//This is an example of a report I had to create that would be dynamic, depending on the view.
//My task would be to create or convert old reports from the Telerik style to a new blend of Telerik and dynamic style.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Telerik.Reporting;
using Work.Services;
using Work.DataAccess.DataMembers;
using System.Collections;
using Work.Api.Models.Integration;

namespace Work.Web.ReportDefinitions.DataProviders
{
   public class ClaimsByReferralSourceReportDataProvider : ISetData
   {
      public string GetSubTitle(IDynamicReportingModuleService reportingModuleService,
ReportParameterCollection reportParameters)
      {
         var startDate = reportParameters["startDate"].Value as String;
         var endDate = reportParameters["endDate"].Value as String;
         return $"{startDate} - {endDate}";
      }

      public string GetTitle(IDynamicReportingModuleService reportingModuleService,
ReportParameterCollection reportParameters)
      {
         var userId = new Guid(reportParameters["userId"].Value as string);
         var user =
reportingModuleService.GetServiceUnitOfWork().GetUserService().GetUserById(userId);
         var referralText = Text.Content(user.Email, Enumerations.CustomText.FollowUpService,
reportingModuleService.GetServiceUnitOfWork()).ToTitleCase();
         return $"Claims By {referralText} Source Report";

      }

      public void SetData(IDynamicReportingModuleService reportingModuleService,
ReportParameterCollection reportParameters, DetailSection detailsSection)
      {

         var providerId = ReportParameterHelpers.GetProviderId(reportParameters);
         var userId = new Guid(reportParameters["userId"].Value as string);
         var user =
reportingModuleService.GetServiceUnitOfWork().GetUserService().GetUserById(userId);
         var referralText = Text.Content(user.Email, Enumerations.CustomText.FollowUpService,
reportingModuleService.GetServiceUnitOfWork()).ToTitleCase();
```

```
        var orgText = Text.Content(user.Email, Enumerations.CustomText.Organization,
reportingModuleService.GetServiceUnitOfWork()).ToTitleCase();

        var headerList = new List<Telerik.Reporting.TextBox>();
        var itemList = new ArrayList();
        var colWidth = 150;
        var cellHeight = 30;

        var dateStart  = ReportParameterHelpers.GetStartDate(reportParameters);
        var dateEnd = ReportParameterHelpers.GetEndDate(reportParameters);


        headerList.Add(ReportBuilderHelpers.createCellText("Date", "HealthyRoster.TableHeader",
"Claim Created", 120, cellHeight, null, "= Fields.Date"));
        headerList.Add(ReportBuilderHelpers.createCellText("ReferralSource",
"HealthyRoster.TableHeader", $"{referralText} Source", colWidth, cellHeight, null, "=
Fields.ReferralSource"));
        headerList.Add(ReportBuilderHelpers.createCellText("Organization",
"HealthyRoster.TableHeader", $"{orgText}", colWidth, cellHeight, null, "= Fields.Organization"));
        headerList.Add(ReportBuilderHelpers.createCellText("BillAmount",
"HealthyRoster.TableHeader", "Bill Amount", 120, cellHeight, null, "= Fields.BillAmount"));
        headerList.Add(ReportBuilderHelpers.createCellText("Claimout1",
"HealthyRoster.TableHeader", "Claim Out 1", 120, cellHeight, null, "= Fields.Claimout1"));
        headerList.Add(ReportBuilderHelpers.createCellText("Claimout2",
"HealthyRoster.TableHeader", "Claim Out 2", 120, cellHeight, null, "= Fields.Claimout2"));
        headerList.Add(ReportBuilderHelpers.createCellText("Claimout3",
"HealthyRoster.TableHeader", "Claim Out 3", 120, cellHeight, null, "= Fields.Claimout3"));
        headerList.Add(ReportBuilderHelpers.createCellText("NetPayment",
"HealthyRoster.TableHeader", "Net Payment", 120, cellHeight, null, "= Fields.NetPayment"));


        var columnDescriptors = new List<CellDescriptor>
           {
              new CellDescriptor
              {
                 CellName = "Date",
                 FieldKey = "Date"
              },
              new CellDescriptor
              {
                 CellName = "ReferralSource",
                 FieldKey = "ReferralSource"
              },
              new CellDescriptor
              {
                 CellName = "Organization",
                 FieldKey = "Organization"
              },
              new CellDescriptor
```

```
                {
                    CellName = "BillAmount",
                    FieldKey = "BillAmount"
                },
                new CellDescriptor
                {
                    CellName = "Claimout1",
                    FieldKey = "Claimout1"
                },
                new CellDescriptor
                {
                    CellName = "Claimout2",
                    FieldKey = "Claimout2"
                },
                new CellDescriptor
                {
                    CellName = "Claimout3",
                    FieldKey = "Claimout3"
                },
                new CellDescriptor
                {
                    CellName = "NetPayment",
                    FieldKey = "NetPayment"
                }
            };

        var allClaims = new List<ClaimByReferral>();
        var pageIndex = 1;

        while
(reportingModuleService.GetServiceUnitOfWork().GetIntegrationService().GetClaimsByReferral(prov
iderId, dateStart.Date, dateEnd, pageIndex, 1000).Count > 0)
        {

allClaims.AddRange(reportingModuleService.GetServiceUnitOfWork().GetIntegrationService().GetCl
aimsByReferral(providerId, dateStart.Date, dateEnd, pageIndex, 1000));
            pageIndex++;
        }

        var items = allClaims.Select(r => new
        {
            Date = r.ClaimCreatedDate != null ? r.ClaimCreatedDate.Value.ToString("MM/dd/yyyy") :
"",
            ReferralSource =
reportingModuleService.GetServiceUnitOfWork().GetReportService().GetProviderById(r.ReferralProv
iderUserId),
            Organization =
reportingModuleService.GetServiceUnitOfWork().GetReportService().GetOrganizationById(r.Organiz
ationId),
```

```
            BillAmount = r.TotalChargeAmount,
            Claimout1 = r.TransactionDetails.Count > 0 ?
r.TransactionDetails.First().TransactionAmount.ToString() : "",
            Claimout2 = r.TransactionDetails.Count > 1 ?
r.TransactionDetails.ElementAt(1).TransactionAmount.ToString() : "",
            Claimout3 = r.TransactionDetails.Count > 2 ?
r.TransactionDetails.ElementAt(2).TransactionAmount.ToString() : "",
            NetPayment = r.TransactionDetails.Sum(t => t.TransactionAmount)
        }).ToList();

        itemList.AddRange(items);
        var newTable = ReportBuilderHelpers.CreateTable("enagementDetails", headerList, itemList,
columnDescriptors, 100, 30, 0, 0);
        detailsSection.Items.AddRange(new Telerik.Reporting.ReportItemBase[] { newTable });
    }

    public void SetReportParameters(IDynamicReportingModuleService reportingModuleService,
ReportParameterCollection reportParameters, IDictionary<string, string> incomingParameters)
    {
        ReportParameterHelpers.AddOrganizationIdsParameter(reportingModuleService,
reportParameters, incomingParameters, true, false);
        ReportParameterHelpers.AddPatientListParameter(reportingModuleService, reportParameters,
incomingParameters);
    }


  }
}
```