

LEDNINGSÄGARMODUL

Systemgränssnitt

1. Distributionslista

Dokumentet ska distribueras som leverans till PTS. Ska vara tillgängligt för PTS samt projektmedlemmar.

2. Revisionsinformation

Rev.	Datum	Av	Kommentar
B	2015-02-11	Peter Thorin	Nytt dokument-id 2006, tidigare 8503. Pappersstorlek satt till A4. Informationsklassad som öppen. Revisionsinformation tillagd.
B1	2015-04-13	Daniel Garcia	Added a quick start guide to show the steps needed in order to start using the API and how to deploy it.

3. Innehåll

[1. Distributionslista](#)

[2. Revisionsinformation](#)

[3. Innehåll](#)

[4. Inledning](#)

[4.1. Bakgrund](#)

[5. Gränssnitt](#)

[5.1. Livscykel](#)

[5.1.1. Exempel på livscykel](#)

[5.2. ICustomDataAccess](#)

[5.2.1. BeginTransaction](#)

[5.2.2. EndTransaction](#)

[5.2.3. GetAreasOfInterest](#)

[5.2.4. GetGeometries](#)

[5.3. IExternalAreaOfInterest](#)

[5.3.1. UniqueName](#)

[5.3.2. Wkt](#)

[5.4. AreaOfInterestsBadStateException](#)

[6. Getting Started](#)

[7. Relaterade dokument](#)

4. Inledning

Dokumentet beskriver hur gränssnittet mellan Ledningsägarmodulen och ledningsägarens databas skall se ut. Det ger underlag för hur ledningsägaren själv implementerar ett gränssnitt mot sin geografiska databas för integration med Ledningsägarmodulen.

4.1. Bakgrund

De flesta användare av Ledningsägarmodulen kommer att exportera den geografiska informationen om sina ledningsnät från sin egen databas till filer. Filerna omformas sedan av Ledningsägarmodulen till intresseområden, som exporteras till Ledningskollen.

Några ledningsägare har dock så pass omfattande ledningsnät att det inte är praktiskt att regelbundet exportera den geografiska informationen till fil. Då finns istället alternativet att själva implementera ett gränssnitt mellan Ledningsägarmodulen och sin databas. Därmed kan Ledningsägarmodulen arbeta direkt mot ledningsägarens databas istället för att gå via exporterade filer.

5. Gränssnitt

Det gränssnitt som ledningsägaren skapar för kommunikation mellan Ledningsägarmodulen och ledningsägarens databas görs tekniskt genom att implementera två stycken s.k. *interface* i Microsofts .NET-plattform.

Ledningsägarmodulen använder sig av version 3.5 av Microsoft .NET.

Den huvudsakliga delen av gränssnittet utgörs av interfacet *ICustomDataAccess*, som Ledningsägarmodulen använder för att ställa alla frågor mot ledningsägarens databas.

Utöver detta behöver ledningsägaren även implementera hjälp-interfacet *IExternalAreaOfInterest*, som är en enkel representation av ett intresseområde.

För felhantering innehåller gränssnittet även en Exception-klass, *AreaOfInterestsBadStateException*.

5.1. Livscykel

Då Ledningsägarmodulen startas instansierar den ledningsägarens implementation av *ICustomDataAccess*. Denna instans återanvänds sedan fram till dess att ledningsägarmodulen avslutas, då implementationens *Dispose*-metod anropas. Livslängden för en instans av *ICustomDataAccess* är således mycket lång – potentiellt dagar, veckor eller till och med månader.

Alla frågor som ställs mot ledningsägarens databas via *ICustomDataAccess* görs i transaktioner. En transaktion kan bestå av en eller flera frågor mot databasen. En transaktion inleds med att metoden *BeginTransaction* anropas, och avslutas med ett motsvarande anrop till *EndTransaction*.

Eftersom gränssnittet enbart används för att läsa data ur databasen har de vanliga transaktionsbegreppen `commit` och `rollback` ingen mening i detta sammanhang och ersätts helt av `EndTransaction`.

Medan instansen av `ICustomDataAccess` har mycket lång livslängd, har en transaktion kort livslängd; från en enstaka fråga till ett tiotal, i normala fall. Tanken är att ledningsägarens implementation genom transaktionernas avgränsning kan avgöra när den skall allokera respektive frigöra begränsade resurser, t ex databasanslutningar. Samtidigt finns då möjligheten att slippa hantera resurserna för varje enskild fråga till databasen.

Exakt hur transaktionsgränserna utnyttjas avgörs i varje implementation. De skall endast betraktas som rådgivande information från ledningsägarmodulen.

5.1.1. Exempel på livscykel

Nedan följer ett exempel på hur en sekvens anrop till `ICustomDataAccess` kan se ut under komponentens livslängd.

1. Konstruktorn anropas
2. `BeginTransaction()`
3. `GetAreasOfInterest()`
4. `EndTransaction()`
5. `BeginTransaction()`
6. `GetGeometries(...)`
7. `GetGeometries(...)`
8. `EndTransaction()`
9. `Dispose()`

5.2. `ICustomDataAccess`

Interfacet `ICustomDataAccess` är den huvudsakliga delen av det gränssnitt som behöver implementeras av ledningsägaren. Interfacet har följande utseende:

```
public interface ICustomDataAccess : IDisposable {

    /// <summary>
    /// This method will be called as a transaction starts.
    /// No transaction calls will be made before this has returned.
    /// </summary>
    void BeginTransaction();

    /// <summary>
    /// This method will be called as a transaction ends.
    /// No transaction calls will be made after this.
    /// </summary>
    void EndTransaction();

    /// <summary>
    /// Get all external area of interest to be updated.
    /// </summary>
```

```
/// <returns>A list of IExternalAreaOfInterest to be updated</returns>
IEnumerable<IExternalAreaOfInterest> GetAreasOfInterest();

/// <summary>
/// Returns all geometries of interest in the given bounding box
/// </summary>
/// <param name="srid">Spatial Reference System Identifier</param>
/// <param name="x1">X value for coordinate 1 of bounding box</param>
/// <param name="y1">Y value for coordinate 1 of bounding box</param>
/// <param name="x2">X value for coordinate 2 of bounding box</param>
/// <param name="y2">Y value for coordinate 2 of bounding box</param>
/// <returns>
/// Wkt for all geometries of interest within the given bounding
/// box
/// </returns>
string GetGeometries(int srid, double x1, double y1, double x2, double y2);
}
```

5.2.1. BeginTransaction

Metoden *BeginTransaction* markerar att ledningskollen påbörjar en sekvens av en eller flera frågor till ledningsägarens databas.

Ett anrop till *BeginTransaction* följs alltid av ett anrop till *EndTransaction* innan en ny transaktion påbörjas.

5.2.2. EndTransaction

Metoden *EndTransaction* markerar att ledningsägarmodulen nu avslutar en sekvens av frågor till ledningsägarens databas.

Ett anrop till *EndTransaction* har alltid föregåtts av ett anrop till *BeginTransaction*.

5.2.3. GetAreasOfInterest

Metoden *GetAreasOfInterest* returnerar inom vilket eller vilka områden ledningsägaren har ledningsnät som Ledningskollen och Ledningsägarmodulen skall ta hänsyn till. De returnerade områdena behöver inte vara detaljerade, eftersom de enbart används för att avgöra när ledningsägarmodulen behöver ställa mer detaljerade frågor till ledningsägarens databas med metoden *GetGeometries*.

De returnerade geometrierna används för att generera ledningsägarens intresseområden i Ledningskollen, efter att geometrierna omvandlats till s.k. kilometerrutor.

Returvärdet är en uppräknings av *IExternalAreaOfInterest*-instanser (se 3.3), där varje instans beskriver ett intresseområde.

5.2.4. GetGeometries

Metoden *GetGeometries* returnerar geometrierna för de ledningsnät som ledningsägaren har inom den av argumenten angivna begränsningsrektangeln. De returnerade geometrierna skall vara formaterade enligt OGC WKT. Begränsningsrektangeln anges i det koordinatsystem som ges från parametern *srid* (Spatial Reference ID), t.ex. 4326 för WGS84, 3006 för SWEREF99 TM, o.s.v.

De frågor som ställs via denna metod kommer ha en begränsad geografisk utbredning, d.v.s. begränsningsrektangelns yta kommer vanligtvis vara högst ett fåtal kvadratkilometer.

5.3. IExternalAreaOfInterest

Ledningsägaren implementerar detta interface och returnerar instanser av detta då metoden *GetAreasOfInterest* i *ICustomDataAccess* anropas.

Detta interface används för att förmedla information om ledningsägarens intresseområden till Ledningsägarmodulen. Ledningsägarmodulen behöver både den geografiska informationen och ett namn för varje intresseområde, och denna information grupperas i *IExternalAreaOfInterest*.

```
public interface IExternalAreaOfInterest {  
  
    /// <summary>  
    /// Unique name of the area of interest.  
    /// </summary>  
    string UniqueName { get; }  
  
    /// <summary>  
    /// Wkt representation of all geometries belonging to the area of interest  
    /// </summary>  
    string Wkt { get; }  
}
```

5.3.1. UniqueName

Metoden *UniqueName* innehåller intresseområdets namn, som både fungerar för att identifiera området för Ledningskollens och ledningsägarmodulens användare, samt internt för systemet. Som namnet säger måste namnet vara unikt för ledningsägaren, d.v.s. om ledningsägaren har flera intresseområden måste samtliga ha olika namn.

5.3.2. Wkt

Metoden *Wkt* innehåller områdets geometrier, formaterade som OGC WKT.

5.4. AreaOfInterestsBadStateException

Om den geometriska informationen befinner sig i ett sådant tillstånd att några anrop ej kan accepteras eller om något annat oförutsätt går snett så bör ett Exception av typen *AreaOfInterestsBadStateException* kastas. Detta kommer omedelbart avsluta transaktionen utan att genomföra den.

6. Komma igång

Nedan följer ett par tips som kan komma till användning när du skapar ditt tillägg för Ledningsägar-modulen.

1. Ditt projekt
 - a. Måste vara ett Visual Studio, Class Library (dll) project
 - b. Kan döpas till vad som helst.
 - c. För att du ska kunna använda gränssnittet (interfacet), som beskrivs i kapitel 5, behöver du använda biblioteket "ICustomDataAccess.dll". Detta hittar du i lib-katalogen i Ledningsägarmodul-installationen.
 - d. Dina klasser som ärver från gränssnittet får inte ha några restriktioner vad gäller namngivning.

2. Driftsätta din implementation

- a. Skapa en ny mapp i Ledningsägar modul-installationen (döp den förslagsvis till 'addons').
- b. I mappen (addons), lägg till biblioteket som ditt projekt skapat (dll).
- c. Nu måste du ändra web.config-filen för Ledningsägar modulprojektet.
- d. Två nya entiteter skall läggas till under appSettings.
- e. Lägg till "CustomDataAccess" sätt värdet till "true"
- f. Lägg till "PluginPath" värdet är pathen till den mapp där dll-filen ligger (.../addons).
- g. Starta om app poolen och gå till Ledningsägar modulsiten. Du behöver inte logga in, vänta till login sidan har laddats in och kolla sedan på filen Lemmy.log. Du skall se följande rad: Lemmy.Service.Impl.PluginService - Create Plugin Kernel, plugin to load:
 Lemmy.Service.ICustomDataAccess->LemmyAddon.YOUR_CLASS_NAME

7. Relaterade dokument

Dokumentnamn	Beskrivning	Utgåva
Ledningsägar modulen Installation och drift	Beskriver installation samt rutiner kring driften.	
Ledningsägar modulen Användarhandledning	Beskriver Ledningsägar modulens funktioner och användargränssnitt.	
Ledningsägar modulen Systemkrav	Beskriver krav på systemmiljön.	
Ledningsägar modulen Översikt	Beskriver Ledningsägar modulen översiktligt	