

LANDSBYGGEFONDEN – DCAB

DD130 - DETALJERET DESIGN DATAMODUL

Version: 1.0
Status: 04 - Reviewet
Godkender: [Navn på godkender]
Forfatter: [Author]

netcompany

Dokumenthistorik

Version	Dato	Forfatter	Status	Bemærkninger
0.1	Dd-mm-åååå		Udkast	
0.3	05-10-2022		Udkast	Tilføjelse efter review
1.0	21-10-2022			Review og godkendelse

Referencer

Reference	Titel	Forfatter	Version

Indholdsfortegnelse

1	Introduktion.....	3
2	Servicebag.....	3
3	SQL Server Analysis Services (SSAS).....	4
3.1	Oprettelse af forbindelse	4
3.2	Forespørgsel af data med DAX	5
3.3	Mapping af SSAS Forespørgsel til Klasser.....	5
3.4	Returnering af dataobjekt	6
4	Deployment	7
5	DAX Queries.....	8
5.1	Pris for PPV-plan	8
5.2	Pris for PPV-plan for bygningsdel	8
5.3	Pris for PPV-plan for bygning.....	9

1 Introduktion

Formålet med dette dokument er at give en struktureret beskrivelse af DCABs Datamodul som skal håndtere statistik forespørgsler fra DCAB, og returnere statistisk filtreret og beregnet data fra DCABs Statistiks SQL Server Analysis Services (SSAS) Tabulare Kube.

API Servicen udstiller et REST API Endpoint med en opsat Swagger side, endpointet modtager forespørgsler på data via filtreringsværdier og forespurgte kolonner, og returnere filtreret data opdelt efter Hierarki strukturen for kaldet samt data på de adspurgte filtreringsmuligheder for de angivne kolonner. Dette gøres for at gøre filtreringen af dataene mere effektiv, da kubens umiddelbart efter indtastning af en værdi begynder at filtrere nedad i hierarkiet og dermed hurtigere kan hente de data, der skal præsenteres. Kuben har også nogle forudbestemte beregninger, der optimerer processen. Det giver således et hurtigere svar, end dataene ville blive hentet fra databasen og øger performance.

2 Servicelag

Data vil blive præsenteret i Landsbyggefondens UI som en separat fane 'Statistik'.

Når en bruger logger på, så tages brugerinformation fra AD(?) og på den måde går det at se hvilken bruger som er loggede på og hvilke rettigheder brugeren har. Brugeren ser kun den eller de organisationer som det tilhører. Input bliver så en liste af OrganisationId, og disse anvendes for at ge data fra de andre tabellerne ned i hierarkiet. Region → Bygninger hentes fra DCAB database og PPVPlan → PPVPlanCost hentes fra SSAS kube. Brugeren kan siden selv filtrere på specifikke kolonner og værdier.

Hierarkiet ser ud sådan her:

hierarkietHentes fra DCAB Databas:

implantation

- Region(DCAB)
 - Kommune(DCAB)
 - Organisation(DCAB)
 - Afdeling(DCAB)
 - Byggeafsnit(DCAB)
 - Bygninger(BBR)

Hentes fra SSAS kube:

- PPVPlan(DCAB)
 - PPVBuilding/BuildingUnit(DCAB)
 - PPVBuildingPartDCAB)
 - PPVActivity(DCAB)
 - PPVCost(DCAB)

3 SQL Server Analysis Services (SSAS)

For at tilgå SSAS skabes der en generel metode til lagring og eksekvering af forespørgsler i datamodulets Visual Studio-løsning.

3.1 Oprettelse af forbindelse

SSAS kaldes fra ExecuteDaxQueryAsync i QueryTool.cs:

```
public DataTable ExecuteDaxQueryAsync(string query)
{
    var tabularResults = new DataTable();

    using (var connection = CreateConnection())
    {
        connection.Open();

        var currentDataAdapter = new AdomdDataAdapter(query, connection);
        currentDataAdapter.Fill(tabularResults);

        connection.Close();
    }

    return tabularResults;
}
```

SSAS tilgås via CreateConnection() metoden i QueryTool.cs:

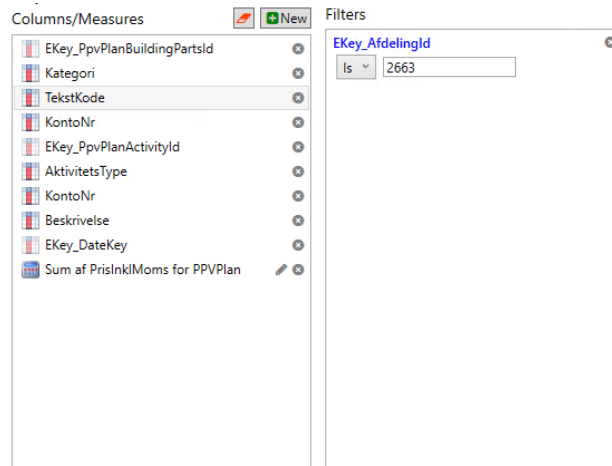
```
private AdomdConnection CreateConnection()
{
    /******
    Define Connection
    *****/
    var servername = _configuration["SSASConnectionString:Servername"];
    var catalog = _configuration["SSASConnectionString:Catalog"];
    AdomdConnection adomdConnection = new AdomdConnection($"Data Source={servername};
Initial Catalog={catalog}");
    return adomdConnection;
}
```

Metoden burger en SSAS Connections string som indeholder servername og catalog. CreateConnection() returnerer en AdomdConnection till DCAB_STAT_TAB. Forespørgslerne oprettes mod SSAS-kuben via ADOMD-biblioteket, hvilket er et nødvendigt link imellem DCAB og SSAS.

Type	Value
Data Source	localhost
Initial Catalog	DCAB_STAT_TAB

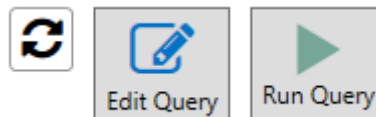
3.2 Forespørgsel af data med DAX

Gennem forespørgsler bestemmes det, hvilke data der skal hentes, og hvordan de skal filtreres. Forespørgslerne skrives i DAX queries og bygges i DaxStudio, og implementeres siden i Visual Studio. Fra Dax studio oprettes en forbindelse til SSAS-kuben og tabellerne downloades. Der er også en funktion kaldet QueryBuilder, hvorfra du kan trække og slippe de kolonner, du vil bruge. Der er også en filterfunktion, der fungerer på lignende måde. Da relationerne mellem tabellerne allerede er oprettet i kubens, så er det nok at tage in AfdelingId som input og sedan filtreres det ned alle trin (se 2 Servicelag).



Figur 1: Query builder

Efter at have valgt kolonner, målinger og filtre, kan du redigere forespørgslen for at få en færdigkrevet forespørgsel.



Figur 2 Execute Query builder

Forespørgsler henter data fra forskellige kolonner som siden sammensættes til tabeller og kolonner. De data, vi fokuserer på, er de som indeholder data om PPV-planer, Bygningsstatistik og Energidata. Når query køres, hentes data fra SSAS-kuben og sæts til en DataTable. Vi bruger en datatabel, fordi den omtrent svarer til en tabel i en database. En datatabel er en repræsentation i hukommelsen af en enkelt databasetabel, som har samlingen af kolonner og rækker, og den henter kun én række ad gangen. De data, som datatabellen er udfyldt med, skal ikke specificeres på forhånd. De specifikke forespørgsler som bruges findes i afsnit 0

DAX Queries.

3.3 Mapping af SSAS Forespørgsel til Klasser

Næste trin er at mappe daten i DataTable til et objekt 'PPVPlanDto'. Der er 3 forskellige mappers, som hver især bruges alt efter om PPV-planen er for en building, buildingpart eller buildingunit. Nedenfor er et eksempel på, hvordan en PPV-plan for en buildingpart mappes. Denne mapper kaldes også fra de andre to mappers.

```
namespace LBF.DCAB.Datamodul.DTL.Mappers.PPVPlanMappers
{
    public class DataTableToBuildingPartsDtoMapper : IMapper<DataTable, List<PPVPlanBuildingPartDto>>
    {
        public List<PPVPlanBuildingPartDto> MapTo(DataTable dt)
        {
            return dt.AsEnumerable().GroupBy(x => new { a = x[0], b = x[1], c = x[2], d = x[3] })
                .Select(x =>
                    new PPVPlanBuildingPartDto
                    {
                        EKey_PpvPlanBuildingPartsId = Guid.Parse((string)x.Key.a),
                        Kategori = x.Key.b is DBNull ? String.Empty : (string)x.Key.b,
                        TekstKode = x.Key.c is DBNull ? String.Empty : (string)x.Key.c,
                        KontoNr = x.Key.d is DBNull ? String.Empty : (string)x.Key.d,
                        Activities = x.GroupBy(x => new { a = x[4], b = x[5], c = x[6], d = x[7] })
                            .Select(x =>
                                new PPVPlanActivityDto
                                {
                                    EKey_PpvPlanActivityId = Guid.Parse((string)x.Key.a),
                                    AktivitetsType = Convert.ToInt32(x.Key.b),
                                    KontoNr = x.Key.c is DBNull ? String.Empty : (string)x.Key.c,
                                    Beskrivelse = x.Key.d is DBNull ? String.Empty : (string)x.Key.d,
                                    PPVPlanCostDtos = x.Select(x => new PPVPlanCostDto()
                                        {
                                            EKey_DateKey = (string)x[8],
                                            SumAfPrisInklMoms = Convert.ToInt32(x[9])
                                        }).ToList(),
                                    PPVPlanActivitySum = x.Sum(x => Convert.ToInt32(x[9]))
                                }
                            ).ToList()
                    }
                ).ToList();
        }
    }
}
```

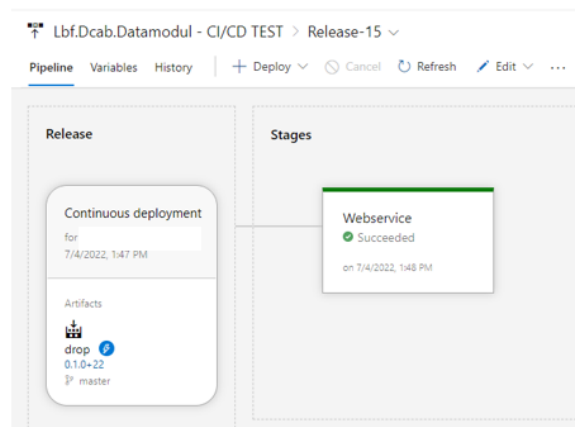
3.4 Returnering af dataobjekt

Hvis en bruger er koplet på flere organisationer, så må de laves et kald til kuben per organisation. Derfor returneres ikke kun et objekt, uden en **List<PPVPlanDto>** til API. Nedenfor vises hvordan en PPVPlanDto ser ud.

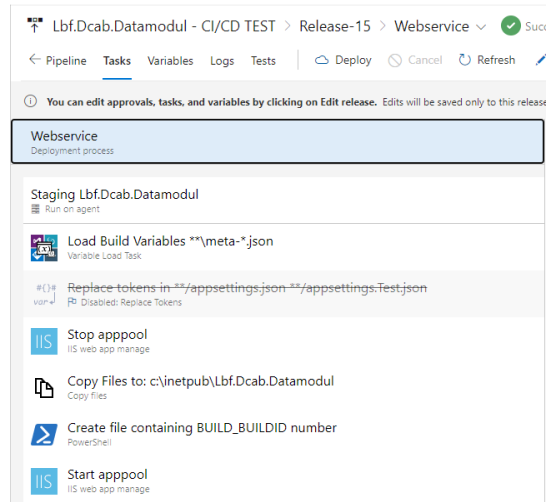
```
namespace LBF.DCAB.Datamodul.DTL
{
    public class PPVPlanDto
    {
        public int AfdelingsId { get; set; }
        public List<PPVPlanBuildingPartDto> PPVPlanBuildingParts { get; set; }
        public List<PPVPlanBuildingUnitDto> PPVPlanBuildingUnits { get; set; }
        public List<PPVPlanBuildingDto> PPVPlanBuildings { get; set; }
    }
}
```

4 Deployment

Deployment laves i Azure DevOps via Continuous Integration. For en mere detaljeret beskrivelse af deployment, se [00300-Vedligeholdelses-vejledning Statistik](#).



Figur 3: Deployment af datamodul



Figur 4 Trinene til deployment

5 DAX Queries

Nedenfor er DAX forespørgslerne til PPV-planer.

5.1 Pris for PPV-plan

```
private Task<DataTable> GetCostForPPVPlansDto(int afdelingIds)
{
    string query = "EVALUATE " +
        "SUMMARIZECOLUMNS( " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[EKey_PpvPlanBuildingPartsId], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[Kategori], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[TekstKode], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[EKey_PpvPlanActivityId], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[AktivitetsType], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[Beskrivelse], " +
        "'DMSA_Fact DCAB_PPVPlanCost'[EKey_DateKey], " +
        "KEEPFILTERS(TREATAS( { " + afdelingIds + " }, 'DMSA_Dimension DCAB_PpvPlans'[EKey_Afdel
ingId] )), " +
        "\"Sum af PrisInklMoms for PPVPlan\", [Sum af PrisInklMoms for PPVPlan]);

    return Task.FromResult(_queryTool.ExecuteDaxQueryAsync(query));
}
```

5.2 Pris for PPV-plan for bygningsdel

```
private Task<DataTable> GetCostForPPVPlanBuildingUnitDto(int afdelingIds)
{
    string query = "EVALUATE " +
        "SUMMARIZECOLUMNS( " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingUnits'[EKey_PpvPlanBuildingUnitsId], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingUnits'[LbfEnhedsNr], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[EKey_PpvPlanBuildingPartsId], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[Kategori], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[TekstKode], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[EKey_PpvPlanActivityId], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[AktivitetsType], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[Beskrivelse], " +
        "'DMSA_Fact DCAB_PPVPlanCost'[EKey_DateKey], " +
        "KEEPFILTERS(TREATAS( { " + afdelingIds + " }, 'DMSA_Dimension DCAB_PpvPlans'[EKey_Afdel
ingId] )), " +
        "\"Sum af PrisInklMoms for byggeafsnit\", [Sum af PrisInklMoms for byggeafsnit]);

    return Task.FromResult(_queryTool.ExecuteDaxQueryAsync(query));
}
```

5.3 Pris for PPV-plan for bygning

```
private Task<DataTable> GetCostForPPVPlanBuildingDto(int afdelingIds)
{
    string query = "EVALUATE " +
        "SUMMARIZECOLUMNS( " +
        "'DMSA_Dimension DCAB_PpvPlanBuildings'[EKey_PpvPlanBuildingsId], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildings'[EjendomsNr], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[EKey_PpvPlanBuildingPartsId], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[Kategori], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[TekstKode], " +
        "'DMSA_Dimension DCAB_PpvPlanBuildingParts'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[EKey_PpvPlanActivityId], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[AktivitetsType], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[KontoNr], " +
        "'DMSA_Dimension DCAB_PpvPlanActivities'[Beskrivelse], " +
        "'DMSA_Fact DCAB_PPVPlanCost'[EKey_DateKey], " +
        "KEEPFILTERS(TREATAS( { " + afdelingIds + " }, 'DMSA_Dimension DCAB_PpvPlans'[EKey_Afdel
ingId] )), " +
        "\\Sum af PrisInklMoms for bygninger\\", [Sum af PrisInklMoms for bygninger]);

    return Task.FromResult(_queryTool.ExecuteDaxQueryAsync(query));
}
```