{

data.RepetorViewUrl = CommonUtil.TransLocalhostForClient(data.RepetorViewUrl, url);

}

if (data.HasImage.HasValue && data.HasImage.Value)

{

data.ImageViewUrl = CommonUtil.TransLocalhostForClient(data.ImageViewUrl, url);

}

if (data.HasFilm.HasValue && data.HasFilm.Value)

{

data.FilmViewUrl = CommonUtil.TransLocalhostForClient(data.FilmViewUrl, url);

}

}

else

{

modelResult.code = ResultCodeEnum.Fail;

modelResult.msg = "未检索到检查记录";

}

老简的意思就是放这里了，能在云归档的wcf里面处理掉，就不要放到能力开放平台。

要不加一个state，我调的时候，置为1，然后在bll层去判断，然后我能力开放平台调用的时候，直接入参的时候，state=1，传过来调用云归档的wcf。

9.

把结果集都挪到契约层

dto

枚举：

namespace eWorld.Core.Enums

{

/// <summary>

/// 接口返回状态值

/// </summary>

public enum ReturnInfoEnum

{

base：

namespace eWorld.WebAPI.Models

{

/// <summary>

/// 接口返回基础参数

/// </summary>

public class BaseReturnResult

{

result：

namespace eWorld.WebAPI.Models

{

/// <summary>

/// 基础Model返回模型

/// </summary>

/// <typeparam name="T"></typeparam>

public class ModelReturnResult<T> : BaseReturnResult

{

/// <summary>

/// 结果模型

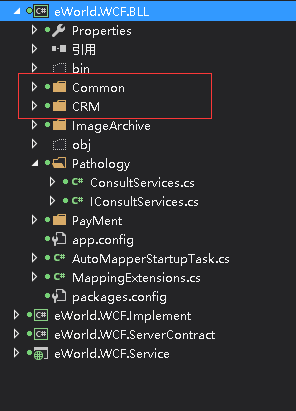
/// </summary>

public T data { get; set; }

}

}

这两个不用改：



最终要去掉这两个

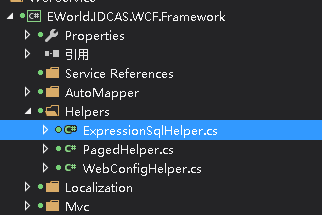
10.

ModelReturnResult

ModelReturnResult<model>

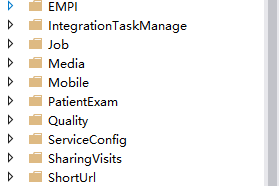
ModelReturnResult<list>

ModelReturnResult<pagelist>



PageReturnResult

11.



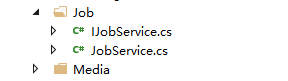
ReturnInfoEnum

法学这边的分页还没有传上来，所以其他都先改掉。

DataResult、ListResult和PageResult继承于BaseResult

70001

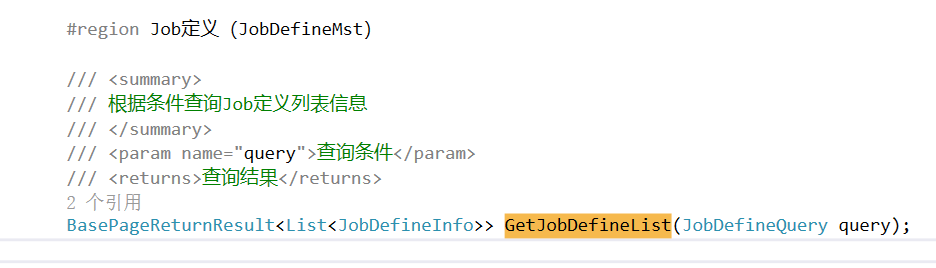
job可以最后去改



集成任务Service改造

手机端用户查询服务改造

到这个



集成任务服务改造一部分

12.

分页改造

var result = new BasePageReturnResult<List<JobTriggerDataMapInfo>>();

// pageResult = querySql.Project().To<JobTriggerDataMapInfo>().ToPageList(query.PageIndex, query.PageSize, query.Sort, query.Order);

querySql = ExpressionSqlHelper.ExecuteSort(querySql, query.Sort, query.Order);

PagedList<JobTriggerDataMapMst> page = new PagedList<JobTriggerDataMapMst>(querySql, query.PageIndex, query.PageSize);

Mapper.CreateMap<JobTriggerDataMapMst, JobTriggerDataMapInfo>();

List<JobTriggerDataMapInfo> sPurpostList = Mapper.Map<List<JobTriggerDataMapMst>, List<JobTriggerDataMapInfo>>(page);

result.data = sPurpostList;

result.page = new PageInfo()

{

HasNextPage = page.PageIndex + 1 < page.TotalPages,

HasPreviousPage = page.PageIndex > 0,

PageIndex = page.PageIndex,

PageSize = page.PageSize,

TotalCount = page.TotalCount,

TotalPages = page.TotalPages

};

return result;

}

catch (Exception ex)

{

\_log.Error(ex.Message, ex);

result.code = ReturnInfoEnum.ServiceBusy;

result.msg = "系统繁忙，请稍后再试";

return result;

}

普通的返回：

result.code = 70001;

result.msg = "系统繁忙，请稍后再试";

return result;

带泛型T model 的返回

result.code = 0;

result.msg = "成功";

result.data = jobServerInfo;

list返回

result.code = 0;

result.msg = "成功";

result.data = resultList;

return result;

13.

数据存储策略服务修改

14.

采集服务的思维导图

校验，后台服务，每天同步了多少，成功了多少，失败了多少，失败原因。

15.

// begin

//Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

//Mapper.CreateMap<PatientIndex, PatientExamIndexDTO>();

//Mapper.CreateMap<Visit, PatientExamVisitDTO>();

//Mapper.CreateMap<ObservationResult, PatientObservationResultDTO>();

//var mark = new short[] { 1111, 1144, 1141, 1114 }; // 不需要返回采集列表的采集标识

//var observationRequestSql = from a in \_observationRequestRepository.Table

// join p in \_patientIndexRepository.Table on

// new {patientId = a.PatientID, pid = a.PIDAssigningAuthority}

// equals new {patientId = p.PatientID, pid = p.PIDAssigningAuthority}

// join v in \_visitRepository.Table on a.VisitUID equals v.VisitUID

// join o in \_observationResultRepository.Table on a.ObservationUID equals o.ObservationUID

// where

// a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue)

// where v.DeleteFlag == false

// where o.DeleteFlag == false

// select new CloudExamListDTO

// {

// ObservationRequest= Mapper.Map<ObservationRequest, PatientObservationRequestDTO>(a),

// PatientIndex= Mapper.Map<PatientIndex, PatientExamIndexDTO>(p),

// PatientVisit = Mapper.Map<Visit, PatientExamVisitDTO>(v),

// ObservationResultList =

// };

Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

Mapper.CreateMap<PatientIndex, PatientExamIndexDTO>();

Mapper.CreateMap<Visit, PatientExamVisitDTO>();

Mapper.CreateMap<ObservationResult, PatientObservationResultDTO>();

var mark = new short[] { 1111, 1144, 1141, 1114 }; // 不需要返回采集列表的采集标识

var observationRequestSql = from a in \_observationRequestRepository.Table

join p in \_patientIndexRepository.Table on

new { patientId = a.PatientID, pid = a.PIDAssigningAuthority }

equals new { patientId = p.PatientID, pid = p.PIDAssigningAuthority }

join v in \_visitRepository.Table on a.VisitUID equals v.VisitUID

//join o in \_observationResultRepository.Table on a.ObservationUID equals o.ObservationUID

where

a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue)

where v.DeleteFlag == false

//where o.DeleteFlag == false

select new

{

ObservationRequest = a,

PatientIndex = p,

PatientVisit = v

};

//select new CloudExamListDTO

// {

// ObservationRequest = Mapper.Map<ObservationRequest, PatientObservationRequestDTO>(a),

// PatientIndex = Mapper.Map<PatientIndex, PatientExamIndexDTO>(p),

// PatientVisit = Mapper.Map<Visit, PatientExamVisitDTO>(v)

// };

//var query = from ccgm in \_customerCircleGroupMemberRepository.Table

// join c in \_customerRepository.Table on ccgm.FriendCustomerGuid equals c.CustomerGuid

// join dm in \_doctorMemberRepository.Table on ccgm.FriendCustomerGuid equals dm.CustomerGuid

// where param.GroupId.Contains(ccgm.GroupId.Value)

// select new AddressBookModel

// {

// FriendCustomerGuid = ccgm.FriendCustomerGuid,

// HeadPicGuid = c.HeadPicGuid,

// PeerId = c.IMPeerId,

// CustomerName = c.CustomerName,

// HospitalName = dm.HospitalName,

// Initials = c.NameFirstLetter,

// //GroupId=ccgm.GroupId

// };

if (query.ObservationDate.HasValue)

{

observationRequestSql =

observationRequestSql.Where(

a => a.ObservationRequest.ObservationDate >= query.ObservationDate.Value);

}

if (!string.IsNullOrWhiteSpace(query.OrganizationID))

{

query.OrganizationID = query.OrganizationID.TrimEnd(',');

var organizationIDList = query.OrganizationID.Split(',');

if (organizationIDList.Length == 1)

{

observationRequestSql =

observationRequestSql.Where(p => p.ObservationRequest.OrganizationID == query.OrganizationID);

}

else

{

observationRequestSql =

observationRequestSql.Where(

p => organizationIDList.Contains(p.ObservationRequest.OrganizationID));

}

}

if (!string.IsNullOrWhiteSpace(query.ServiceSectID))

{

var examTypeList = new List<string>();

query.ServiceSectID = query.ServiceSectID.TrimEnd(',');

var serviceSectIDList = query.ServiceSectID.Split(',');

var examTypeSql = from a in \_dataSourceSetupRepository.Table

select a;

if (serviceSectIDList.Length == 1)

{

examTypeSql = examTypeSql.Where(p => p.DataSource == query.ServiceSectID);

}

else

{

examTypeSql = examTypeSql.Where(p => serviceSectIDList.Contains(p.DataSource));

}

if (examTypeSql.Any())

{

foreach (var item in examTypeSql.ToList())

{

examTypeList.Add(item.ExamineType);

}

}

if (examTypeList.Count == 1)

{

var examtype = examTypeList[0];

observationRequestSql =

observationRequestSql.Where(p => p.ObservationRequest.ServiceSectID == examtype);

}

else if (examTypeList.Count > 1)

{

observationRequestSql =

observationRequestSql.Where(p => examTypeList.Contains(p.ObservationRequest.ServiceSectID));

}

}

if (!query.RowCount.HasValue)

{

query.RowCount = 50;

}

observationRequestSql =

observationRequestSql.OrderByDescending(a => a.ObservationRequest.ObservationDate.Value);

observationRequestSql = observationRequestSql.Take(query.RowCount.Value);

Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

// var list = observationRequestSql.Project().To<PatientObservationRequestDTO>().ToList();

var list = observationRequestSql.ToList();

if (list.Count > 0)

{

foreach (var item in list)

{

if (!item.ObservationRequest.CollectionMark.HasValue)

{

item.ObservationRequest.CollectionMark = 1000;

}

{

model = new CloudExamListDTO

{

ObservationRequest = Mapper.Map<ObservationRequest,

PatientObservationRequestDTO>(item.ObservationRequest),

PatientIndex = Mapper.Map<PatientIndex,

PatientExamIndexDTO>(item.PatientIndex),

PatientVisit = Mapper.Map<Visit,

PatientExamVisitDTO>(item.PatientVisit),

ObservationResultList =

GetPatientExamObservationRequestList(item.ObservationRequest.ObservationUID)

};

resultList.Add(model);

}

}

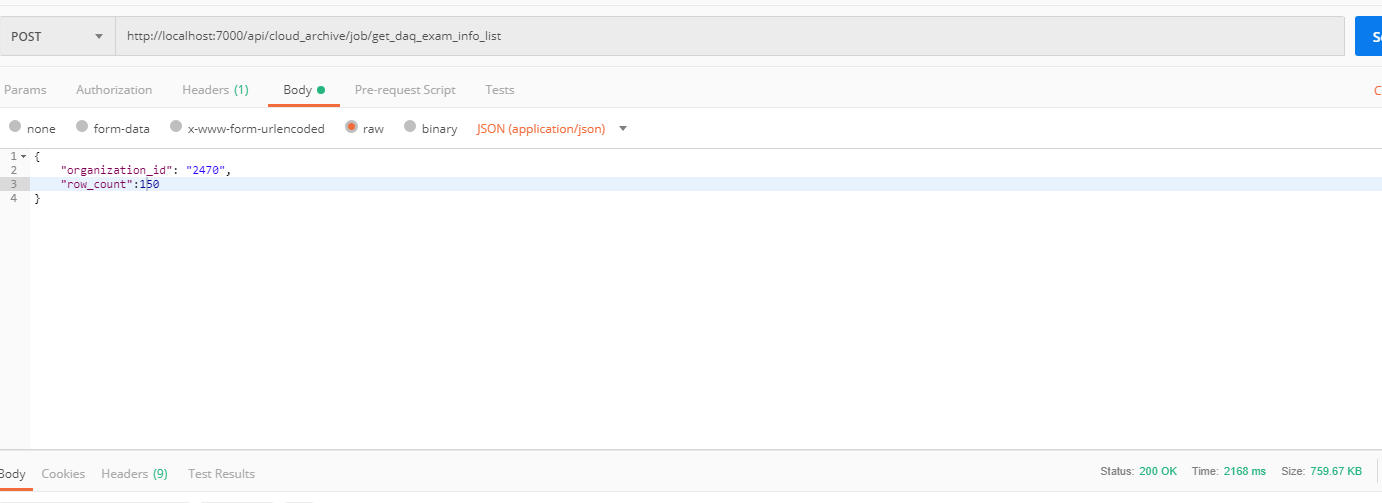
}

// end

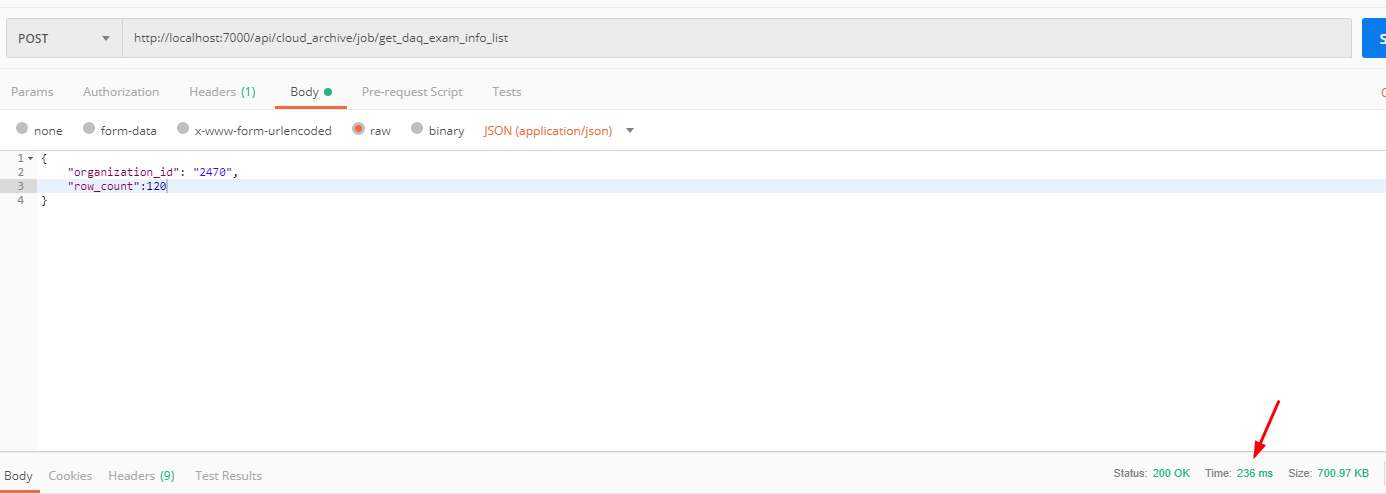
16.

150

原来 2168



120 2秒左右



换成连表以后，

236毫秒就出来了

不一起查出来，先把病人索引，观察，观察结果，三个连表出来，再最后for循环处理观察结果，这样就七百多毫秒了，所以用第二种一起查出来最好。

733 ms

三种代码我都保存下：

/// <summary>

/// 获取影像云采集列表

/// </summary>

/// <param name="query">查询条件</param>

/// <returns>影像云采集列表</returns>

public ListResult<CloudExamListDTO> GetCloudExamList(GetCloudExamListQuery query)

{

List<CloudExamListDTO> resultList = new List<CloudExamListDTO>();

CloudExamListDTO model = new CloudExamListDTO();

Stopwatch sw = null;

var result = new ListResult<CloudExamListDTO>() { code = (int)ReturnInfoEnum.Failed, msg = "Failed" };

try

{

if (\_logDebug)

{

sw = new Stopwatch();

sw.Start();

\_log.Debug("收到GetPatientExamList获取云到云采集列表方法调用，参数：" + JsonConvert.SerializeObject(query));

}

// var observationRequestSql = from a in \_observationRequestRepository.Table

// where a.DeleteFlag == false && ((a.CollectionMark.Value != 1111 && a.CollectionMark != 1144) || !a.CollectionMark.HasValue)

// select a;

//之前

//var mark = new short[] { 1111, 1144, 1141, 1114 }; // 不需要返回采集列表的采集标识

//var observationRequestSql = from a in \_observationRequestRepository.Table

// where a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue)

// select a;

//if (query.ObservationDate.HasValue)

//{

// observationRequestSql = observationRequestSql.Where(a => a.ObservationDate >= query.ObservationDate.Value);

//}

//if (!string.IsNullOrWhiteSpace(query.OrganizationID))

//{

// query.OrganizationID = query.OrganizationID.TrimEnd(',');

// var organizationIDList = query.OrganizationID.Split(',');

// if (organizationIDList.Length == 1)

// {

// observationRequestSql = observationRequestSql.Where(p => p.OrganizationID == query.OrganizationID);

// }

// else

// {

// observationRequestSql = observationRequestSql.Where(p => organizationIDList.Contains(p.OrganizationID));

// }

//}

//if (!string.IsNullOrWhiteSpace(query.ServiceSectID))

//{

// var examTypeList = new List<string>();

// query.ServiceSectID = query.ServiceSectID.TrimEnd(',');

// var serviceSectIDList = query.ServiceSectID.Split(',');

// var examTypeSql = from a in \_dataSourceSetupRepository.Table

// select a;

// if (serviceSectIDList.Length == 1)

// {

// examTypeSql = examTypeSql.Where(p => p.DataSource == query.ServiceSectID);

// }

// else

// {

// examTypeSql = examTypeSql.Where(p => serviceSectIDList.Contains(p.DataSource));

// }

// if (examTypeSql.Any())

// {

// foreach (var item in examTypeSql.ToList())

// {

// examTypeList.Add(item.ExamineType);

// }

// }

// if (examTypeList.Count == 1)

// {

// var examtype = examTypeList[0];

// observationRequestSql = observationRequestSql.Where(p => p.ServiceSectID == examtype);

// }

// else if (examTypeList.Count > 1)

// {

// observationRequestSql = observationRequestSql.Where(p => examTypeList.Contains(p.ServiceSectID));

// }

//}

//if (!query.RowCount.HasValue)

//{

// query.RowCount = 50;

//}

//observationRequestSql = observationRequestSql.OrderByDescending(a => a.ObservationDate.Value);

//observationRequestSql = observationRequestSql.Take(query.RowCount.Value);

//Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

//var list = observationRequestSql.Project().To<PatientObservationRequestDTO>().ToList();

//if (list.Count > 0)

//{

// foreach (var item in list)

// {

// if (!item.CollectionMark.HasValue)

// {

// item.CollectionMark = 1000;

// }

// model = new CloudExamListDTO

// {

// ObservationRequest = item,

// PatientIndex = GetPatientExamIndexDTO(item.PatientID, item.PIDAssigningAuthority),

// PatientVisit = GetPatientExamVisitDTO(item.VisitUID),

// ObservationResultList = GetPatientExamObservationRequestList(item.ObservationUID)

// };

// resultList.Add(model);

// }

//}

// begin

//Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

//Mapper.CreateMap<PatientIndex, PatientExamIndexDTO>();

//Mapper.CreateMap<Visit, PatientExamVisitDTO>();

//Mapper.CreateMap<ObservationResult, PatientObservationResultDTO>();

//var mark = new short[] { 1111, 1144, 1141, 1114 }; // 不需要返回采集列表的采集标识

//var observationRequestSql = from a in \_observationRequestRepository.Table

// join p in \_patientIndexRepository.Table on

// new { patientId = a.PatientID, pid = a.PIDAssigningAuthority }

// equals new { patientId = p.PatientID, pid = p.PIDAssigningAuthority }

// join v in \_visitRepository.Table on a.VisitUID equals v.VisitUID

// //join o in \_observationResultRepository.Table on a.ObservationUID equals o.ObservationUID

// where

// a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue)

// where v.DeleteFlag == false

// //where o.DeleteFlag == false

// select new

// {

// ObservationRequest = a,

// PatientIndex = p,

// PatientVisit = v,

// ObservationResultList = \_observationResultRepository.Table.Where(o => o.ObservationUID == a.ObservationUID && o.DeleteFlag == false)/\*.ToList()\*/

// };

//if (query.ObservationDate.HasValue)

//{

// observationRequestSql =

// observationRequestSql.Where(

// a => a.ObservationRequest.ObservationDate >= query.ObservationDate.Value);

//}

//if (!string.IsNullOrWhiteSpace(query.OrganizationID))

//{

// query.OrganizationID = query.OrganizationID.TrimEnd(',');

// var organizationIDList = query.OrganizationID.Split(',');

// if (organizationIDList.Length == 1)

// {

// observationRequestSql =

// observationRequestSql.Where(p => p.ObservationRequest.OrganizationID == query.OrganizationID);

// }

// else

// {

// observationRequestSql =

// observationRequestSql.Where(

// p => organizationIDList.Contains(p.ObservationRequest.OrganizationID));

// }

//}

//if (!string.IsNullOrWhiteSpace(query.ServiceSectID))

//{

// var examTypeList = new List<string>();

// query.ServiceSectID = query.ServiceSectID.TrimEnd(',');

// var serviceSectIDList = query.ServiceSectID.Split(',');

// var examTypeSql = from a in \_dataSourceSetupRepository.Table

// select a;

// if (serviceSectIDList.Length == 1)

// {

// examTypeSql = examTypeSql.Where(p => p.DataSource == query.ServiceSectID);

// }

// else

// {

// examTypeSql = examTypeSql.Where(p => serviceSectIDList.Contains(p.DataSource));

// }

// if (examTypeSql.Any())

// {

// foreach (var item in examTypeSql.ToList())

// {

// examTypeList.Add(item.ExamineType);

// }

// }

// if (examTypeList.Count == 1)

// {

// var examtype = examTypeList[0];

// observationRequestSql =

// observationRequestSql.Where(p => p.ObservationRequest.ServiceSectID == examtype);

// }

// else if (examTypeList.Count > 1)

// {

// observationRequestSql =

// observationRequestSql.Where(p => examTypeList.Contains(p.ObservationRequest.ServiceSectID));

// }

//}

//if (!query.RowCount.HasValue)

//{

// query.RowCount = 50;

//}

//observationRequestSql =

// observationRequestSql.OrderByDescending(a => a.ObservationRequest.ObservationDate.Value);

//observationRequestSql = observationRequestSql.Take(query.RowCount.Value);

//// var list = observationRequestSql.Project().To<PatientObservationRequestDTO>().ToList();

//var list = observationRequestSql.ToList();

//if (list.Count > 0)

//{

// foreach (var item in list)

// {

// if (!item.ObservationRequest.CollectionMark.HasValue)

// {

// item.ObservationRequest.CollectionMark = 1000;

// }

// {

// if (item.PatientVisit != null)

// {

// var ageAndUnit = CommonUtil.AgeUnitFormat(item.PatientVisit.Age ?? 0, item.PatientVisit.AgeUnit);

// item.PatientVisit.Age = ageAndUnit.Item1;

// item.PatientVisit.AgeUnit = ageAndUnit.Item2;

// }

// model = new CloudExamListDTO

// {

// ObservationRequest = Mapper.Map<ObservationRequest,

// PatientObservationRequestDTO>(item.ObservationRequest),

// PatientIndex = Mapper.Map<PatientIndex,

// PatientExamIndexDTO>(item.PatientIndex),

// PatientVisit = Mapper.Map<Visit,

// PatientExamVisitDTO>(item.PatientVisit),

// ObservationResultList = Mapper.Map<List<ObservationResult>, List<PatientObservationResultDTO>>(item.ObservationResultList.ToList())

// };

// resultList.Add(model);

// }

// }

//}

// end

//2

Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

Mapper.CreateMap<PatientIndex, PatientExamIndexDTO>();

Mapper.CreateMap<Visit, PatientExamVisitDTO>();

Mapper.CreateMap<ObservationResult, PatientObservationResultDTO>();

var mark = new short[] { 1111, 1144, 1141, 1114 }; // 不需要返回采集列表的采集标识

var observationRequestSql = from a in \_observationRequestRepository.Table

join p in \_patientIndexRepository.Table on

new { patientId = a.PatientID, pid = a.PIDAssigningAuthority }

equals new { patientId = p.PatientID, pid = p.PIDAssigningAuthority }

join v in \_visitRepository.Table on a.VisitUID equals v.VisitUID

//join o in \_observationResultRepository.Table on a.ObservationUID equals o.ObservationUID

where

a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue)

where v.DeleteFlag == false

//where o.DeleteFlag == false

select new

{

ObservationRequest = a,

PatientIndex = p,

PatientVisit = v

};

if (query.ObservationDate.HasValue)

{

observationRequestSql =

observationRequestSql.Where(

a => a.ObservationRequest.ObservationDate >= query.ObservationDate.Value);

}

if (!string.IsNullOrWhiteSpace(query.OrganizationID))

{

query.OrganizationID = query.OrganizationID.TrimEnd(',');

var organizationIDList = query.OrganizationID.Split(',');

if (organizationIDList.Length == 1)

{

observationRequestSql =

observationRequestSql.Where(p => p.ObservationRequest.OrganizationID == query.OrganizationID);

}

else

{

observationRequestSql =

observationRequestSql.Where(

p => organizationIDList.Contains(p.ObservationRequest.OrganizationID));

}

}

if (!string.IsNullOrWhiteSpace(query.ServiceSectID))

{

var examTypeList = new List<string>();

query.ServiceSectID = query.ServiceSectID.TrimEnd(',');

var serviceSectIDList = query.ServiceSectID.Split(',');

var examTypeSql = from a in \_dataSourceSetupRepository.Table

select a;

if (serviceSectIDList.Length == 1)

{

examTypeSql = examTypeSql.Where(p => p.DataSource == query.ServiceSectID);

}

else

{

examTypeSql = examTypeSql.Where(p => serviceSectIDList.Contains(p.DataSource));

}

if (examTypeSql.Any())

{

foreach (var item in examTypeSql.ToList())

{

examTypeList.Add(item.ExamineType);

}

}

if (examTypeList.Count == 1)

{

var examtype = examTypeList[0];

observationRequestSql =

observationRequestSql.Where(p => p.ObservationRequest.ServiceSectID == examtype);

}

else if (examTypeList.Count > 1)

{

observationRequestSql =

observationRequestSql.Where(p => examTypeList.Contains(p.ObservationRequest.ServiceSectID));

}

}

if (!query.RowCount.HasValue)

{

query.RowCount = 50;

}

observationRequestSql =

observationRequestSql.OrderByDescending(a => a.ObservationRequest.ObservationDate.Value);

observationRequestSql = observationRequestSql.Take(query.RowCount.Value);

//Mapper.CreateMap<ObservationRequest, PatientObservationRequestDTO>();

// var list = observationRequestSql.Project().To<PatientObservationRequestDTO>().ToList();

var list = observationRequestSql.ToList();

if (list.Count > 0)

{

foreach (var item in list)

{

if (!item.ObservationRequest.CollectionMark.HasValue)

{

item.ObservationRequest.CollectionMark = 1000;

}

if (item.PatientVisit != null)

{

var ageAndUnit = CommonUtil.AgeUnitFormat(item.PatientVisit.Age ?? 0, item.PatientVisit.AgeUnit);

item.PatientVisit.Age = ageAndUnit.Item1;

item.PatientVisit.AgeUnit = ageAndUnit.Item2;

}

{

model = new CloudExamListDTO

{

ObservationRequest = Mapper.Map<ObservationRequest,

PatientObservationRequestDTO>(item.ObservationRequest),

PatientIndex = Mapper.Map<PatientIndex,

PatientExamIndexDTO>(item.PatientIndex),

PatientVisit = Mapper.Map<Visit,

PatientExamVisitDTO>(item.PatientVisit),

ObservationResultList =

GetPatientExamObservationRequestList(item.ObservationRequest.ObservationUID)

};

resultList.Add(model);

}

}

}

//2

result.code = (int)ReturnInfoEnum.Success;

result.msg = "OK";

result.data = resultList;

return result;

17.

云到云获取检查列表，最终的sql语句：

SELECT

`Project2`.`C1`,

`Project2`.`ObservationUID`,

`Project2`.`PatientID`,

`Project2`.`PIDAssigningAuthority`,

`Project2`.`PatientMasterID`,

`Project2`.`PatientClass`,

`Project2`.`VisitUID`,

`Project2`.`OrderUID`,

`Project2`.`OrganizationID`,

`Project2`.`OrganizationName`,

`Project2`.`SourceOrganizationID`,

`Project2`.`SourceOrganizationName`,

`Project2`.`ClinicInfoType`,

`Project2`.`PlacerOrderNO`,

`Project2`.`PlacerAssigningAuthority`,

`Project2`.`PlacerOrderDetailNO`,

`Project2`.`FillerOrderNO`,

`Project2`.`FillerAssigningAuthority`,

`Project2`.`FillerPatientID`,

`Project2`.`AccessionNumber`,

`Project2`.`ServiceID`,

`Project2`.`ServiceCodeScheme`,

`Project2`.`ServiceText`,

`Project2`.`ServiceIDInsur`,

`Project2`.`ServiceSectID`,

`Project2`.`ProcedureID`,

`Project2`.`ProcedureName`,

`Project2`.`ProviderID`,

`Project2`.`ProviderName`,

`Project2`.`ProviderPhone`,

`Project2`.`RequestDeptID`,

`Project2`.`RequestDeptName`,

`Project2`.`RequestOrgID`,

`Project2`.`RequestOrgName`,

`Project2`.`RequestedDate`,

`Project2`.`Reason`,

`Project2`.`Attention`,

`Project2`.`Symptom`,

`Project2`.`AdverseReaction`,

`Project2`.`ClinicDiagnosis`,

`Project2`.`RelevantClinicalInfo`,

`Project2`.`FastingFlag`,

`Project2`.`TransportationMode`,

`Project2`.`SpecimenID`,

`Project2`.`SpecimenName`,

`Project2`.`SpecimenType`,

`Project2`.`SpecimenSource`,

`Project2`.`CollectorID`,

`Project2`.`CollectorName`,

`Project2`.`CollectionDept`,

`Project2`.`CollectionMethod`,

`Project2`.`CollectionVolume`,

`Project2`.`SpecimenReceivedDate`,

`Project2`.`ScheduledDate`,

`Project2`.`ScheduledDateEnd`,

`Project2`.`RegTime`,

`Project2`.`RegisterID`,

`Project2`.`RegisterName`,

`Project2`.`ObservationDeptID`,

`Project2`.`ObservationDeptName`,

`Project2`.`ObservationDate`,

`Project2`.`ObservationEndDate`,

`Project2`.`ObservationLocation`,

`Project2`.`ObservationEquipmentID`,

`Project2`.`ObservationEquipment`,

`Project2`.`EquipmentModel`,

`Project2`.`ObservationMethod`,

`Project2`.`StudyInstanceUID`,

`Project2`.`TechnicianID`,

`Project2`.`TechnicianName`,

`Project2`.`ResultAssistantID`,

`Project2`.`ResultAssistantName`,

`Project2`.`ResultAssistantPhone`,

`Project2`.`ResultPrincipalID`,

`Project2`.`ResultPrincipalName`,

`Project2`.`ResultReviseID`,

`Project2`.`ResultReviseName`,

`Project2`.`PreliminaryDate`,

`Project2`.`AuditDate`,

`Project2`.`ReviseDate`,

`Project2`.`ResultDate`,

`Project2`.`ResultOrganizationID`,

`Project2`.`ResultServiceCenterUID`,

`Project2`.`ResultStatus`,

`Project2`.`ResultStatusCode`,

`Project2`.`ResultPrintCount`,

`Project2`.`ResultPrintTime`,

`Project2`.`AbnormalFlags`,

`Project2`.`CriticalValue`,

`Project2`.`InfectionName`,

`Project2`.`PrivacyLevel`,

`Project2`.`PrivacyLevelExt`,

`Project2`.`Charges`,

`Project2`.`Payments`,

`Project2`.`PaymentsFlag`,

`Project2`.`FilmCount`,

`Project2`.`FilmNeed`,

`Project2`.`HasImage`,

`Project2`.`ImageLocation`,

`Project2`.`ConsultStatus`,

`Project2`.`ConsultRequestDate`,

`Project2`.`ConsultEndDate`,

`Project2`.`ArchiveDate`,

`Project2`.`LastUpdateDate`,

`Project2`.`DataSource`,

`Project2`.`LockFlag`,

`Project2`.`LockUserUID`,

`Project2`.`LockReason`,

`Project2`.`InWritingUserUID`,

`Project2`.`MessageCount`,

`Project2`.`UnProcessWorkflowCount`,

`Project2`.`PushState`,

`Project2`.`DeleteFlag`,

`Project2`.`RowTimestamp`,

`Project2`.`DiskSpace`,

`Project2`.`IsSummary`,

`Project2`.`IsRepeatExamSum`,

`Project2`.`IsDiseaseAnalysisSum`,

`Project2`.`CollectionMark`,

`Project2`.`PatientID1`,

`Project2`.`PIDAssigningAuthority1`,

`Project2`.`IsMPI`,

`Project2`.`PatientMasterID1`,

`Project2`.`Password`,

`Project2`.`CreateDate`,

`Project2`.`CreateUserID`,

`Project2`.`CreateUserName`,

`Project2`.`CreateOrgnizationID`,

`Project2`.`Name`,

`Project2`.`NameSpell`,

`Project2`.`MotherName`,

`Project2`.`Sex`,

`Project2`.`BirthDate`,

`Project2`.`BirthPlace`,

`Project2`.`Nation`,

`Project2`.`Citizenship`,

`Project2`.`MaritalStatus`,

`Project2`.`IdentityType`,

`Project2`.`IdentityID`,

`Project2`.`IDCardNO`,

`Project2`.`HealthCardNO`,

`Project2`.`InsuranceType`,

`Project2`.`InsuranceID`,

`Project2`.`ContactPhoneNO`,

`Project2`.`HomePhoneNO`,

`Project2`.`BusinessPhoneNO`,

`Project2`.`Email`,

`Project2`.`AddressProvince`,

`Project2`.`AddressCity`,

`Project2`.`AddressDistrict`,

`Project2`.`AddressStreet`,

`Project2`.`AddressRoad`,

`Project2`.`AddressDetail`,

`Project2`.`Postalcode`,

`Project2`.`Occupation`,

`Project2`.`WorkUnit`,

`Project2`.`Language`,

`Project2`.`LastUpdateDate1`,

`Project2`.`InHospitalFlag`,

`Project2`.`Status`,

`Project2`.`RowTimestamp1`,

`Project2`.`VisitUID1`,

`Project2`.`PatientID2`,

`Project2`.`PIDAssigningAuthority2`,

`Project2`.`PatientMasterID2`,

`Project2`.`VisitID`,

`Project2`.`AlternateVisitID`,

`Project2`.`AssigningAuthority`,

`Project2`.`PatientClass1`,

`Project2`.`PatientType`,

`Project2`.`MedRecNO`,

`Project2`.`OutPatientNO`,

`Project2`.`InPatientNO`,

`Project2`.`CardType`,

`Project2`.`CardNO`,

`Project2`.`Age`,

`Project2`.`AgeUnit`,

`Project2`.`MaritalStatus1`,

`Project2`.`PregnancyFlag`,

`Project2`.`Occupation1`,

`Project2`.`OrganizationID1`,

`Project2`.`AdmitDate`,

`Project2`.`ReAmissionFlag`,

`Project2`.`AdmissionType`,

`Project2`.`AdmitPatCondition`,

`Project2`.`AdmitDeptID`,

`Project2`.`AdmitDeptName`,

`Project2`.`DeptID`,

`Project2`.`DeptName`,

`Project2`.`PointOfCare`,

`Project2`.`PointOfCareID`,

`Project2`.`Room`,

`Project2`.`Bed`,

`Project2`.`AdmittingDoctorID`,

`Project2`.`AdmittingDoctorName`,

`Project2`.`AttendingDoctorID`,

`Project2`.`AttendingDoctorName`,

`Project2`.`DirectorDoctorID`,

`Project2`.`DirectorDoctorName`,

`Project2`.`CurPatCondition`,

`Project2`.`NursingGrade`,

`Project2`.`DietType`,

`Project2`.`IsHasPacemaker`,

`Project2`.`PrimaryDiagnosis`,

`Project2`.`DischargeDate`,

`Project2`.`DischargeDisposition`,

`Project2`.`TotalCharges`,

`Project2`.`TotalPayments`,

`Project2`.`PaymentMode`,

`Project2`.`MedicalPaymentMode`,

`Project2`.`LastUpdateDate2`,

`Project2`.`DeleteFlag1`,

`Project2`.`RowTimestamp2`,

`Project2`.`C2`,

`Project2`.`ResultSeq`,

`Project2`.`ObservationUID1`,

`Project2`.`SetID`,

`Project2`.`ObservationID`,

`Project2`.`ObservationCodeScheme`,

`Project2`.`ObservationText`,

`Project2`.`SubID`,

`Project2`.`ValueTitle`,

`Project2`.`ValueType`,

`Project2`.`ValueText`,

`Project2`.`ValueCode`,

`Project2`.`ValueCodeSystem`,

`Project2`.`ValueUnits`,

`Project2`.`ResultStatus1`,

`Project2`.`ResultStatusCode1`,

`Project2`.`ReferencesRange`,

`Project2`.`AbnormalFlags1`,

`Project2`.`ObservationMethod1`,

`Project2`.`ObservationDate1`,

`Project2`.`PreliminaryDate1`,

`Project2`.`AuditDate1`,

`Project2`.`ResultPrincipalID1`,

`Project2`.`ResultPrincipalName1`,

`Project2`.`ResultAssistantID1`,

`Project2`.`ResultAssistantName1`,

`Project2`.`LowerLimit`,

`Project2`.`UpperLimit`,

`Project2`.`Memo`,

`Project2`.`ParentResultSeq`,

`Project2`.`ParentObservationID`,

`Project2`.`LastUpdateDate3`,

`Project2`.`DeleteFlag2`,

`Project2`.`RowTimestamp3`

FROM

(

SELECT

`Limit1`.`ObservationUID`,

`Limit1`.`PatientID`,

`Limit1`.`PIDAssigningAuthority`,

`Limit1`.`PatientMasterID`,

`Limit1`.`PatientClass`,

`Limit1`.`VisitUID`,

`Limit1`.`OrderUID`,

`Limit1`.`OrganizationID`,

`Limit1`.`OrganizationName`,

`Limit1`.`SourceOrganizationID`,

`Limit1`.`SourceOrganizationName`,

`Limit1`.`ClinicInfoType`,

`Limit1`.`PlacerOrderNO`,

`Limit1`.`PlacerAssigningAuthority`,

`Limit1`.`PlacerOrderDetailNO`,

`Limit1`.`FillerOrderNO`,

`Limit1`.`FillerAssigningAuthority`,

`Limit1`.`FillerPatientID`,

`Limit1`.`AccessionNumber`,

`Limit1`.`ServiceID`,

`Limit1`.`ServiceCodeScheme`,

`Limit1`.`ServiceText`,

`Limit1`.`ServiceIDInsur`,

`Limit1`.`ServiceSectID`,

`Limit1`.`ProcedureID`,

`Limit1`.`ProcedureName`,

`Limit1`.`ProviderID`,

`Limit1`.`ProviderName`,

`Limit1`.`ProviderPhone`,

`Limit1`.`RequestDeptID`,

`Limit1`.`RequestDeptName`,

`Limit1`.`RequestOrgID`,

`Limit1`.`RequestOrgName`,

`Limit1`.`RequestedDate`,

`Limit1`.`Reason`,

`Limit1`.`Attention`,

`Limit1`.`Symptom`,

`Limit1`.`AdverseReaction`,

`Limit1`.`ClinicDiagnosis`,

`Limit1`.`RelevantClinicalInfo`,

`Limit1`.`FastingFlag`,

`Limit1`.`TransportationMode`,

`Limit1`.`SpecimenID`,

`Limit1`.`SpecimenName`,

`Limit1`.`SpecimenType`,

`Limit1`.`SpecimenSource`,

`Limit1`.`CollectorID`,

`Limit1`.`CollectorName`,

`Limit1`.`CollectionDept`,

`Limit1`.`CollectionMethod`,

`Limit1`.`CollectionVolume`,

`Limit1`.`SpecimenReceivedDate`,

`Limit1`.`ScheduledDate`,

`Limit1`.`ScheduledDateEnd`,

`Limit1`.`RegTime`,

`Limit1`.`RegisterID`,

`Limit1`.`RegisterName`,

`Limit1`.`ObservationDeptID`,

`Limit1`.`ObservationDeptName`,

`Limit1`.`ObservationDate`,

`Limit1`.`ObservationEndDate`,

`Limit1`.`ObservationLocation`,

`Limit1`.`ObservationEquipmentID`,

`Limit1`.`ObservationEquipment`,

`Limit1`.`EquipmentModel`,

`Limit1`.`ObservationMethod`,

`Limit1`.`StudyInstanceUID`,

`Limit1`.`TechnicianID`,

`Limit1`.`TechnicianName`,

`Limit1`.`ResultAssistantID`,

`Limit1`.`ResultAssistantName`,

`Limit1`.`ResultAssistantPhone`,

`Limit1`.`ResultPrincipalID`,

`Limit1`.`ResultPrincipalName`,

`Limit1`.`ResultReviseID`,

`Limit1`.`ResultReviseName`,

`Limit1`.`PreliminaryDate`,

`Limit1`.`AuditDate`,

`Limit1`.`ReviseDate`,

`Limit1`.`ResultDate`,

`Limit1`.`ResultOrganizationID`,

`Limit1`.`ResultServiceCenterUID`,

`Limit1`.`ResultStatus`,

`Limit1`.`ResultStatusCode`,

`Limit1`.`ResultPrintCount`,

`Limit1`.`ResultPrintTime`,

`Limit1`.`AbnormalFlags`,

`Limit1`.`CriticalValue`,

`Limit1`.`InfectionName`,

`Limit1`.`PrivacyLevel`,

`Limit1`.`PrivacyLevelExt`,

`Limit1`.`Charges`,

`Limit1`.`Payments`,

`Limit1`.`PaymentsFlag`,

`Limit1`.`FilmCount`,

`Limit1`.`FilmNeed`,

`Limit1`.`HasImage`,

`Limit1`.`ImageLocation`,

`Limit1`.`ConsultStatus`,

`Limit1`.`ConsultRequestDate`,

`Limit1`.`ConsultEndDate`,

`Limit1`.`ArchiveDate`,

`Limit1`.`LastUpdateDate`,

`Limit1`.`DataSource`,

`Limit1`.`LockFlag`,

`Limit1`.`LockUserUID`,

`Limit1`.`LockReason`,

`Limit1`.`InWritingUserUID`,

`Limit1`.`MessageCount`,

`Limit1`.`UnProcessWorkflowCount`,

`Limit1`.`PushState`,

`Limit1`.`DeleteFlag`,

`Limit1`.`RowTimestamp`,

`Limit1`.`DiskSpace`,

`Limit1`.`IsSummary`,

`Limit1`.`IsRepeatExamSum`,

`Limit1`.`IsDiseaseAnalysisSum`,

`Limit1`.`CollectionMark`,

`Limit1`.`PatientID1`,

`Limit1`.`PIDAssigningAuthority1`,

`Limit1`.`IsMPI`,

`Limit1`.`PatientMasterID1`,

`Limit1`.`Password`,

`Limit1`.`CreateDate`,

`Limit1`.`CreateUserID`,

`Limit1`.`CreateUserName`,

`Limit1`.`CreateOrgnizationID`,

`Limit1`.`Name`,

`Limit1`.`NameSpell`,

`Limit1`.`MotherName`,

`Limit1`.`Sex`,

`Limit1`.`BirthDate`,

`Limit1`.`BirthPlace`,

`Limit1`.`Nation`,

`Limit1`.`Citizenship`,

`Limit1`.`MaritalStatus`,

`Limit1`.`IdentityType`,

`Limit1`.`IdentityID`,

`Limit1`.`IDCardNO`,

`Limit1`.`HealthCardNO`,

`Limit1`.`InsuranceType`,

`Limit1`.`InsuranceID`,

`Limit1`.`ContactPhoneNO`,

`Limit1`.`HomePhoneNO`,

`Limit1`.`BusinessPhoneNO`,

`Limit1`.`Email`,

`Limit1`.`AddressProvince`,

`Limit1`.`AddressCity`,

`Limit1`.`AddressDistrict`,

`Limit1`.`AddressStreet`,

`Limit1`.`AddressRoad`,

`Limit1`.`AddressDetail`,

`Limit1`.`Postalcode`,

`Limit1`.`Occupation`,

`Limit1`.`WorkUnit`,

`Limit1`.`Language`,

`Limit1`.`LastUpdateDate1`,

`Limit1`.`InHospitalFlag`,

`Limit1`.`Status`,

`Limit1`.`RowTimestamp1`,

`Limit1`.`VisitUID1`,

`Limit1`.`PatientID2`,

`Limit1`.`PIDAssigningAuthority2`,

`Limit1`.`PatientMasterID2`,

`Limit1`.`VisitID`,

`Limit1`.`AlternateVisitID`,

`Limit1`.`AssigningAuthority`,

`Limit1`.`PatientClass1`,

`Limit1`.`PatientType`,

`Limit1`.`MedRecNO`,

`Limit1`.`OutPatientNO`,

`Limit1`.`InPatientNO`,

`Limit1`.`CardType`,

`Limit1`.`CardNO`,

`Limit1`.`Age`,

`Limit1`.`AgeUnit`,

`Limit1`.`MaritalStatus1`,

`Limit1`.`PregnancyFlag`,

`Limit1`.`Occupation1`,

`Limit1`.`OrganizationID1`,

`Limit1`.`AdmitDate`,

`Limit1`.`ReAmissionFlag`,

`Limit1`.`AdmissionType`,

`Limit1`.`AdmitPatCondition`,

`Limit1`.`AdmitDeptID`,

`Limit1`.`AdmitDeptName`,

`Limit1`.`DeptID`,

`Limit1`.`DeptName`,

`Limit1`.`PointOfCare`,

`Limit1`.`PointOfCareID`,

`Limit1`.`Room`,

`Limit1`.`Bed`,

`Limit1`.`AdmittingDoctorID`,

`Limit1`.`AdmittingDoctorName`,

`Limit1`.`AttendingDoctorID`,

`Limit1`.`AttendingDoctorName`,

`Limit1`.`DirectorDoctorID`,

`Limit1`.`DirectorDoctorName`,

`Limit1`.`CurPatCondition`,

`Limit1`.`NursingGrade`,

`Limit1`.`DietType`,

`Limit1`.`IsHasPacemaker`,

`Limit1`.`PrimaryDiagnosis`,

`Limit1`.`DischargeDate`,

`Limit1`.`DischargeDisposition`,

`Limit1`.`TotalCharges`,

`Limit1`.`TotalPayments`,

`Limit1`.`PaymentMode`,

`Limit1`.`MedicalPaymentMode`,

`Limit1`.`LastUpdateDate2`,

`Limit1`.`DeleteFlag1`,

`Limit1`.`RowTimestamp2`,

`Limit1`.`C1`,

`Extent4`.`ResultSeq`,

`Extent4`.`ObservationUID` AS `ObservationUID1`,

`Extent4`.`SetID`,

`Extent4`.`ObservationID`,

`Extent4`.`ObservationCodeScheme`,

`Extent4`.`ObservationText`,

`Extent4`.`SubID`,

`Extent4`.`ValueTitle`,

`Extent4`.`ValueType`,

`Extent4`.`ValueText`,

`Extent4`.`ValueCode`,

`Extent4`.`ValueCodeSystem`,

`Extent4`.`ValueUnits`,

`Extent4`.`ResultStatus` AS `ResultStatus1`,

`Extent4`.`ResultStatusCode` AS `ResultStatusCode1`,

`Extent4`.`ReferencesRange`,

`Extent4`.`AbnormalFlags` AS `AbnormalFlags1`,

`Extent4`.`ObservationMethod` AS `ObservationMethod1`,

`Extent4`.`ObservationDate` AS `ObservationDate1`,

`Extent4`.`PreliminaryDate` AS `PreliminaryDate1`,

`Extent4`.`AuditDate` AS `AuditDate1`,

`Extent4`.`ResultPrincipalID` AS `ResultPrincipalID1`,

`Extent4`.`ResultPrincipalName` AS `ResultPrincipalName1`,

`Extent4`.`ResultAssistantID` AS `ResultAssistantID1`,

`Extent4`.`ResultAssistantName` AS `ResultAssistantName1`,

`Extent4`.`LowerLimit`,

`Extent4`.`UpperLimit`,

`Extent4`.`Memo`,

`Extent4`.`ParentResultSeq`,

`Extent4`.`ParentObservationID`,

`Extent4`.`LastUpdateDate` AS `LastUpdateDate3`,

`Extent4`.`DeleteFlag` AS `DeleteFlag2`,

`Extent4`.`RowTimestamp` AS `RowTimestamp3`,

CASE

WHEN (

`Extent4`.`ResultSeq` IS NULL

) THEN

(NULL)

ELSE

(1)

END AS `C2`

FROM

(

SELECT

`Project1`.`ObservationUID`,

`Project1`.`PatientID`,

`Project1`.`PIDAssigningAuthority`,

`Project1`.`PatientMasterID`,

`Project1`.`PatientClass`,

`Project1`.`VisitUID`,

`Project1`.`OrderUID`,

`Project1`.`OrganizationID`,

`Project1`.`OrganizationName`,

`Project1`.`SourceOrganizationID`,

`Project1`.`SourceOrganizationName`,

`Project1`.`ClinicInfoType`,

`Project1`.`PlacerOrderNO`,

`Project1`.`PlacerAssigningAuthority`,

`Project1`.`PlacerOrderDetailNO`,

`Project1`.`FillerOrderNO`,

`Project1`.`FillerAssigningAuthority`,

`Project1`.`FillerPatientID`,

`Project1`.`AccessionNumber`,

`Project1`.`ServiceID`,

`Project1`.`ServiceCodeScheme`,

`Project1`.`ServiceText`,

`Project1`.`ServiceIDInsur`,

`Project1`.`ServiceSectID`,

`Project1`.`ProcedureID`,

`Project1`.`ProcedureName`,

`Project1`.`ProviderID`,

`Project1`.`ProviderName`,

`Project1`.`ProviderPhone`,

`Project1`.`RequestDeptID`,

`Project1`.`RequestDeptName`,

`Project1`.`RequestOrgID`,

`Project1`.`RequestOrgName`,

`Project1`.`RequestedDate`,

`Project1`.`Reason`,

`Project1`.`Attention`,

`Project1`.`Symptom`,

`Project1`.`AdverseReaction`,

`Project1`.`ClinicDiagnosis`,

`Project1`.`RelevantClinicalInfo`,

`Project1`.`FastingFlag`,

`Project1`.`TransportationMode`,

`Project1`.`SpecimenID`,

`Project1`.`SpecimenName`,

`Project1`.`SpecimenType`,

`Project1`.`SpecimenSource`,

`Project1`.`CollectorID`,

`Project1`.`CollectorName`,

`Project1`.`CollectionDept`,

`Project1`.`CollectionMethod`,

`Project1`.`CollectionVolume`,

`Project1`.`SpecimenReceivedDate`,

`Project1`.`ScheduledDate`,

`Project1`.`ScheduledDateEnd`,

`Project1`.`RegTime`,

`Project1`.`RegisterID`,

`Project1`.`RegisterName`,

`Project1`.`ObservationDeptID`,

`Project1`.`ObservationDeptName`,

`Project1`.`ObservationDate`,

`Project1`.`ObservationEndDate`,

`Project1`.`ObservationLocation`,

`Project1`.`ObservationEquipmentID`,

`Project1`.`ObservationEquipment`,

`Project1`.`EquipmentModel`,

`Project1`.`ObservationMethod`,

`Project1`.`StudyInstanceUID`,

`Project1`.`TechnicianID`,

`Project1`.`TechnicianName`,

`Project1`.`ResultAssistantID`,

`Project1`.`ResultAssistantName`,

`Project1`.`ResultAssistantPhone`,

`Project1`.`ResultPrincipalID`,

`Project1`.`ResultPrincipalName`,

`Project1`.`ResultReviseID`,

`Project1`.`ResultReviseName`,

`Project1`.`PreliminaryDate`,

`Project1`.`AuditDate`,

`Project1`.`ReviseDate`,

`Project1`.`ResultDate`,

`Project1`.`ResultOrganizationID`,

`Project1`.`ResultServiceCenterUID`,

`Project1`.`ResultStatus`,

`Project1`.`ResultStatusCode`,

`Project1`.`ResultPrintCount`,

`Project1`.`ResultPrintTime`,

`Project1`.`AbnormalFlags`,

`Project1`.`CriticalValue`,

`Project1`.`InfectionName`,

`Project1`.`PrivacyLevel`,

`Project1`.`PrivacyLevelExt`,

`Project1`.`Charges`,

`Project1`.`Payments`,

`Project1`.`PaymentsFlag`,

`Project1`.`FilmCount`,

`Project1`.`FilmNeed`,

`Project1`.`HasImage`,

`Project1`.`ImageLocation`,

`Project1`.`ConsultStatus`,

`Project1`.`ConsultRequestDate`,

`Project1`.`ConsultEndDate`,

`Project1`.`ArchiveDate`,

`Project1`.`LastUpdateDate`,

`Project1`.`DataSource`,

`Project1`.`LockFlag`,

`Project1`.`LockUserUID`,

`Project1`.`LockReason`,

`Project1`.`InWritingUserUID`,

`Project1`.`MessageCount`,

`Project1`.`UnProcessWorkflowCount`,

`Project1`.`PushState`,

`Project1`.`DeleteFlag`,

`Project1`.`RowTimestamp`,

`Project1`.`DiskSpace`,

`Project1`.`IsSummary`,

`Project1`.`IsRepeatExamSum`,

`Project1`.`IsDiseaseAnalysisSum`,

`Project1`.`CollectionMark`,

`Project1`.`PatientID1`,

`Project1`.`PIDAssigningAuthority1`,

`Project1`.`IsMPI`,

`Project1`.`PatientMasterID1`,

`Project1`.`Password`,

`Project1`.`CreateDate`,

`Project1`.`CreateUserID`,

`Project1`.`CreateUserName`,

`Project1`.`CreateOrgnizationID`,

`Project1`.`Name`,

`Project1`.`NameSpell`,

`Project1`.`MotherName`,

`Project1`.`Sex`,

`Project1`.`BirthDate`,

`Project1`.`BirthPlace`,

`Project1`.`Nation`,

`Project1`.`Citizenship`,

`Project1`.`MaritalStatus`,

`Project1`.`IdentityType`,

`Project1`.`IdentityID`,

`Project1`.`IDCardNO`,

`Project1`.`HealthCardNO`,

`Project1`.`InsuranceType`,

`Project1`.`InsuranceID`,

`Project1`.`ContactPhoneNO`,

`Project1`.`HomePhoneNO`,

`Project1`.`BusinessPhoneNO`,

`Project1`.`Email`,

`Project1`.`AddressProvince`,

`Project1`.`AddressCity`,

`Project1`.`AddressDistrict`,

`Project1`.`AddressStreet`,

`Project1`.`AddressRoad`,

`Project1`.`AddressDetail`,

`Project1`.`Postalcode`,

`Project1`.`Occupation`,

`Project1`.`WorkUnit`,

`Project1`.`Language`,

`Project1`.`LastUpdateDate1`,

`Project1`.`InHospitalFlag`,

`Project1`.`Status`,

`Project1`.`RowTimestamp1`,

`Project1`.`VisitUID1`,

`Project1`.`PatientID2`,

`Project1`.`PIDAssigningAuthority2`,

`Project1`.`PatientMasterID2`,

`Project1`.`VisitID`,

`Project1`.`AlternateVisitID`,

`Project1`.`AssigningAuthority`,

`Project1`.`PatientClass1`,

`Project1`.`PatientType`,

`Project1`.`MedRecNO`,

`Project1`.`OutPatientNO`,

`Project1`.`InPatientNO`,

`Project1`.`CardType`,

`Project1`.`CardNO`,

`Project1`.`Age`,

`Project1`.`AgeUnit`,

`Project1`.`MaritalStatus1`,

`Project1`.`PregnancyFlag`,

`Project1`.`Occupation1`,

`Project1`.`OrganizationID1`,

`Project1`.`AdmitDate`,

`Project1`.`ReAmissionFlag`,

`Project1`.`AdmissionType`,

`Project1`.`AdmitPatCondition`,

`Project1`.`AdmitDeptID`,

`Project1`.`AdmitDeptName`,

`Project1`.`DeptID`,

`Project1`.`DeptName`,

`Project1`.`PointOfCare`,

`Project1`.`PointOfCareID`,

`Project1`.`Room`,

`Project1`.`Bed`,

`Project1`.`AdmittingDoctorID`,

`Project1`.`AdmittingDoctorName`,

`Project1`.`AttendingDoctorID`,

`Project1`.`AttendingDoctorName`,

`Project1`.`DirectorDoctorID`,

`Project1`.`DirectorDoctorName`,

`Project1`.`CurPatCondition`,

`Project1`.`NursingGrade`,

`Project1`.`DietType`,

`Project1`.`IsHasPacemaker`,

`Project1`.`PrimaryDiagnosis`,

`Project1`.`DischargeDate`,

`Project1`.`DischargeDisposition`,

`Project1`.`TotalCharges`,

`Project1`.`TotalPayments`,

`Project1`.`PaymentMode`,

`Project1`.`MedicalPaymentMode`,

`Project1`.`LastUpdateDate2`,

`Project1`.`DeleteFlag1`,

`Project1`.`RowTimestamp2`,

`Project1`.`C1`

FROM

(

SELECT

`Filter1`.`ObservationUID`,

`Filter1`.`PatientID`,

`Filter1`.`PIDAssigningAuthority`,

`Filter1`.`PatientMasterID`,

`Filter1`.`PatientClass`,

`Filter1`.`VisitUID`,

`Filter1`.`OrderUID`,

`Filter1`.`OrganizationID`,

`Filter1`.`OrganizationName`,

`Filter1`.`SourceOrganizationID`,

`Filter1`.`SourceOrganizationName`,

`Filter1`.`ClinicInfoType`,

`Filter1`.`PlacerOrderNO`,

`Filter1`.`PlacerAssigningAuthority`,

`Filter1`.`PlacerOrderDetailNO`,

`Filter1`.`FillerOrderNO`,

`Filter1`.`FillerAssigningAuthority`,

`Filter1`.`FillerPatientID`,

`Filter1`.`AccessionNumber`,

`Filter1`.`ServiceID`,

`Filter1`.`ServiceCodeScheme`,

`Filter1`.`ServiceText`,

`Filter1`.`ServiceIDInsur`,

`Filter1`.`ServiceSectID`,

`Filter1`.`ProcedureID`,

`Filter1`.`ProcedureName`,

`Filter1`.`ProviderID`,

`Filter1`.`ProviderName`,

`Filter1`.`ProviderPhone`,

`Filter1`.`RequestDeptID`,

`Filter1`.`RequestDeptName`,

`Filter1`.`RequestOrgID`,

`Filter1`.`RequestOrgName`,

`Filter1`.`RequestedDate`,

`Filter1`.`Reason`,

`Filter1`.`Attention`,

`Filter1`.`Symptom`,

`Filter1`.`AdverseReaction`,

`Filter1`.`ClinicDiagnosis`,

`Filter1`.`RelevantClinicalInfo`,

`Filter1`.`FastingFlag`,

`Filter1`.`TransportationMode`,

`Filter1`.`SpecimenID`,

`Filter1`.`SpecimenName`,

`Filter1`.`SpecimenType`,

`Filter1`.`SpecimenSource`,

`Filter1`.`CollectorID`,

`Filter1`.`CollectorName`,

`Filter1`.`CollectionDept`,

`Filter1`.`CollectionMethod`,

`Filter1`.`CollectionVolume`,

`Filter1`.`SpecimenReceivedDate`,

`Filter1`.`ScheduledDate`,

`Filter1`.`ScheduledDateEnd`,

`Filter1`.`RegTime`,

`Filter1`.`RegisterID`,

`Filter1`.`RegisterName`,

`Filter1`.`ObservationDeptID`,

`Filter1`.`ObservationDeptName`,

`Filter1`.`ObservationDate`,

`Filter1`.`ObservationEndDate`,

`Filter1`.`ObservationLocation`,

`Filter1`.`ObservationEquipmentID`,

`Filter1`.`ObservationEquipment`,

`Filter1`.`EquipmentModel`,

`Filter1`.`ObservationMethod`,

`Filter1`.`StudyInstanceUID`,

`Filter1`.`TechnicianID`,

`Filter1`.`TechnicianName`,

`Filter1`.`ResultAssistantID`,

`Filter1`.`ResultAssistantName`,

`Filter1`.`ResultAssistantPhone`,

`Filter1`.`ResultPrincipalID`,

`Filter1`.`ResultPrincipalName`,

`Filter1`.`ResultReviseID`,

`Filter1`.`ResultReviseName`,

`Filter1`.`PreliminaryDate`,

`Filter1`.`AuditDate`,

`Filter1`.`ReviseDate`,

`Filter1`.`ResultDate`,

`Filter1`.`ResultOrganizationID`,

`Filter1`.`ResultServiceCenterUID`,

`Filter1`.`ResultStatus`,

`Filter1`.`ResultStatusCode`,

`Filter1`.`ResultPrintCount`,

`Filter1`.`ResultPrintTime`,

`Filter1`.`AbnormalFlags`,

`Filter1`.`CriticalValue`,

`Filter1`.`InfectionName`,

`Filter1`.`PrivacyLevel`,

`Filter1`.`PrivacyLevelExt`,

`Filter1`.`Charges`,

`Filter1`.`Payments`,

`Filter1`.`PaymentsFlag`,

`Filter1`.`FilmCount`,

`Filter1`.`FilmNeed`,

`Filter1`.`HasImage`,

`Filter1`.`ImageLocation`,

`Filter1`.`ConsultStatus`,

`Filter1`.`ConsultRequestDate`,

`Filter1`.`ConsultEndDate`,

`Filter1`.`ArchiveDate`,

`Filter1`.`LastUpdateDate`,

`Filter1`.`DataSource`,

`Filter1`.`LockFlag`,

`Filter1`.`LockUserUID`,

`Filter1`.`LockReason`,

`Filter1`.`InWritingUserUID`,

`Filter1`.`MessageCount`,

`Filter1`.`UnProcessWorkflowCount`,

`Filter1`.`PushState`,

`Filter1`.`DeleteFlag`,

`Filter1`.`RowTimestamp`,

`Filter1`.`DiskSpace`,

`Filter1`.`IsSummary`,

`Filter1`.`IsRepeatExamSum`,

`Filter1`.`IsDiseaseAnalysisSum`,

`Filter1`.`CollectionMark`,

`Filter1`.`PATIENTID1` AS `PatientID1`,

`Filter1`.`PIDASSIGNINGAUTHORITY1` AS `PIDAssigningAuthority1`,

`Filter1`.`IsMPI`,

`Filter1`.`PATIENTMASTERID1` AS `PatientMasterID1`,

`Filter1`.`Password`,

`Filter1`.`CreateDate`,

`Filter1`.`CreateUserID`,

`Filter1`.`CreateUserName`,

`Filter1`.`CreateOrgnizationID`,

`Filter1`.`Name`,

`Filter1`.`NameSpell`,

`Filter1`.`MotherName`,

`Filter1`.`Sex`,

`Filter1`.`BirthDate`,

`Filter1`.`BirthPlace`,

`Filter1`.`Nation`,

`Filter1`.`Citizenship`,

`Filter1`.`MaritalStatus`,

`Filter1`.`IdentityType`,

`Filter1`.`IdentityID`,

`Filter1`.`IDCardNO`,

`Filter1`.`HealthCardNO`,

`Filter1`.`InsuranceType`,

`Filter1`.`InsuranceID`,

`Filter1`.`ContactPhoneNO`,

`Filter1`.`HomePhoneNO`,

`Filter1`.`BusinessPhoneNO`,

`Filter1`.`Email`,

`Filter1`.`AddressProvince`,

`Filter1`.`AddressCity`,

`Filter1`.`AddressDistrict`,

`Filter1`.`AddressStreet`,

`Filter1`.`AddressRoad`,

`Filter1`.`AddressDetail`,

`Filter1`.`Postalcode`,

`Filter1`.`Occupation`,

`Filter1`.`WorkUnit`,

`Filter1`.`Language`,

`Filter1`.`LASTUPDATEDATE1` AS `LastUpdateDate1`,

`Filter1`.`InHospitalFlag`,

`Filter1`.`Status`,

`Filter1`.`ROWTIMESTAMP1` AS `RowTimestamp1`,

`Extent3`.`VisitUID` AS `VisitUID1`,

`Extent3`.`PatientID` AS `PatientID2`,

`Extent3`.`PIDAssigningAuthority` AS `PIDAssigningAuthority2`,

`Extent3`.`PatientMasterID` AS `PatientMasterID2`,

`Extent3`.`VisitID`,

`Extent3`.`AlternateVisitID`,

`Extent3`.`AssigningAuthority`,

`Extent3`.`PatientClass` AS `PatientClass1`,

`Extent3`.`PatientType`,

`Extent3`.`MedRecNO`,

`Extent3`.`OutPatientNO`,

`Extent3`.`InPatientNO`,

`Extent3`.`CardType`,

`Extent3`.`CardNO`,

`Extent3`.`Age`,

`Extent3`.`AgeUnit`,

`Extent3`.`MaritalStatus` AS `MaritalStatus1`,

`Extent3`.`PregnancyFlag`,

`Extent3`.`Occupation` AS `Occupation1`,

`Extent3`.`OrganizationID` AS `OrganizationID1`,

`Extent3`.`AdmitDate`,

`Extent3`.`ReAmissionFlag`,

`Extent3`.`AdmissionType`,

`Extent3`.`AdmitPatCondition`,

`Extent3`.`AdmitDeptID`,

`Extent3`.`AdmitDeptName`,

`Extent3`.`DeptID`,

`Extent3`.`DeptName`,

`Extent3`.`PointOfCare`,

`Extent3`.`PointOfCareID`,

`Extent3`.`Room`,

`Extent3`.`Bed`,

`Extent3`.`AdmittingDoctorID`,

`Extent3`.`AdmittingDoctorName`,

`Extent3`.`AttendingDoctorID`,

`Extent3`.`AttendingDoctorName`,

`Extent3`.`DirectorDoctorID`,

`Extent3`.`DirectorDoctorName`,

`Extent3`.`CurPatCondition`,

`Extent3`.`NursingGrade`,

`Extent3`.`DietType`,

`Extent3`.`IsHasPacemaker`,

`Extent3`.`PrimaryDiagnosis`,

`Extent3`.`DischargeDate`,

`Extent3`.`DischargeDisposition`,

`Extent3`.`TotalCharges`,

`Extent3`.`TotalPayments`,

`Extent3`.`PaymentMode`,

`Extent3`.`MedicalPaymentMode`,

`Extent3`.`LastUpdateDate` AS `LastUpdateDate2`,

`Extent3`.`DeleteFlag` AS `DeleteFlag1`,

`Extent3`.`RowTimestamp` AS `RowTimestamp2`,

1 AS `C1`

FROM

(

SELECT

`Extent1`.`ObservationUID`,

`Extent1`.`PatientID`,

`Extent1`.`PIDAssigningAuthority`,

`Extent1`.`PatientMasterID`,

`Extent1`.`PatientClass`,

`Extent1`.`VisitUID`,

`Extent1`.`OrderUID`,

`Extent1`.`OrganizationID`,

`Extent1`.`OrganizationName`,

`Extent1`.`SourceOrganizationID`,

`Extent1`.`SourceOrganizationName`,

`Extent1`.`ClinicInfoType`,

`Extent1`.`PlacerOrderNO`,

`Extent1`.`PlacerAssigningAuthority`,

`Extent1`.`PlacerOrderDetailNO`,

`Extent1`.`FillerOrderNO`,

`Extent1`.`FillerAssigningAuthority`,

`Extent1`.`FillerPatientID`,

`Extent1`.`AccessionNumber`,

`Extent1`.`ServiceID`,

`Extent1`.`ServiceCodeScheme`,

`Extent1`.`ServiceText`,

`Extent1`.`ServiceIDInsur`,

`Extent1`.`ServiceSectID`,

`Extent1`.`ProcedureID`,

`Extent1`.`ProcedureName`,

`Extent1`.`ProviderID`,

`Extent1`.`ProviderName`,

`Extent1`.`ProviderPhone`,

`Extent1`.`RequestDeptID`,

`Extent1`.`RequestDeptName`,

`Extent1`.`RequestOrgID`,

`Extent1`.`RequestOrgName`,

`Extent1`.`RequestedDate`,

`Extent1`.`Reason`,

`Extent1`.`Attention`,

`Extent1`.`Symptom`,

`Extent1`.`AdverseReaction`,

`Extent1`.`ClinicDiagnosis`,

`Extent1`.`RelevantClinicalInfo`,

`Extent1`.`FastingFlag`,

`Extent1`.`TransportationMode`,

`Extent1`.`SpecimenID`,

`Extent1`.`SpecimenName`,

`Extent1`.`SpecimenType`,

`Extent1`.`SpecimenSource`,

`Extent1`.`CollectorID`,

`Extent1`.`CollectorName`,

`Extent1`.`CollectionDept`,

`Extent1`.`CollectionMethod`,

`Extent1`.`CollectionVolume`,

`Extent1`.`SpecimenReceivedDate`,

`Extent1`.`ScheduledDate`,

`Extent1`.`ScheduledDateEnd`,

`Extent1`.`RegTime`,

`Extent1`.`RegisterID`,

`Extent1`.`RegisterName`,

`Extent1`.`ObservationDeptID`,

`Extent1`.`ObservationDeptName`,

`Extent1`.`ObservationDate`,

`Extent1`.`ObservationEndDate`,

`Extent1`.`ObservationLocation`,

`Extent1`.`ObservationEquipmentID`,

`Extent1`.`ObservationEquipment`,

`Extent1`.`EquipmentModel`,

`Extent1`.`ObservationMethod`,

`Extent1`.`StudyInstanceUID`,

`Extent1`.`TechnicianID`,

`Extent1`.`TechnicianName`,

`Extent1`.`ResultAssistantID`,

`Extent1`.`ResultAssistantName`,

`Extent1`.`ResultAssistantPhone`,

`Extent1`.`ResultPrincipalID`,

`Extent1`.`ResultPrincipalName`,

`Extent1`.`ResultReviseID`,

`Extent1`.`ResultReviseName`,

`Extent1`.`PreliminaryDate`,

`Extent1`.`AuditDate`,

`Extent1`.`ReviseDate`,

`Extent1`.`ResultDate`,

`Extent1`.`ResultOrganizationID`,

`Extent1`.`ResultServiceCenterUID`,

`Extent1`.`ResultStatus`,

`Extent1`.`ResultStatusCode`,

`Extent1`.`ResultPrintCount`,

`Extent1`.`ResultPrintTime`,

`Extent1`.`AbnormalFlags`,

`Extent1`.`CriticalValue`,

`Extent1`.`InfectionName`,

`Extent1`.`PrivacyLevel`,

`Extent1`.`PrivacyLevelExt`,

`Extent1`.`Charges`,

`Extent1`.`Payments`,

`Extent1`.`PaymentsFlag`,

`Extent1`.`FilmCount`,

`Extent1`.`FilmNeed`,

`Extent1`.`HasImage`,

`Extent1`.`ImageLocation`,

`Extent1`.`ConsultStatus`,

`Extent1`.`ConsultRequestDate`,

`Extent1`.`ConsultEndDate`,

`Extent1`.`ArchiveDate`,

`Extent1`.`LastUpdateDate`,

`Extent1`.`DataSource`,

`Extent1`.`LockFlag`,

`Extent1`.`LockUserUID`,

`Extent1`.`LockReason`,

`Extent1`.`InWritingUserUID`,

`Extent1`.`MessageCount`,

`Extent1`.`UnProcessWorkflowCount`,

`Extent1`.`PushState`,

`Extent1`.`DeleteFlag`,

`Extent1`.`RowTimestamp`,

`Extent1`.`DiskSpace`,

`Extent1`.`IsSummary`,

`Extent1`.`IsRepeatExamSum`,

`Extent1`.`IsDiseaseAnalysisSum`,

`Extent1`.`CollectionMark`,

`Extent2`.`PatientID` AS `PATIENTID1`,

`Extent2`.`PIDAssigningAuthority` AS `PIDASSIGNINGAUTHORITY1`,

`Extent2`.`IsMPI`,

`Extent2`.`PatientMasterID` AS `PATIENTMASTERID1`,

`Extent2`.`Password`,

`Extent2`.`CreateDate`,

`Extent2`.`CreateUserID`,

`Extent2`.`CreateUserName`,

`Extent2`.`CreateOrgnizationID`,

`Extent2`.`Name`,

`Extent2`.`NameSpell`,

`Extent2`.`MotherName`,

`Extent2`.`Sex`,

`Extent2`.`BirthDate`,

`Extent2`.`BirthPlace`,

`Extent2`.`Nation`,

`Extent2`.`Citizenship`,

`Extent2`.`MaritalStatus`,

`Extent2`.`IdentityType`,

`Extent2`.`IdentityID`,

`Extent2`.`IDCardNO`,

`Extent2`.`HealthCardNO`,

`Extent2`.`InsuranceType`,

`Extent2`.`InsuranceID`,

`Extent2`.`ContactPhoneNO`,

`Extent2`.`HomePhoneNO`,

`Extent2`.`BusinessPhoneNO`,

`Extent2`.`Email`,

`Extent2`.`AddressProvince`,

`Extent2`.`AddressCity`,

`Extent2`.`AddressDistrict`,

`Extent2`.`AddressStreet`,

`Extent2`.`AddressRoad`,

`Extent2`.`AddressDetail`,

`Extent2`.`Postalcode`,

`Extent2`.`Occupation`,

`Extent2`.`WorkUnit`,

`Extent2`.`Language`,

`Extent2`.`LastUpdateDate` AS `LASTUPDATEDATE1`,

`Extent2`.`InHospitalFlag`,

`Extent2`.`Status`,

`Extent2`.`RowTimestamp` AS `ROWTIMESTAMP1`

FROM

`ObservationRequest` AS `Extent1`

INNER JOIN `PatientIndex` AS `Extent2` ON (

`Extent1`.`PatientID` = `Extent2`.`PatientID`

)

AND (

`Extent1`.`PIDAssigningAuthority` = `Extent2`.`PIDAssigningAuthority`

)

WHERE

(0 = `Extent1`.`DeleteFlag`)

AND (

(

`Extent1`.`CollectionMark` NOT IN (1111, 1144, 1141, 1114)

)

OR (

`Extent1`.`CollectionMark` IS NULL

)

)

) AS `Filter1`

INNER JOIN `Visit` AS `Extent3` ON `Filter1`.`VisitUID` = `Extent3`.`VisitUID`

WHERE

(0 = `Extent3`.`DeleteFlag`)

AND (

`Filter1`.`OrganizationID` = @p\_\_linq\_\_0

)

) AS `Project1`

ORDER BY

`Project1`.`ObservationDate` DESC

LIMIT 120

) AS `Limit1`

LEFT OUTER JOIN `ObservationResult` AS `Extent4` ON (

`Extent4`.`ObservationUID` = `Limit1`.`ObservationUID`

)

AND (0 = `Extent4`.`DeleteFlag`)

) AS `Project2`

ORDER BY

`Project2`.`ObservationDate` DESC,

`Project2`.`ObservationUID` ASC,

`Project2`.`PatientID1` ASC,

`Project2`.`PIDAssigningAuthority1` ASC,

`Project2`.`VisitUID1` ASC,

`Project2`.`C2` ASC

2）改成在sql中查询检查结果就ToList();

SELECT

`Project2`.`C1`,

`Project2`.`ObservationUID`,

`Project2`.`PatientID`,

`Project2`.`PIDAssigningAuthority`,

`Project2`.`PatientMasterID`,

`Project2`.`PatientClass`,

`Project2`.`VisitUID`,

`Project2`.`OrderUID`,

`Project2`.`OrganizationID`,

`Project2`.`OrganizationName`,

`Project2`.`SourceOrganizationID`,

`Project2`.`SourceOrganizationName`,

`Project2`.`ClinicInfoType`,

`Project2`.`PlacerOrderNO`,

`Project2`.`PlacerAssigningAuthority`,

`Project2`.`PlacerOrderDetailNO`,

`Project2`.`FillerOrderNO`,

`Project2`.`FillerAssigningAuthority`,

`Project2`.`FillerPatientID`,

`Project2`.`AccessionNumber`,

`Project2`.`ServiceID`,

`Project2`.`ServiceCodeScheme`,

`Project2`.`ServiceText`,

`Project2`.`ServiceIDInsur`,

`Project2`.`ServiceSectID`,

`Project2`.`ProcedureID`,

`Project2`.`ProcedureName`,

`Project2`.`ProviderID`,

`Project2`.`ProviderName`,

`Project2`.`ProviderPhone`,

`Project2`.`RequestDeptID`,

`Project2`.`RequestDeptName`,

`Project2`.`RequestOrgID`,

`Project2`.`RequestOrgName`,

`Project2`.`RequestedDate`,

`Project2`.`Reason`,

`Project2`.`Attention`,

`Project2`.`Symptom`,

`Project2`.`AdverseReaction`,

`Project2`.`ClinicDiagnosis`,

`Project2`.`RelevantClinicalInfo`,

`Project2`.`FastingFlag`,

`Project2`.`TransportationMode`,

`Project2`.`SpecimenID`,

`Project2`.`SpecimenName`,

`Project2`.`SpecimenType`,

`Project2`.`SpecimenSource`,

`Project2`.`CollectorID`,

`Project2`.`CollectorName`,

`Project2`.`CollectionDept`,

`Project2`.`CollectionMethod`,

`Project2`.`CollectionVolume`,

`Project2`.`SpecimenReceivedDate`,

`Project2`.`ScheduledDate`,

`Project2`.`ScheduledDateEnd`,

`Project2`.`RegTime`,

`Project2`.`RegisterID`,

`Project2`.`RegisterName`,

`Project2`.`ObservationDeptID`,

`Project2`.`ObservationDeptName`,

`Project2`.`ObservationDate`,

`Project2`.`ObservationEndDate`,

`Project2`.`ObservationLocation`,

`Project2`.`ObservationEquipmentID`,

`Project2`.`ObservationEquipment`,

`Project2`.`EquipmentModel`,

`Project2`.`ObservationMethod`,

`Project2`.`StudyInstanceUID`,

`Project2`.`TechnicianID`,

`Project2`.`TechnicianName`,

`Project2`.`ResultAssistantID`,

`Project2`.`ResultAssistantName`,

`Project2`.`ResultAssistantPhone`,

`Project2`.`ResultPrincipalID`,

`Project2`.`ResultPrincipalName`,

`Project2`.`ResultReviseID`,

`Project2`.`ResultReviseName`,

`Project2`.`PreliminaryDate`,

`Project2`.`AuditDate`,

`Project2`.`ReviseDate`,

`Project2`.`ResultDate`,

`Project2`.`ResultOrganizationID`,

`Project2`.`ResultServiceCenterUID`,

`Project2`.`ResultStatus`,

`Project2`.`ResultStatusCode`,

`Project2`.`ResultPrintCount`,

`Project2`.`ResultPrintTime`,

`Project2`.`AbnormalFlags`,

`Project2`.`CriticalValue`,

`Project2`.`InfectionName`,

`Project2`.`PrivacyLevel`,

`Project2`.`PrivacyLevelExt`,

`Project2`.`Charges`,

`Project2`.`Payments`,

`Project2`.`PaymentsFlag`,

`Project2`.`FilmCount`,

`Project2`.`FilmNeed`,

`Project2`.`HasImage`,

`Project2`.`ImageLocation`,

`Project2`.`ConsultStatus`,

`Project2`.`ConsultRequestDate`,

`Project2`.`ConsultEndDate`,

`Project2`.`ArchiveDate`,

`Project2`.`LastUpdateDate`,

`Project2`.`DataSource`,

`Project2`.`LockFlag`,

`Project2`.`LockUserUID`,

`Project2`.`LockReason`,

`Project2`.`InWritingUserUID`,

`Project2`.`MessageCount`,

`Project2`.`UnProcessWorkflowCount`,

`Project2`.`PushState`,

`Project2`.`DeleteFlag`,

`Project2`.`RowTimestamp`,

`Project2`.`DiskSpace`,

`Project2`.`IsSummary`,

`Project2`.`IsRepeatExamSum`,

`Project2`.`IsDiseaseAnalysisSum`,

`Project2`.`CollectionMark`,

`Project2`.`PatientID1`,

`Project2`.`PIDAssigningAuthority1`,

`Project2`.`IsMPI`,

`Project2`.`PatientMasterID1`,

`Project2`.`Password`,

`Project2`.`CreateDate`,

`Project2`.`CreateUserID`,

`Project2`.`CreateUserName`,

`Project2`.`CreateOrgnizationID`,

`Project2`.`Name`,

`Project2`.`NameSpell`,

`Project2`.`MotherName`,

`Project2`.`Sex`,

`Project2`.`BirthDate`,

`Project2`.`BirthPlace`,

`Project2`.`Nation`,

`Project2`.`Citizenship`,

`Project2`.`MaritalStatus`,

`Project2`.`IdentityType`,

`Project2`.`IdentityID`,

`Project2`.`IDCardNO`,

`Project2`.`HealthCardNO`,

`Project2`.`InsuranceType`,

`Project2`.`InsuranceID`,

`Project2`.`ContactPhoneNO`,

`Project2`.`HomePhoneNO`,

`Project2`.`BusinessPhoneNO`,

`Project2`.`Email`,

`Project2`.`AddressProvince`,

`Project2`.`AddressCity`,

`Project2`.`AddressDistrict`,

`Project2`.`AddressStreet`,

`Project2`.`AddressRoad`,

`Project2`.`AddressDetail`,

`Project2`.`Postalcode`,

`Project2`.`Occupation`,

`Project2`.`WorkUnit`,

`Project2`.`Language`,

`Project2`.`LastUpdateDate1`,

`Project2`.`InHospitalFlag`,

`Project2`.`Status`,

`Project2`.`RowTimestamp1`,

`Project2`.`VisitUID1`,

`Project2`.`PatientID2`,

`Project2`.`PIDAssigningAuthority2`,

`Project2`.`PatientMasterID2`,

`Project2`.`VisitID`,

`Project2`.`AlternateVisitID`,

`Project2`.`AssigningAuthority`,

`Project2`.`PatientClass1`,

`Project2`.`PatientType`,

`Project2`.`MedRecNO`,

`Project2`.`OutPatientNO`,

`Project2`.`InPatientNO`,

`Project2`.`CardType`,

`Project2`.`CardNO`,

`Project2`.`Age`,

`Project2`.`AgeUnit`,

`Project2`.`MaritalStatus1`,

`Project2`.`PregnancyFlag`,

`Project2`.`Occupation1`,

`Project2`.`OrganizationID1`,

`Project2`.`AdmitDate`,

`Project2`.`ReAmissionFlag`,

`Project2`.`AdmissionType`,

`Project2`.`AdmitPatCondition`,

`Project2`.`AdmitDeptID`,

`Project2`.`AdmitDeptName`,

`Project2`.`DeptID`,

`Project2`.`DeptName`,

`Project2`.`PointOfCare`,

`Project2`.`PointOfCareID`,

`Project2`.`Room`,

`Project2`.`Bed`,

`Project2`.`AdmittingDoctorID`,

`Project2`.`AdmittingDoctorName`,

`Project2`.`AttendingDoctorID`,

`Project2`.`AttendingDoctorName`,

`Project2`.`DirectorDoctorID`,

`Project2`.`DirectorDoctorName`,

`Project2`.`CurPatCondition`,

`Project2`.`NursingGrade`,

`Project2`.`DietType`,

`Project2`.`IsHasPacemaker`,

`Project2`.`PrimaryDiagnosis`,

`Project2`.`DischargeDate`,

`Project2`.`DischargeDisposition`,

`Project2`.`TotalCharges`,

`Project2`.`TotalPayments`,

`Project2`.`PaymentMode`,

`Project2`.`MedicalPaymentMode`,

`Project2`.`LastUpdateDate2`,

`Project2`.`DeleteFlag1`,

`Project2`.`RowTimestamp2`,

`Project2`.`C2`,

`Project2`.`ResultSeq`,

`Project2`.`ObservationUID1`,

`Project2`.`SetID`,

`Project2`.`ObservationID`,

`Project2`.`ObservationCodeScheme`,

`Project2`.`ObservationText`,

`Project2`.`SubID`,

`Project2`.`ValueTitle`,

`Project2`.`ValueType`,

`Project2`.`ValueText`,

`Project2`.`ValueCode`,

`Project2`.`ValueCodeSystem`,

`Project2`.`ValueUnits`,

`Project2`.`ResultStatus1`,

`Project2`.`ResultStatusCode1`,

`Project2`.`ReferencesRange`,

`Project2`.`AbnormalFlags1`,

`Project2`.`ObservationMethod1`,

`Project2`.`ObservationDate1`,

`Project2`.`PreliminaryDate1`,

`Project2`.`AuditDate1`,

`Project2`.`ResultPrincipalID1`,

`Project2`.`ResultPrincipalName1`,

`Project2`.`ResultAssistantID1`,

`Project2`.`ResultAssistantName1`,

`Project2`.`LowerLimit`,

`Project2`.`UpperLimit`,

`Project2`.`Memo`,

`Project2`.`ParentResultSeq`,

`Project2`.`ParentObservationID`,

`Project2`.`LastUpdateDate3`,

`Project2`.`DeleteFlag2`,

`Project2`.`RowTimestamp3`

FROM

(

SELECT

`Limit1`.`ObservationUID`,

`Limit1`.`PatientID`,

`Limit1`.`PIDAssigningAuthority`,

`Limit1`.`PatientMasterID`,

`Limit1`.`PatientClass`,

`Limit1`.`VisitUID`,

`Limit1`.`OrderUID`,

`Limit1`.`OrganizationID`,

`Limit1`.`OrganizationName`,

`Limit1`.`SourceOrganizationID`,

`Limit1`.`SourceOrganizationName`,

`Limit1`.`ClinicInfoType`,

`Limit1`.`PlacerOrderNO`,

`Limit1`.`PlacerAssigningAuthority`,

`Limit1`.`PlacerOrderDetailNO`,

`Limit1`.`FillerOrderNO`,

`Limit1`.`FillerAssigningAuthority`,

`Limit1`.`FillerPatientID`,

`Limit1`.`AccessionNumber`,

`Limit1`.`ServiceID`,

`Limit1`.`ServiceCodeScheme`,

`Limit1`.`ServiceText`,

`Limit1`.`ServiceIDInsur`,

`Limit1`.`ServiceSectID`,

`Limit1`.`ProcedureID`,

`Limit1`.`ProcedureName`,

`Limit1`.`ProviderID`,

`Limit1`.`ProviderName`,

`Limit1`.`ProviderPhone`,

`Limit1`.`RequestDeptID`,

`Limit1`.`RequestDeptName`,

`Limit1`.`RequestOrgID`,

`Limit1`.`RequestOrgName`,

`Limit1`.`RequestedDate`,

`Limit1`.`Reason`,

`Limit1`.`Attention`,

`Limit1`.`Symptom`,

`Limit1`.`AdverseReaction`,

`Limit1`.`ClinicDiagnosis`,

`Limit1`.`RelevantClinicalInfo`,

`Limit1`.`FastingFlag`,

`Limit1`.`TransportationMode`,

`Limit1`.`SpecimenID`,

`Limit1`.`SpecimenName`,

`Limit1`.`SpecimenType`,

`Limit1`.`SpecimenSource`,

`Limit1`.`CollectorID`,

`Limit1`.`CollectorName`,

`Limit1`.`CollectionDept`,

`Limit1`.`CollectionMethod`,

`Limit1`.`CollectionVolume`,

`Limit1`.`SpecimenReceivedDate`,

`Limit1`.`ScheduledDate`,

`Limit1`.`ScheduledDateEnd`,

`Limit1`.`RegTime`,

`Limit1`.`RegisterID`,

`Limit1`.`RegisterName`,

`Limit1`.`ObservationDeptID`,

`Limit1`.`ObservationDeptName`,

`Limit1`.`ObservationDate`,

`Limit1`.`ObservationEndDate`,

`Limit1`.`ObservationLocation`,

`Limit1`.`ObservationEquipmentID`,

`Limit1`.`ObservationEquipment`,

`Limit1`.`EquipmentModel`,

`Limit1`.`ObservationMethod`,

`Limit1`.`StudyInstanceUID`,

`Limit1`.`TechnicianID`,

`Limit1`.`TechnicianName`,

`Limit1`.`ResultAssistantID`,

`Limit1`.`ResultAssistantName`,

`Limit1`.`ResultAssistantPhone`,

`Limit1`.`ResultPrincipalID`,

`Limit1`.`ResultPrincipalName`,

`Limit1`.`ResultReviseID`,

`Limit1`.`ResultReviseName`,

`Limit1`.`PreliminaryDate`,

`Limit1`.`AuditDate`,

`Limit1`.`ReviseDate`,

`Limit1`.`ResultDate`,

`Limit1`.`ResultOrganizationID`,

`Limit1`.`ResultServiceCenterUID`,

`Limit1`.`ResultStatus`,

`Limit1`.`ResultStatusCode`,

`Limit1`.`ResultPrintCount`,

`Limit1`.`ResultPrintTime`,

`Limit1`.`AbnormalFlags`,

`Limit1`.`CriticalValue`,

`Limit1`.`InfectionName`,

`Limit1`.`PrivacyLevel`,

`Limit1`.`PrivacyLevelExt`,

`Limit1`.`Charges`,

`Limit1`.`Payments`,

`Limit1`.`PaymentsFlag`,

`Limit1`.`FilmCount`,

`Limit1`.`FilmNeed`,

`Limit1`.`HasImage`,

`Limit1`.`ImageLocation`,

`Limit1`.`ConsultStatus`,

`Limit1`.`ConsultRequestDate`,

`Limit1`.`ConsultEndDate`,

`Limit1`.`ArchiveDate`,

`Limit1`.`LastUpdateDate`,

`Limit1`.`DataSource`,

`Limit1`.`LockFlag`,

`Limit1`.`LockUserUID`,

`Limit1`.`LockReason`,

`Limit1`.`InWritingUserUID`,

`Limit1`.`MessageCount`,

`Limit1`.`UnProcessWorkflowCount`,

`Limit1`.`PushState`,

`Limit1`.`DeleteFlag`,

`Limit1`.`RowTimestamp`,

`Limit1`.`DiskSpace`,

`Limit1`.`IsSummary`,

`Limit1`.`IsRepeatExamSum`,

`Limit1`.`IsDiseaseAnalysisSum`,

`Limit1`.`CollectionMark`,

`Limit1`.`PatientID1`,

`Limit1`.`PIDAssigningAuthority1`,

`Limit1`.`IsMPI`,

`Limit1`.`PatientMasterID1`,

`Limit1`.`Password`,

`Limit1`.`CreateDate`,

`Limit1`.`CreateUserID`,

`Limit1`.`CreateUserName`,

`Limit1`.`CreateOrgnizationID`,

`Limit1`.`Name`,

`Limit1`.`NameSpell`,

`Limit1`.`MotherName`,

`Limit1`.`Sex`,

`Limit1`.`BirthDate`,

`Limit1`.`BirthPlace`,

`Limit1`.`Nation`,

`Limit1`.`Citizenship`,

`Limit1`.`MaritalStatus`,

`Limit1`.`IdentityType`,

`Limit1`.`IdentityID`,

`Limit1`.`IDCardNO`,

`Limit1`.`HealthCardNO`,

`Limit1`.`InsuranceType`,

`Limit1`.`InsuranceID`,

`Limit1`.`ContactPhoneNO`,

`Limit1`.`HomePhoneNO`,

`Limit1`.`BusinessPhoneNO`,

`Limit1`.`Email`,

`Limit1`.`AddressProvince`,

`Limit1`.`AddressCity`,

`Limit1`.`AddressDistrict`,

`Limit1`.`AddressStreet`,

`Limit1`.`AddressRoad`,

`Limit1`.`AddressDetail`,

`Limit1`.`Postalcode`,

`Limit1`.`Occupation`,

`Limit1`.`WorkUnit`,

`Limit1`.`Language`,

`Limit1`.`LastUpdateDate1`,

`Limit1`.`InHospitalFlag`,

`Limit1`.`Status`,

`Limit1`.`RowTimestamp1`,

`Limit1`.`VisitUID1`,

`Limit1`.`PatientID2`,

`Limit1`.`PIDAssigningAuthority2`,

`Limit1`.`PatientMasterID2`,

`Limit1`.`VisitID`,

`Limit1`.`AlternateVisitID`,

`Limit1`.`AssigningAuthority`,

`Limit1`.`PatientClass1`,

`Limit1`.`PatientType`,

`Limit1`.`MedRecNO`,

`Limit1`.`OutPatientNO`,

`Limit1`.`InPatientNO`,

`Limit1`.`CardType`,

`Limit1`.`CardNO`,

`Limit1`.`Age`,

`Limit1`.`AgeUnit`,

`Limit1`.`MaritalStatus1`,

`Limit1`.`PregnancyFlag`,

`Limit1`.`Occupation1`,

`Limit1`.`OrganizationID1`,

`Limit1`.`AdmitDate`,

`Limit1`.`ReAmissionFlag`,

`Limit1`.`AdmissionType`,

`Limit1`.`AdmitPatCondition`,

`Limit1`.`AdmitDeptID`,

`Limit1`.`AdmitDeptName`,

`Limit1`.`DeptID`,

`Limit1`.`DeptName`,

`Limit1`.`PointOfCare`,

`Limit1`.`PointOfCareID`,

`Limit1`.`Room`,

`Limit1`.`Bed`,

`Limit1`.`AdmittingDoctorID`,

`Limit1`.`AdmittingDoctorName`,

`Limit1`.`AttendingDoctorID`,

`Limit1`.`AttendingDoctorName`,

`Limit1`.`DirectorDoctorID`,

`Limit1`.`DirectorDoctorName`,

`Limit1`.`CurPatCondition`,

`Limit1`.`NursingGrade`,

`Limit1`.`DietType`,

`Limit1`.`IsHasPacemaker`,

`Limit1`.`PrimaryDiagnosis`,

`Limit1`.`DischargeDate`,

`Limit1`.`DischargeDisposition`,

`Limit1`.`TotalCharges`,

`Limit1`.`TotalPayments`,

`Limit1`.`PaymentMode`,

`Limit1`.`MedicalPaymentMode`,

`Limit1`.`LastUpdateDate2`,

`Limit1`.`DeleteFlag1`,

`Limit1`.`RowTimestamp2`,

`Limit1`.`C1`,

`Extent4`.`ResultSeq`,

`Extent4`.`ObservationUID` AS `ObservationUID1`,

`Extent4`.`SetID`,

`Extent4`.`ObservationID`,

`Extent4`.`ObservationCodeScheme`,

`Extent4`.`ObservationText`,

`Extent4`.`SubID`,

`Extent4`.`ValueTitle`,

`Extent4`.`ValueType`,

`Extent4`.`ValueText`,

`Extent4`.`ValueCode`,

`Extent4`.`ValueCodeSystem`,

`Extent4`.`ValueUnits`,

`Extent4`.`ResultStatus` AS `ResultStatus1`,

`Extent4`.`ResultStatusCode` AS `ResultStatusCode1`,

`Extent4`.`ReferencesRange`,

`Extent4`.`AbnormalFlags` AS `AbnormalFlags1`,

`Extent4`.`ObservationMethod` AS `ObservationMethod1`,

`Extent4`.`ObservationDate` AS `ObservationDate1`,

`Extent4`.`PreliminaryDate` AS `PreliminaryDate1`,

`Extent4`.`AuditDate` AS `AuditDate1`,

`Extent4`.`ResultPrincipalID` AS `ResultPrincipalID1`,

`Extent4`.`ResultPrincipalName` AS `ResultPrincipalName1`,

`Extent4`.`ResultAssistantID` AS `ResultAssistantID1`,

`Extent4`.`ResultAssistantName` AS `ResultAssistantName1`,

`Extent4`.`LowerLimit`,

`Extent4`.`UpperLimit`,

`Extent4`.`Memo`,

`Extent4`.`ParentResultSeq`,

`Extent4`.`ParentObservationID`,

`Extent4`.`LastUpdateDate` AS `LastUpdateDate3`,

`Extent4`.`DeleteFlag` AS `DeleteFlag2`,

`Extent4`.`RowTimestamp` AS `RowTimestamp3`,

CASE

WHEN (

`Extent4`.`ResultSeq` IS NULL

) THEN

(NULL)

ELSE

(1)

END AS `C2`

FROM

(

SELECT

`Project1`.`ObservationUID`,

`Project1`.`PatientID`,

`Project1`.`PIDAssigningAuthority`,

`Project1`.`PatientMasterID`,

`Project1`.`PatientClass`,

`Project1`.`VisitUID`,

`Project1`.`OrderUID`,

`Project1`.`OrganizationID`,

`Project1`.`OrganizationName`,

`Project1`.`SourceOrganizationID`,

`Project1`.`SourceOrganizationName`,

`Project1`.`ClinicInfoType`,

`Project1`.`PlacerOrderNO`,

`Project1`.`PlacerAssigningAuthority`,

`Project1`.`PlacerOrderDetailNO`,

`Project1`.`FillerOrderNO`,

`Project1`.`FillerAssigningAuthority`,

`Project1`.`FillerPatientID`,

`Project1`.`AccessionNumber`,

`Project1`.`ServiceID`,

`Project1`.`ServiceCodeScheme`,

`Project1`.`ServiceText`,

`Project1`.`ServiceIDInsur`,

`Project1`.`ServiceSectID`,

`Project1`.`ProcedureID`,

`Project1`.`ProcedureName`,

`Project1`.`ProviderID`,

`Project1`.`ProviderName`,

`Project1`.`ProviderPhone`,

`Project1`.`RequestDeptID`,

`Project1`.`RequestDeptName`,

`Project1`.`RequestOrgID`,

`Project1`.`RequestOrgName`,

`Project1`.`RequestedDate`,

`Project1`.`Reason`,

`Project1`.`Attention`,

`Project1`.`Symptom`,

`Project1`.`AdverseReaction`,

`Project1`.`ClinicDiagnosis`,

`Project1`.`RelevantClinicalInfo`,

`Project1`.`FastingFlag`,

`Project1`.`TransportationMode`,

`Project1`.`SpecimenID`,

`Project1`.`SpecimenName`,

`Project1`.`SpecimenType`,

`Project1`.`SpecimenSource`,

`Project1`.`CollectorID`,

`Project1`.`CollectorName`,

`Project1`.`CollectionDept`,

`Project1`.`CollectionMethod`,

`Project1`.`CollectionVolume`,

`Project1`.`SpecimenReceivedDate`,

`Project1`.`ScheduledDate`,

`Project1`.`ScheduledDateEnd`,

`Project1`.`RegTime`,

`Project1`.`RegisterID`,

`Project1`.`RegisterName`,

`Project1`.`ObservationDeptID`,

`Project1`.`ObservationDeptName`,

`Project1`.`ObservationDate`,

`Project1`.`ObservationEndDate`,

`Project1`.`ObservationLocation`,

`Project1`.`ObservationEquipmentID`,

`Project1`.`ObservationEquipment`,

`Project1`.`EquipmentModel`,

`Project1`.`ObservationMethod`,

`Project1`.`StudyInstanceUID`,

`Project1`.`TechnicianID`,

`Project1`.`TechnicianName`,

`Project1`.`ResultAssistantID`,

`Project1`.`ResultAssistantName`,

`Project1`.`ResultAssistantPhone`,

`Project1`.`ResultPrincipalID`,

`Project1`.`ResultPrincipalName`,

`Project1`.`ResultReviseID`,

`Project1`.`ResultReviseName`,

`Project1`.`PreliminaryDate`,

`Project1`.`AuditDate`,

`Project1`.`ReviseDate`,

`Project1`.`ResultDate`,

`Project1`.`ResultOrganizationID`,

`Project1`.`ResultServiceCenterUID`,

`Project1`.`ResultStatus`,

`Project1`.`ResultStatusCode`,

`Project1`.`ResultPrintCount`,

`Project1`.`ResultPrintTime`,

`Project1`.`AbnormalFlags`,

`Project1`.`CriticalValue`,

`Project1`.`InfectionName`,

`Project1`.`PrivacyLevel`,

`Project1`.`PrivacyLevelExt`,

`Project1`.`Charges`,

`Project1`.`Payments`,

`Project1`.`PaymentsFlag`,

`Project1`.`FilmCount`,

`Project1`.`FilmNeed`,

`Project1`.`HasImage`,

`Project1`.`ImageLocation`,

`Project1`.`ConsultStatus`,

`Project1`.`ConsultRequestDate`,

`Project1`.`ConsultEndDate`,

`Project1`.`ArchiveDate`,

`Project1`.`LastUpdateDate`,

`Project1`.`DataSource`,

`Project1`.`LockFlag`,

`Project1`.`LockUserUID`,

`Project1`.`LockReason`,

`Project1`.`InWritingUserUID`,

`Project1`.`MessageCount`,

`Project1`.`UnProcessWorkflowCount`,

`Project1`.`PushState`,

`Project1`.`DeleteFlag`,

`Project1`.`RowTimestamp`,

`Project1`.`DiskSpace`,

`Project1`.`IsSummary`,

`Project1`.`IsRepeatExamSum`,

`Project1`.`IsDiseaseAnalysisSum`,

`Project1`.`CollectionMark`,

`Project1`.`PatientID1`,

`Project1`.`PIDAssigningAuthority1`,

`Project1`.`IsMPI`,

`Project1`.`PatientMasterID1`,

`Project1`.`Password`,

`Project1`.`CreateDate`,

`Project1`.`CreateUserID`,

`Project1`.`CreateUserName`,

`Project1`.`CreateOrgnizationID`,

`Project1`.`Name`,

`Project1`.`NameSpell`,

`Project1`.`MotherName`,

`Project1`.`Sex`,

`Project1`.`BirthDate`,

`Project1`.`BirthPlace`,

`Project1`.`Nation`,

`Project1`.`Citizenship`,

`Project1`.`MaritalStatus`,

`Project1`.`IdentityType`,

`Project1`.`IdentityID`,

`Project1`.`IDCardNO`,

`Project1`.`HealthCardNO`,

`Project1`.`InsuranceType`,

`Project1`.`InsuranceID`,

`Project1`.`ContactPhoneNO`,

`Project1`.`HomePhoneNO`,

`Project1`.`BusinessPhoneNO`,

`Project1`.`Email`,

`Project1`.`AddressProvince`,

`Project1`.`AddressCity`,

`Project1`.`AddressDistrict`,

`Project1`.`AddressStreet`,

`Project1`.`AddressRoad`,

`Project1`.`AddressDetail`,

`Project1`.`Postalcode`,

`Project1`.`Occupation`,

`Project1`.`WorkUnit`,

`Project1`.`Language`,

`Project1`.`LastUpdateDate1`,

`Project1`.`InHospitalFlag`,

`Project1`.`Status`,

`Project1`.`RowTimestamp1`,

`Project1`.`VisitUID1`,

`Project1`.`PatientID2`,

`Project1`.`PIDAssigningAuthority2`,

`Project1`.`PatientMasterID2`,

`Project1`.`VisitID`,

`Project1`.`AlternateVisitID`,

`Project1`.`AssigningAuthority`,

`Project1`.`PatientClass1`,

`Project1`.`PatientType`,

`Project1`.`MedRecNO`,

`Project1`.`OutPatientNO`,

`Project1`.`InPatientNO`,

`Project1`.`CardType`,

`Project1`.`CardNO`,

`Project1`.`Age`,

`Project1`.`AgeUnit`,

`Project1`.`MaritalStatus1`,

`Project1`.`PregnancyFlag`,

`Project1`.`Occupation1`,

`Project1`.`OrganizationID1`,

`Project1`.`AdmitDate`,

`Project1`.`ReAmissionFlag`,

`Project1`.`AdmissionType`,

`Project1`.`AdmitPatCondition`,

`Project1`.`AdmitDeptID`,

`Project1`.`AdmitDeptName`,

`Project1`.`DeptID`,

`Project1`.`DeptName`,

`Project1`.`PointOfCare`,

`Project1`.`PointOfCareID`,

`Project1`.`Room`,

`Project1`.`Bed`,

`Project1`.`AdmittingDoctorID`,

`Project1`.`AdmittingDoctorName`,

`Project1`.`AttendingDoctorID`,

`Project1`.`AttendingDoctorName`,

`Project1`.`DirectorDoctorID`,

`Project1`.`DirectorDoctorName`,

`Project1`.`CurPatCondition`,

`Project1`.`NursingGrade`,

`Project1`.`DietType`,

`Project1`.`IsHasPacemaker`,

`Project1`.`PrimaryDiagnosis`,

`Project1`.`DischargeDate`,

`Project1`.`DischargeDisposition`,

`Project1`.`TotalCharges`,

`Project1`.`TotalPayments`,

`Project1`.`PaymentMode`,

`Project1`.`MedicalPaymentMode`,

`Project1`.`LastUpdateDate2`,

`Project1`.`DeleteFlag1`,

`Project1`.`RowTimestamp2`,

`Project1`.`C1`

FROM

(

SELECT

`Filter1`.`ObservationUID`,

`Filter1`.`PatientID`,

`Filter1`.`PIDAssigningAuthority`,

`Filter1`.`PatientMasterID`,

`Filter1`.`PatientClass`,

`Filter1`.`VisitUID`,

`Filter1`.`OrderUID`,

`Filter1`.`OrganizationID`,

`Filter1`.`OrganizationName`,

`Filter1`.`SourceOrganizationID`,

`Filter1`.`SourceOrganizationName`,

`Filter1`.`ClinicInfoType`,

`Filter1`.`PlacerOrderNO`,

`Filter1`.`PlacerAssigningAuthority`,

`Filter1`.`PlacerOrderDetailNO`,

`Filter1`.`FillerOrderNO`,

`Filter1`.`FillerAssigningAuthority`,

`Filter1`.`FillerPatientID`,

`Filter1`.`AccessionNumber`,

`Filter1`.`ServiceID`,

`Filter1`.`ServiceCodeScheme`,

`Filter1`.`ServiceText`,

`Filter1`.`ServiceIDInsur`,

`Filter1`.`ServiceSectID`,

`Filter1`.`ProcedureID`,

`Filter1`.`ProcedureName`,

`Filter1`.`ProviderID`,

`Filter1`.`ProviderName`,

`Filter1`.`ProviderPhone`,

`Filter1`.`RequestDeptID`,

`Filter1`.`RequestDeptName`,

`Filter1`.`RequestOrgID`,

`Filter1`.`RequestOrgName`,

`Filter1`.`RequestedDate`,

`Filter1`.`Reason`,

`Filter1`.`Attention`,

`Filter1`.`Symptom`,

`Filter1`.`AdverseReaction`,

`Filter1`.`ClinicDiagnosis`,

`Filter1`.`RelevantClinicalInfo`,

`Filter1`.`FastingFlag`,

`Filter1`.`TransportationMode`,

`Filter1`.`SpecimenID`,

`Filter1`.`SpecimenName`,

`Filter1`.`SpecimenType`,

`Filter1`.`SpecimenSource`,

`Filter1`.`CollectorID`,

`Filter1`.`CollectorName`,

`Filter1`.`CollectionDept`,

`Filter1`.`CollectionMethod`,

`Filter1`.`CollectionVolume`,

`Filter1`.`SpecimenReceivedDate`,

`Filter1`.`ScheduledDate`,

`Filter1`.`ScheduledDateEnd`,

`Filter1`.`RegTime`,

`Filter1`.`RegisterID`,

`Filter1`.`RegisterName`,

`Filter1`.`ObservationDeptID`,

`Filter1`.`ObservationDeptName`,

`Filter1`.`ObservationDate`,

`Filter1`.`ObservationEndDate`,

`Filter1`.`ObservationLocation`,

`Filter1`.`ObservationEquipmentID`,

`Filter1`.`ObservationEquipment`,

`Filter1`.`EquipmentModel`,

`Filter1`.`ObservationMethod`,

`Filter1`.`StudyInstanceUID`,

`Filter1`.`TechnicianID`,

`Filter1`.`TechnicianName`,

`Filter1`.`ResultAssistantID`,

`Filter1`.`ResultAssistantName`,

`Filter1`.`ResultAssistantPhone`,

`Filter1`.`ResultPrincipalID`,

`Filter1`.`ResultPrincipalName`,

`Filter1`.`ResultReviseID`,

`Filter1`.`ResultReviseName`,

`Filter1`.`PreliminaryDate`,

`Filter1`.`AuditDate`,

`Filter1`.`ReviseDate`,

`Filter1`.`ResultDate`,

`Filter1`.`ResultOrganizationID`,

`Filter1`.`ResultServiceCenterUID`,

`Filter1`.`ResultStatus`,

`Filter1`.`ResultStatusCode`,

`Filter1`.`ResultPrintCount`,

`Filter1`.`ResultPrintTime`,

`Filter1`.`AbnormalFlags`,

`Filter1`.`CriticalValue`,

`Filter1`.`InfectionName`,

`Filter1`.`PrivacyLevel`,

`Filter1`.`PrivacyLevelExt`,

`Filter1`.`Charges`,

`Filter1`.`Payments`,

`Filter1`.`PaymentsFlag`,

`Filter1`.`FilmCount`,

`Filter1`.`FilmNeed`,

`Filter1`.`HasImage`,

`Filter1`.`ImageLocation`,

`Filter1`.`ConsultStatus`,

`Filter1`.`ConsultRequestDate`,

`Filter1`.`ConsultEndDate`,

`Filter1`.`ArchiveDate`,

`Filter1`.`LastUpdateDate`,

`Filter1`.`DataSource`,

`Filter1`.`LockFlag`,

`Filter1`.`LockUserUID`,

`Filter1`.`LockReason`,

`Filter1`.`InWritingUserUID`,

`Filter1`.`MessageCount`,

`Filter1`.`UnProcessWorkflowCount`,

`Filter1`.`PushState`,

`Filter1`.`DeleteFlag`,

`Filter1`.`RowTimestamp`,

`Filter1`.`DiskSpace`,

`Filter1`.`IsSummary`,

`Filter1`.`IsRepeatExamSum`,

`Filter1`.`IsDiseaseAnalysisSum`,

`Filter1`.`CollectionMark`,

`Filter1`.`PATIENTID1` AS `PatientID1`,

`Filter1`.`PIDASSIGNINGAUTHORITY1` AS `PIDAssigningAuthority1`,

`Filter1`.`IsMPI`,

`Filter1`.`PATIENTMASTERID1` AS `PatientMasterID1`,

`Filter1`.`Password`,

`Filter1`.`CreateDate`,

`Filter1`.`CreateUserID`,

`Filter1`.`CreateUserName`,

`Filter1`.`CreateOrgnizationID`,

`Filter1`.`Name`,

`Filter1`.`NameSpell`,

`Filter1`.`MotherName`,

`Filter1`.`Sex`,

`Filter1`.`BirthDate`,

`Filter1`.`BirthPlace`,

`Filter1`.`Nation`,

`Filter1`.`Citizenship`,

`Filter1`.`MaritalStatus`,

`Filter1`.`IdentityType`,

`Filter1`.`IdentityID`,

`Filter1`.`IDCardNO`,

`Filter1`.`HealthCardNO`,

`Filter1`.`InsuranceType`,

`Filter1`.`InsuranceID`,

`Filter1`.`ContactPhoneNO`,

`Filter1`.`HomePhoneNO`,

`Filter1`.`BusinessPhoneNO`,

`Filter1`.`Email`,

`Filter1`.`AddressProvince`,

`Filter1`.`AddressCity`,

`Filter1`.`AddressDistrict`,

`Filter1`.`AddressStreet`,

`Filter1`.`AddressRoad`,

`Filter1`.`AddressDetail`,

`Filter1`.`Postalcode`,

`Filter1`.`Occupation`,

`Filter1`.`WorkUnit`,

`Filter1`.`Language`,

`Filter1`.`LASTUPDATEDATE1` AS `LastUpdateDate1`,

`Filter1`.`InHospitalFlag`,

`Filter1`.`Status`,

`Filter1`.`ROWTIMESTAMP1` AS `RowTimestamp1`,

`Extent3`.`VisitUID` AS `VisitUID1`,

`Extent3`.`PatientID` AS `PatientID2`,

`Extent3`.`PIDAssigningAuthority` AS `PIDAssigningAuthority2`,

`Extent3`.`PatientMasterID` AS `PatientMasterID2`,

`Extent3`.`VisitID`,

`Extent3`.`AlternateVisitID`,

`Extent3`.`AssigningAuthority`,

`Extent3`.`PatientClass` AS `PatientClass1`,

`Extent3`.`PatientType`,

`Extent3`.`MedRecNO`,

`Extent3`.`OutPatientNO`,

`Extent3`.`InPatientNO`,

`Extent3`.`CardType`,

`Extent3`.`CardNO`,

`Extent3`.`Age`,

`Extent3`.`AgeUnit`,

`Extent3`.`MaritalStatus` AS `MaritalStatus1`,

`Extent3`.`PregnancyFlag`,

`Extent3`.`Occupation` AS `Occupation1`,

`Extent3`.`OrganizationID` AS `OrganizationID1`,

`Extent3`.`AdmitDate`,

`Extent3`.`ReAmissionFlag`,

`Extent3`.`AdmissionType`,

`Extent3`.`AdmitPatCondition`,

`Extent3`.`AdmitDeptID`,

`Extent3`.`AdmitDeptName`,

`Extent3`.`DeptID`,

`Extent3`.`DeptName`,

`Extent3`.`PointOfCare`,

`Extent3`.`PointOfCareID`,

`Extent3`.`Room`,

`Extent3`.`Bed`,

`Extent3`.`AdmittingDoctorID`,

`Extent3`.`AdmittingDoctorName`,

`Extent3`.`AttendingDoctorID`,

`Extent3`.`AttendingDoctorName`,

`Extent3`.`DirectorDoctorID`,

`Extent3`.`DirectorDoctorName`,

`Extent3`.`CurPatCondition`,

`Extent3`.`NursingGrade`,

`Extent3`.`DietType`,

`Extent3`.`IsHasPacemaker`,

`Extent3`.`PrimaryDiagnosis`,

`Extent3`.`DischargeDate`,

`Extent3`.`DischargeDisposition`,

`Extent3`.`TotalCharges`,

`Extent3`.`TotalPayments`,

`Extent3`.`PaymentMode`,

`Extent3`.`MedicalPaymentMode`,

`Extent3`.`LastUpdateDate` AS `LastUpdateDate2`,

`Extent3`.`DeleteFlag` AS `DeleteFlag1`,

`Extent3`.`RowTimestamp` AS `RowTimestamp2`,

1 AS `C1`

FROM

(

SELECT

`Extent1`.`ObservationUID`,

`Extent1`.`PatientID`,

`Extent1`.`PIDAssigningAuthority`,

`Extent1`.`PatientMasterID`,

`Extent1`.`PatientClass`,

`Extent1`.`VisitUID`,

`Extent1`.`OrderUID`,

`Extent1`.`OrganizationID`,

`Extent1`.`OrganizationName`,

`Extent1`.`SourceOrganizationID`,

`Extent1`.`SourceOrganizationName`,

`Extent1`.`ClinicInfoType`,

`Extent1`.`PlacerOrderNO`,

`Extent1`.`PlacerAssigningAuthority`,

`Extent1`.`PlacerOrderDetailNO`,

`Extent1`.`FillerOrderNO`,

`Extent1`.`FillerAssigningAuthority`,

`Extent1`.`FillerPatientID`,

`Extent1`.`AccessionNumber`,

`Extent1`.`ServiceID`,

`Extent1`.`ServiceCodeScheme`,

`Extent1`.`ServiceText`,

`Extent1`.`ServiceIDInsur`,

`Extent1`.`ServiceSectID`,

`Extent1`.`ProcedureID`,

`Extent1`.`ProcedureName`,

`Extent1`.`ProviderID`,

`Extent1`.`ProviderName`,

`Extent1`.`ProviderPhone`,

`Extent1`.`RequestDeptID`,

`Extent1`.`RequestDeptName`,

`Extent1`.`RequestOrgID`,

`Extent1`.`RequestOrgName`,

`Extent1`.`RequestedDate`,

`Extent1`.`Reason`,

`Extent1`.`Attention`,

`Extent1`.`Symptom`,

`Extent1`.`AdverseReaction`,

`Extent1`.`ClinicDiagnosis`,

`Extent1`.`RelevantClinicalInfo`,

`Extent1`.`FastingFlag`,

`Extent1`.`TransportationMode`,

`Extent1`.`SpecimenID`,

`Extent1`.`SpecimenName`,

`Extent1`.`SpecimenType`,

`Extent1`.`SpecimenSource`,

`Extent1`.`CollectorID`,

`Extent1`.`CollectorName`,

`Extent1`.`CollectionDept`,

`Extent1`.`CollectionMethod`,

`Extent1`.`CollectionVolume`,

`Extent1`.`SpecimenReceivedDate`,

`Extent1`.`ScheduledDate`,

`Extent1`.`ScheduledDateEnd`,

`Extent1`.`RegTime`,

`Extent1`.`RegisterID`,

`Extent1`.`RegisterName`,

`Extent1`.`ObservationDeptID`,

`Extent1`.`ObservationDeptName`,

`Extent1`.`ObservationDate`,

`Extent1`.`ObservationEndDate`,

`Extent1`.`ObservationLocation`,

`Extent1`.`ObservationEquipmentID`,

`Extent1`.`ObservationEquipment`,

`Extent1`.`EquipmentModel`,

`Extent1`.`ObservationMethod`,

`Extent1`.`StudyInstanceUID`,

`Extent1`.`TechnicianID`,

`Extent1`.`TechnicianName`,

`Extent1`.`ResultAssistantID`,

`Extent1`.`ResultAssistantName`,

`Extent1`.`ResultAssistantPhone`,

`Extent1`.`ResultPrincipalID`,

`Extent1`.`ResultPrincipalName`,

`Extent1`.`ResultReviseID`,

`Extent1`.`ResultReviseName`,

`Extent1`.`PreliminaryDate`,

`Extent1`.`AuditDate`,

`Extent1`.`ReviseDate`,

`Extent1`.`ResultDate`,

`Extent1`.`ResultOrganizationID`,

`Extent1`.`ResultServiceCenterUID`,

`Extent1`.`ResultStatus`,

`Extent1`.`ResultStatusCode`,

`Extent1`.`ResultPrintCount`,

`Extent1`.`ResultPrintTime`,

`Extent1`.`AbnormalFlags`,

`Extent1`.`CriticalValue`,

`Extent1`.`InfectionName`,

`Extent1`.`PrivacyLevel`,

`Extent1`.`PrivacyLevelExt`,

`Extent1`.`Charges`,

`Extent1`.`Payments`,

`Extent1`.`PaymentsFlag`,

`Extent1`.`FilmCount`,

`Extent1`.`FilmNeed`,

`Extent1`.`HasImage`,

`Extent1`.`ImageLocation`,

`Extent1`.`ConsultStatus`,

`Extent1`.`ConsultRequestDate`,

`Extent1`.`ConsultEndDate`,

`Extent1`.`ArchiveDate`,

`Extent1`.`LastUpdateDate`,

`Extent1`.`DataSource`,

`Extent1`.`LockFlag`,

`Extent1`.`LockUserUID`,

`Extent1`.`LockReason`,

`Extent1`.`InWritingUserUID`,

`Extent1`.`MessageCount`,

`Extent1`.`UnProcessWorkflowCount`,

`Extent1`.`PushState`,

`Extent1`.`DeleteFlag`,

`Extent1`.`RowTimestamp`,

`Extent1`.`DiskSpace`,

`Extent1`.`IsSummary`,

`Extent1`.`IsRepeatExamSum`,

`Extent1`.`IsDiseaseAnalysisSum`,

`Extent1`.`CollectionMark`,

`Extent2`.`PatientID` AS `PATIENTID1`,

`Extent2`.`PIDAssigningAuthority` AS `PIDASSIGNINGAUTHORITY1`,

`Extent2`.`IsMPI`,

`Extent2`.`PatientMasterID` AS `PATIENTMASTERID1`,

`Extent2`.`Password`,

`Extent2`.`CreateDate`,

`Extent2`.`CreateUserID`,

`Extent2`.`CreateUserName`,

`Extent2`.`CreateOrgnizationID`,

`Extent2`.`Name`,

`Extent2`.`NameSpell`,

`Extent2`.`MotherName`,

`Extent2`.`Sex`,

`Extent2`.`BirthDate`,

`Extent2`.`BirthPlace`,

`Extent2`.`Nation`,

`Extent2`.`Citizenship`,

`Extent2`.`MaritalStatus`,

`Extent2`.`IdentityType`,

`Extent2`.`IdentityID`,

`Extent2`.`IDCardNO`,

`Extent2`.`HealthCardNO`,

`Extent2`.`InsuranceType`,

`Extent2`.`InsuranceID`,

`Extent2`.`ContactPhoneNO`,

`Extent2`.`HomePhoneNO`,

`Extent2`.`BusinessPhoneNO`,

`Extent2`.`Email`,

`Extent2`.`AddressProvince`,

`Extent2`.`AddressCity`,

`Extent2`.`AddressDistrict`,

`Extent2`.`AddressStreet`,

`Extent2`.`AddressRoad`,

`Extent2`.`AddressDetail`,

`Extent2`.`Postalcode`,

`Extent2`.`Occupation`,

`Extent2`.`WorkUnit`,

`Extent2`.`Language`,

`Extent2`.`LastUpdateDate` AS `LASTUPDATEDATE1`,

`Extent2`.`InHospitalFlag`,

`Extent2`.`Status`,

`Extent2`.`RowTimestamp` AS `ROWTIMESTAMP1`

FROM

`ObservationRequest` AS `Extent1`

INNER JOIN `PatientIndex` AS `Extent2` ON (

`Extent1`.`PatientID` = `Extent2`.`PatientID`

)

AND (

`Extent1`.`PIDAssigningAuthority` = `Extent2`.`PIDAssigningAuthority`

)

WHERE

(0 = `Extent1`.`DeleteFlag`)

AND (

(

`Extent1`.`CollectionMark` NOT IN (1111, 1144, 1141, 1114)

)

OR (

`Extent1`.`CollectionMark` IS NULL

)

)

) AS `Filter1`

INNER JOIN `Visit` AS `Extent3` ON `Filter1`.`VisitUID` = `Extent3`.`VisitUID`

WHERE

(0 = `Extent3`.`DeleteFlag`)

AND (

`Filter1`.`OrganizationID` = @p\_\_linq\_\_0

)

) AS `Project1`

ORDER BY

`Project1`.`ObservationDate` DESC

LIMIT 120

) AS `Limit1`

LEFT OUTER JOIN `ObservationResult` AS `Extent4` ON (

`Extent4`.`ObservationUID` = `Limit1`.`ObservationUID`

)

AND (0 = `Extent4`.`DeleteFlag`)

) AS `Project2`

ORDER BY

`Project2`.`ObservationDate` DESC,

`Project2`.`ObservationUID` ASC,

`Project2`.`PatientID1` ASC,

`Project2`.`PIDAssigningAuthority1` ASC,

`Project2`.`VisitUID1` ASC,

`Project2`.`C2` ASC

18.

背景：邵逸夫医院离线采集影像工具，需要根据影像的AccessionNumber;PatientId;Modality信息;

从归档平台获取检查的observationUID，存在影像文件的Modality;与平台上ServiceSectID不一致的情况，需要新增接口供离线采集工具使用;;

需要实现：邵逸夫医院离线采集影像工具，需要根据影像的AccessionNumber;PatientId;Modality信息;

从归档平台获取检查的observationUID，存在影像文件的Modality;与平台上ServiceSectID不一致的情况，需要新增接口供离线采集工具查询检查类型

GetExamInfoByOffline

19.

严重性 代码 说明 项目 文件 行 禁止显示状态

错误 CS0012 类型“Bitmap”在未引用的程序集中定义。必须添加对程序集“System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a”的引用。 eWorld.Plugin.WebAPI.CloudArchive E:\some\work\AbilityOP\trunk\OpenPlat\Plugins\eWorld.Plugin.WebAPI.CloudArchive\Controllers\PublicController.cs 501 活动

获取的地址：

<http://218.108.111.166:9200/IDCAS/8712b3ad5a422c4>

修改后：

<http://218.108.111.166:9200/IDCAS/Mobile/PatientExam/Index/c7cde3e331112914>

<http://115.236.163.163:9200/IDCAS/Mobile/PatientExam/Index/c7cde3e331112914>

http://115.236.163.163:9200/IDCAS/Mobile/PatientExam/Index/ed92c12b680c00f6

http://localhost:9200/Mobile/PatientExam/Index/c7cde3e331112914

20.

1.0.6的

21.

更改的地方:

1）

<!--影像云存储地址（开发测试用,发布不需要配置） -->

<add key="IDCASUrl" value="http://localhost:9300/" />

2）

// 最终地址

// \_imageArchiveClient = new eWorldWCFClient<IImageArchiveService>(WebServiceUrlManager.ImageArchiveWebServiceUrl);

// 调试地址

\_imageArchiveClient = new eWorldWCFClient<IImageArchiveService>("http://localhost:9301/Services/ImageArchiveService.svc");

22.

ImageArchiveDatabaseInfo.ConnectionString "server=192.168.10.79;port=3306;database=ImageArchive;uid=sa;password=tomtaw" string

知道原因了，初始化连接的工具不对。不是202上面的。

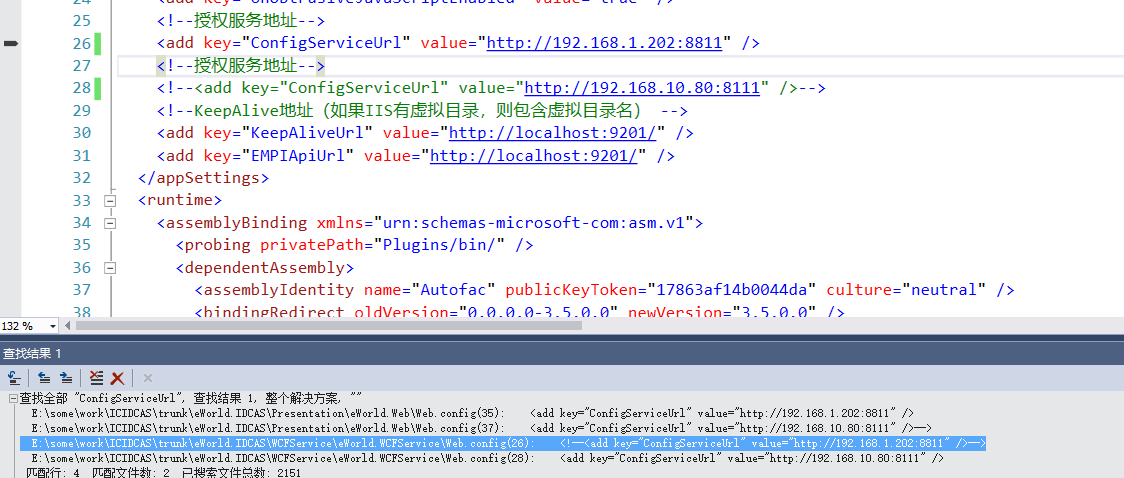
换回来了：

<!--授权服务地址-->

<add key="ConfigServiceUrl" value="http://192.168.1.202:8811" />

<!--授权服务地址--><!--

<add key="ConfigServiceUrl" value="http://192.168.10.80:8111" />-->



wcf和web的初始化配置都改回来了。

加了身份证，然后验证身份证后6位之后就成功了。

测试用户后六位：141448

23.

当不是管理员15，自己配置进去，会造成无法获取金牌机构数据。

是否要通过两个接口，先获取所有所属的机构列表，再来获取对应的机构信息？

24.

1）患者信息多余的就诊表的入参

2）

a.去掉了检查信息注册，会诊的入参。

b.删除所有预约相关的入参，因为无法去和别人去解释。

c.加了result里面的observation\_id扩展，还有检验。它在request中是叫lab，但这里的叫法。——GDT:检查所见、IMP:检查诊断、REC:诊断建议、ADT:诊断备注、ConsultAdvice:会诊意见、Gross:病理巨检、

d.检查改为观察，因为可能是有检验的。

25.

质量报告: GetExamStatistics

如果时间不传，默认是查询距离此刻近7天的数据。

if (query.observationEndDate == null)

{

query.observationEndDate = DateTime.Now;

}

if (query.observationStartDate == null)

{

query.observationStartDate = query.observationEndDate.Value.AddDays(-7);

}

查询的时间默认是存在影像的记录。

var querySql = from t in \_observationRequestRepository.Table

where t.DeleteFlag == false && t.HasImage.Value == true

select t;

新方法，优化之后：

/// <summary>

/// 质控检查统计数量

/// </summary>

/// <param name="query"></param>

/// <returns></returns>

public List<ExtractExamStatisticsDTO> GetExamStatistics(QualityQuery query)

{

Stopwatch sw = null;

var result = new List<ExtractExamStatisticsDTO>();

try

{

if (\_logDebug)

{

sw = new Stopwatch();

sw.Start();

\_log.Debug("收到GetPatientExamList方法调用，参数：" + JsonConvert.SerializeObject(query));

}

var observationList = QueryObservationRequest(query)

.Select(x => new

{

x.ServiceSectID,

x.ServiceText,

x.OrganizationID,

x.OrganizationName

});

if (!string.IsNullOrWhiteSpace(query.serviceText))

{

result = observationList.GroupBy(t => new

{

t.OrganizationID,

t.OrganizationName,

t.ServiceSectID,

t.ServiceText

}).Select(t => new ExtractExamStatisticsDTO

{

OrganizationCode = t.Key.OrganizationID,

OrganizationName = t.Key.OrganizationName,

ServiceSectID = t.Key.ServiceSectID,

ServiceText = t.Key.ServiceText,

Count = t.Count()

}).ToList();

}

else

{

result = observationList.GroupBy(t => new

{

t.OrganizationID,

t.OrganizationName,

t.ServiceSectID

}).Select(t => new ExtractExamStatisticsDTO

{

OrganizationCode = t.Key.OrganizationID,

OrganizationName = t.Key.OrganizationName,

ServiceSectID = t.Key.ServiceSectID,

Count = t.Count()

}).ToList();

}

return result;

}

catch (Exception ex)

{

\_log.Error(ex.Message, ex);

return result;

}

finally

{

if (\_logDebug)

{

sw.Stop();

\_log.Debug(string.Format("函数耗费时间:{0}ms", sw.ElapsedMilliseconds.ToString()));

}

}

}

修订后的速度，秒回



结果集：

{

"code": 0,

"msg": "成功",

"data": {

"DataList": [

{

"ServiceSectID": "CR",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1222

},

{

"ServiceSectID": "CT",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1194

},

{

"ServiceSectID": "DR",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1237

},

{

"ServiceSectID": "ES",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1205

},

{

"ServiceSectID": "MR",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1224

},

{

"ServiceSectID": "PS",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1244

},

{

"ServiceSectID": "RF",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1250

},

{

"ServiceSectID": "US",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1151

},

{

"ServiceSectID": "XA",

"ServiceText": null,

"OrganizationCode": "2402",

"OrganizationName": null,

"Count": 1218

}

]

}

}

26.

我修改区域影像，视图，这块的测试问题。

质量报告，自己写。

crm原来的data：

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "eWorld.Entity", "..\..\..\Data\branches\1.0.6.20190225\_mining\entity\eWorld.Entity\eWorld.Entity.csproj", "{BD066F4B-E1AB-423E-BC3D-B32688CE6EF9}"

27.

有待搜索

ClinicDataOperationRepository

28.



遗留问题，二维码的过期时间我还没有设定。

23.

http://192.168.1.202:9200/IDCAS/SharingVisits/Interface/ExamList?IDCardNO=FF941BE673023CB37A1ACAF2A38B33701F384A8AD2AA4CAE

24.

nb2ss.mtywcloud.com:9200/IDCAS/SharingVisits/Interface/ExamList?IDCardNO= F6A1B82B87A09E2F790F312A63DC12F0D5CABE6280BDDF97

nb2ss.mtywcloud.com:9200/IDCAS/SharingVisits/Interface/ExamList?IDCardNO=9B9CFB5F3FE911D25AA6052652CF9036A08D5802B844A0AF

http://115.231.181.5:9200/IDCAS/SharingVisits/Interface/ExamList?IDCardNO=9B9CFB5F3FE911D25AA6052652CF9036A08D5802B844A0AF

25.

var clinicinfotype = new List<string> { "Exam", "Lab" };

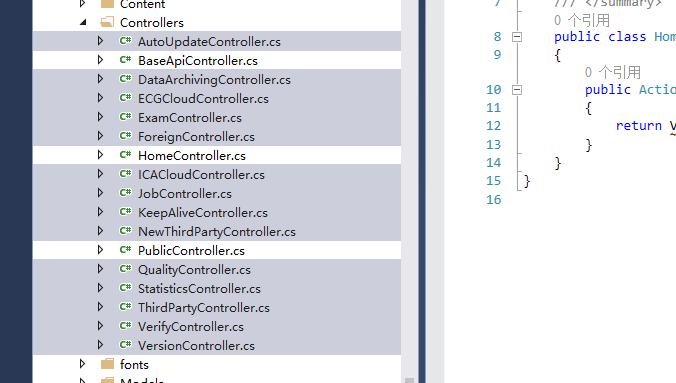
var observationRequestSql = from a in \_observationRequestRepository.Table

where a.DeleteFlag == false && (!mark.Contains(a.CollectionMark.Value) || !a.CollectionMark.HasValue) && clinicinfotype.Contains(a.ClinicInfoType)

26.

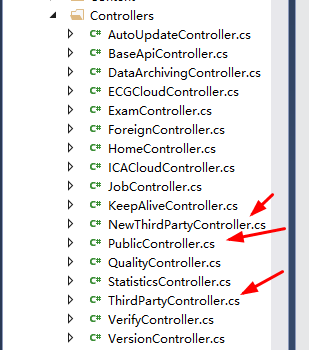
api修改，在分支里面。给app使用。

public和校验先提供出来。



这三个留着，接下来一个一个加入到项目中。

27.



就这三个，其中第三个只需要一个第三方的注册，给 xds就可以了。（也许是xml吧，也不知道）

// 当存在检查状态（这里的状态，app端传过来的实际是code）

if (!string.IsNullOrWhiteSpace(query.ResultStatus))

{

query.ResultStatus = HttpUtility.UrlDecode(query.ResultStatus).TrimEnd(',');

var resultStatusList = query.ResultStatus.Split(',');

if (resultStatusList.Length == 1)

{

observationRequestQuery = observationRequestQuery.Where(p => p.ResultStatusCode == query.ResultStatus);

}

else

{

observationRequestQuery = observationRequestQuery.Where(p => resultStatusList.Contains(p.ResultStatusCode));

}

}

28.

1）存储空间预警

2）归档短信发送给患者去调阅患者影像的

两个后台服务，1.0.6版本，web，task里面。把这两个任务挪到法爷那里，其余三个，兵哥已经挪过去了。

29.

开化，只给一个当天的时间，当天的数据给对方。

原来的api代码：

//foreach (var item in examList)

//{

// if (item.patientIndex != null && item.observationRequest != null && item.patientVisit != null)

// {

// // 数据的重组

// PatPacsResulDTO model = new PatPacsResulDTO()

// {

// OrganizationID = item.observationRequest.OrganizationID,

// Name = item.patientIndex.Name,

// IDCardNO = item.patientIndex.IDCardNO,

// InsuranceID = item.patientIndex.InsuranceID,

// CardNO = item.patientVisit.CardNO,

// MedRecNO = item.patientVisit.MedRecNO,

// VisitID = item.patientVisit.VisitID,

// PatientClass = item.observationRequest.PatientClass,

// RelevantClinicalInfo = item.observationRequest.RelevantClinicalInfo,

// OrderUID = item.observationRequest.OrderUID,

// RegisterID = item.observationRequest.RegisterID,

// RegTime = item.observationRequest.RegTime,

// RequestDeptID = item.observationRequest.RequestDeptID,

// ORDER\_USEFLAG = 1,

// PlacerOrderNO = item.observationRequest.PlacerOrderNO,

// AccessionNumber = item.observationRequest.AccessionNumber,

// ServiceID = item.observationRequest.ServiceID,

// ServiceText = item.observationRequest.ServiceText,

// ServiceSectID = item.observationRequest.ServiceSectID,

// ProcedureID = item.observationRequest.ProcedureID,

// ProcedureName = item.observationRequest.ProcedureName,

// Attention = item.observationRequest.Attention,

// CollectionDept = item.observationRequest.CollectionDept,

// ResultReviseName = item.observationRequest.ResultReviseName,

// AuditDate = item.observationRequest.AuditDate,

// ObservationDeptID = item.observationRequest.ObservationDeptID,

// TechnicianName = item.observationRequest.TechnicianName,

// ObservationDate = item.observationRequest.ObservationDate,

// EXE\_FLAG = 1,

// SpecimenName = item.observationRequest.SpecimenName,

// CollectorName = item.observationRequest.CollectorName,

// SpecimenReceivedDate = item.observationRequest.SpecimenReceivedDate,

// ResultPrincipalName = item.observationRequest.ResultPrincipalName,

// ResultDate = item.observationRequest.ResultDate,

// SourceOrganizationID = item.observationRequest.SourceOrganizationID,

// };

// // 就诊类别

// model.PatientClass = GetPatientClass(model.PatientClass);

// // 机构ID的转换

// //model.OrganizationID = hospitals.FirstOrDefault(x => x.Name == item.ObservationRequest.OrganizationName)?.Code;

// //model.SourceOrganizationID = hospitals.FirstOrDefault(x => x.Name == item.ObservationRequest.SourceOrganizationName)?.Code;

// model.OrganizationID = hospitals.FirstOrDefault(x => x.Name == item.ObservationRequest.OrganizationName)?.Code ?? item.ObservationRequest.OrganizationID;

// model.SourceOrganizationID = hospitals.FirstOrDefault(x => x.Name == item.ObservationRequest.SourceOrganizationName)?.Code ?? item.ObservationRequest.SourceOrganizationID;

// // 申请单号的过滤

// if (model.PlacerOrderNO.StartsWith("Z", StringComparison.OrdinalIgnoreCase) || model.PlacerOrderNO.StartsWith("M", StringComparison.OrdinalIgnoreCase))

// {

// model.PlacerOrderNO = model.PlacerOrderNO.Substring(1);

// }

// // 病历号的处理

// if (string.IsNullOrWhiteSpace(model.MedRecNO))

// {

// // 优先用门诊号代替

// if (!string.IsNullOrWhiteSpace(item.patientVisit.OutPatientNO))

// {

// model.MedRecNO = item.patientVisit.OutPatientNO;

// }

// else

// {

// if (!string.IsNullOrWhiteSpace(item.patientVisit.InPatientNO))

// {

// model.MedRecNO = item.patientVisit.InPatientNO;

// }

// else

// {

// // 跳过该条数据？

// model.MedRecNO = string.Empty;

// }

// }

// }

// if (item.observationResultList.Count > 0)

// {

// foreach (var dto in item.observationResultList)

// {

// //if (dto.ObservationID == "Gross")

// //{

// // patientExamDTO.GrossSight = item.ValueText;

// //}

// if (dto.ObservationID == "GDT")

// {

// model.GDTValueTitle = dto.ValueText;

// }

// if (dto.ObservationID == "IMP")

// {

// model.IMGValueTitle = dto.ValueText;

// }

// }

// }

// \_log.Debug("获取检查文档信息");

// // 文档

// var resultdata = new Core.BasicResult.ModelResult<List<Core.DTO.DocumentViewDTO>>();

// var documentQuery = new DocumentQuery() { BusinessID = item.observationRequest.ObservationUID.ToString() };

// resultdata = \_imageArchiveClient.Invoke(t => t.GetDocumentList(documentQuery));

// List<GetDocumentInfoDTO> examDocuments = new List<GetDocumentInfoDTO>();

// if (resultdata.code == ResultCodeEnum.Success)

// {

// examDocuments = resultdata.Result.ToListDto();

// var idcasurl = ConfigSettings.IDCASUrl;

// examDocuments.ForEach(p =>

// {

// p.FileBusinessViewUrl = CommonUtil.TransLocalhostForClient(

// p.FileBusinessViewUrl, idcasurl);

// });

// }

// // 文档

// //获取报告地址

// if (examDocuments.Any(d => d.TypeCode == "ExamResult"))

// {

// var firstOrDefault = examDocuments.FirstOrDefault(d => d.TypeCode == "ExamResult");

// if (firstOrDefault != null)

// model.REPORT\_URL = firstOrDefault.FileBusinessViewUrl;

// }

// // 增加到结果集中

// list.Add(model);

// }

// else

// {

// \_log.Error($"调用RegistPatientExam出错, ObservationUID:{item.observationRequest.ObservationUID}，患者检查信息不完整");

// }

//}

//// 数据的映射

//listResult.ResultList = list;

//return listResult;

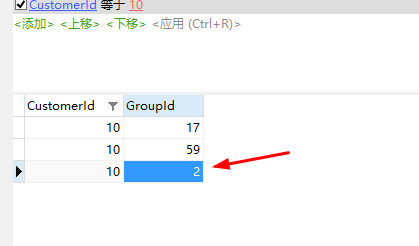
30.

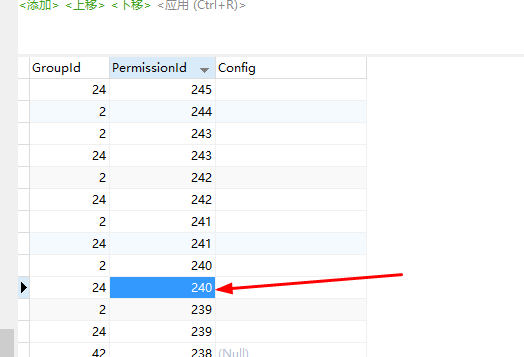
1）影像通知，增加急诊优先

2）view控制器，同步到.7，能力开放平台

3）app获取检查列表增加的DataSource和DataSourceName。

31.





自己加了权限。加了个2，机构管理员的组，这样会有加策略的权限。

这样就有权限了。



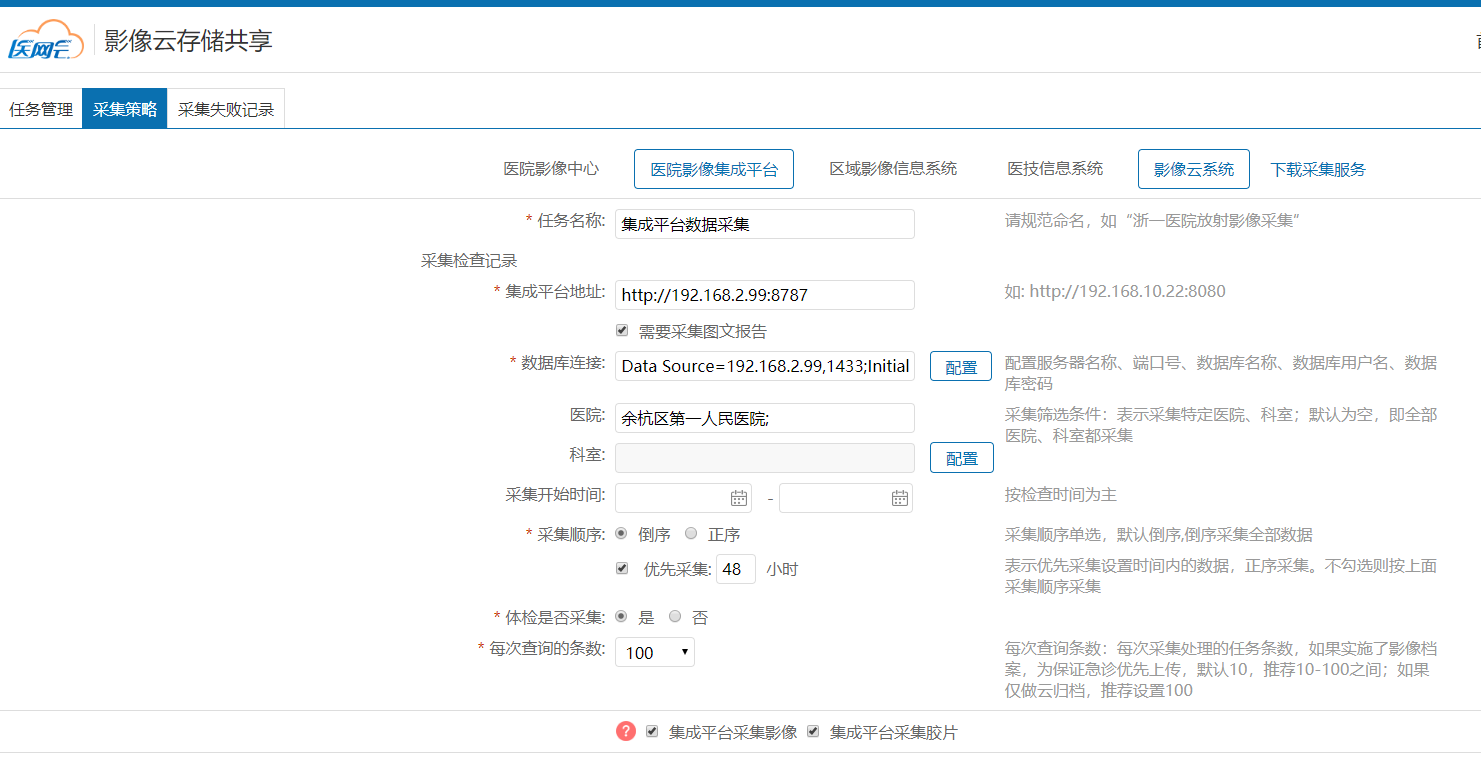
好了。搞定。



建立采集服务。



发现这样配置，在注册检查触发器的同时，就会把影像和胶片触发器给注册，只是ActiveFlag是0，false。查询不到。但其实从集成平台下载影像和胶片的话，是需要启动的。



32.

33.

34.