SPLG-Grouper Integrationsmodul Dotnet

Einführung

Dieses Dokument beschreibt, wie Sie das Integrationsmodul Dotnet in eine .NET-Applikation direkt integrieren können. Für Java-basierte Anwendungen gibt es ein analoges Integrationsmodul Java. Für andere Technologien müssen Sie entweder eine Java- oder .NET-Integrationstechnologie verwenden, oder Sie nutzen den REST-Service, welcher bei beiden Integrationsmodulen verfügbar ist (siehe Dokument «SPLG-Grouper REST-Schnittstelle»).

Referenzierung

Das Grouper-Assembly können Sie im Projektfile wie folgt referenzieren:

```
<ItemGroup>
    <Reference Include="grouper">
        <HintPath>".\grouper.dll</HintPath>
    </Reference>
</ItemGroup>
```

Passen Sie ggf. den Pfad auf die DLL an.

Verwendungsbeispiel

Der folgende Code ist ein kleines Beispiel, wie Sie den Grouper instanziieren und verwenden können.

```
using splg;

namespace demo
{
    public class Program
    {
        static void Main(string[] args)
        {
            // Initialize Grouper
            Grouper grouper = Grouper.Create("config/license.txt");
            grouper.SetDefdir("config/defs-demo.dat");
            grouper.SetSldir("config/spitallisten-demo.dat");

            // Set currently used release (definition/spitalliste)
            var release = "A_2023";
            grouper.SetRelease(release);

            // Get Falldaten (from file or UI or database)
            var falldaten = new Falldaten();
            // ...

            // Group falldaten and get result
            var result = grouper.Group(falldaten);

            // Output result (to file or UI or database or console)
```

```
            Console.Out.WriteLine(result.splg);
        }
    }
}
```

Grouper

Die Grouper-Klasse wird verwendet, um Definitionen und Spitallisten zu laden und Daten zu gruppieren. Sie stellt den zentralen Punkt der Interaktion mit dem SPLG-Grouper dar.

```
public class Grouper
{
    // Erstellt die Grouperinstanz, die Lizenzdatei «licfile» muss
    // existieren und gültig sein, damit gruppiert werden kann.
    public static Grouper Create(string licfile);

    // Setzt das Verzeichnis oder die («Single-File-Def») Datei,
    // in welchen nach Definitionsdateien gesucht wird. Es kann
    // nur ein Verzeichnis respektive eine Datei für Definitionen
    // gesetzt werden. Erneute Aufrufe der Funktion ersetzen
    // frühere Werte.
    public void SetDefdir(string defdir);

    // Setzt das Verzeichnis oder die («Single-File-SL») Datei,
    // in welchen nach Spitallisten gesucht wird. Es können
    // mehrere Verzeichnisse/Dateien durch wiederholten Aufruf
    // der Funktion gesetzt werden.
    public void SetSldir(string sldir);

    // Fügt eine Spitalliste manuell hinzu. Die Spitalliste gilt für
    // den spezifizierten Release «release» und betrifft den durch
    // «burnr» und optional «plz» und «location» identifizierten
    // Betriebsstandort (falls nur «burnr» geliefert wird, werden
    // für «plz» und «location» leere Strings übergeben). Die Liste
    // der Leistungsaufträge «las» bestimmt die dem Betriebsstandort
    // erteilten Leistungsaufträge.
    public void AddSpitalliste(
        String release,
        String burnr,
        String plz,
        String location,
        bool reference,
        String canton,
        List<LA> las
    );

    // Fügt eine Spitalliste manuell hinzu. Die Spitalliste gilt für
    // den spezifizierten Release «release» und betrifft den durch
    // «burnr» und optional «plz» und «location» identifizierten
```

```
    // Betriebsstandort (falls nur «burnr» geliefert wird, werden
    // für «plz» und «location» leere Strings übergeben). Die Liste
    // der SPLG «splgs» bestimmt die dem Betriebsstandort erteilten
    // Leistungsaufträge. Achtung, da hier nur die SPLG spezifiziert
    // werden, können keine Befristungen angegeben werden. Dafür
    // nutzen Sie die «addSpitalliste»-Method, welche eine Liste von
    // LA-Objekten akzeptiert (siehe oben).
    public void AddSpitalliste(
        string release,
        string burnr,
        string plz,
        string location,
        bool reference,
        string canton,
        params string[] splgs
    );

    // Setzt einen Kantons-Override. Der Kantonsoverride
    // muss ein Kantonskürzel sein (z. B. «ZH»). Dieses
    // wird ab nun für alle Fälle beim Leistungscontrolling
    // als Wohnkanton verwendet (bei der Auswahl der zu
    // verwendenden Spitalliste). Ein gesetzter Override
    // kann durch erneuten Aufruf mit leerem String zurück-
    // gesetzt werden.
    public void SetKantonOverride(string kantonOverride);

    // Aktiviert einen Release (z. B. «A_2024» oder «R_2024»)
    // Ab diesem Zeitpunkt werden alle folgenden Fälle mit
    // diesem Release gruppiert. Es kann jederzeit ein neuer
    // oder bereits früher gewählter Release gesetzt werden.
    // Die Definitionen und Spitallisten werden gecacht.
    public void SetRelease(string release);

    // Löscht gecachte Definitionen und Spitallisten.
    public void ClearCache();

    // Löscht konfigurierte sowie gecachte Definitionen.
    public void ClearDefinitions();

    // Löscht konfigurierte sowie gecachte Spitallisten.
    public void ClearSpitallisten();

    // Gruppiert einen Fall
    public Result Group(Falldaten fall);
}
```

```
public class LA
{
    // Erstellt einen unbefristeten Leistungsauftrag für «splg»
    public LA(string splg);

    // Erstellt einen auf das Jahr «year» befristeten
    // Leistungsauftrag für «splg»
    public LA(string splg, int year);

    // Erstellt einen auf den Bereich «validFrom» bis «validTo»
    // befristeten Leistungsauftrag für «splg»
    public LA(string splg, DateOnly validFrom, DateOnly validTo);
}
```

Falldaten

```
public class Falldaten
{
    // controlling-relevant
    public string burnr = "";
    public string plz = "";
    public string standort = "";
    public string wohnkanton = "";
    public string statistikfall = "";
    public string behandlungsart = "";
    public string tarifsystem = "";

    // fallidentifikation
    public string fallid = "";

    // grouper-relevant
    public string agey = "";
    public string aged = "";
    public string ssw = "";
    public string ggw = "";
    public string dmb = "";
    public string freiwilligkeit = "";
    public string austritt = ""; // Austrittsdatum, Format YYYYMMDD

    // grouper-relevant
    public List<FalldatenDiagnose> diagnosen = new();
    public List<FalldatenBehandlung> behandlungen = new();
```

```
    // controlling-relevant

    public List<FalldatenPatientenbewegung> bewegungen = new();

    // output-relevant
    public string ave = "";
    public string weg = "";
    public string sn = "";
    public string skz = "";
    public string ea = "";
    public string ad = "";
    public string ana = "";
    public string ed = "";
    public string mk = "";
    public string ei = "";
    public string gew = "";
    public string hktr = "";

    // output-relevant SwissDRG/TARPSY/STReha
    public string drg = "";
    public string pcg = "";
    public string rcg = "";
    public string mdc = "";
    public string cw = "";
    public string pccl = "";
    public string ecwt = "";

    // zusatz (durchreiche)
    public string zusatz = "";
    public string erhebungsjahr = "";
    public string standortkanton = "";
}

public class FalldatenDiagnose
{
    public int rang;
    public string code;
    public string seitigkeit = "";
    public string zusatz = "";
}

public class FalldatenBehandlung
{
    public int rang;
```

```
    public string code;
    public string seitigkeit = "";
    public string ambExt = "";
    public string beginn = "";
    public List<FalldatenOperateur>? operateure = null;
}

public class FalldatenOperateur
{
    public string gln = "";
    // 1 = Erstoperateur, 2 = Zweitoperateur
    public string funktion = "";
    // 1 = Auf Liste GD,  0 = Nicht auf Liste GD
    public string zulassung = "";
}

public class FalldatenPatientenbewegung
{
    public string beginn = "";
    public string ende = "";
    public string art = "";
    public string burnr = "";
}
```

Resultat

```
public class Result
{
    public string fallid;

    public string splg;
    public List<string> lgs;
    public List<string> quer;
    public List<string> mfzs;
    public List<string> mfzo;

    public List<MFZOPoints> points;

    public int slst;
    public int lactrl;
    public SortedSet<string> lactrlcodes;

    public SortedSet<ErrorCode> errorcode;

    public string notes = "";
    public string zusatz = "";

    public string defversion;
    public List<string> spitallisten;
```

```
    public string GetErrorCode();
}

public class MFZOPoints
{
    public string gln;
    public string lg;
    public double points;
}

public class ErrorCode : IComparable<ErrorCode>
{
    public string GetCode();
    public bool IsOK();
    public bool IsWarning();
    public bool IsError();
}
```

REST-Service

Das Integrationsmodul Dotnet liefert neben dem Grouper Assembly auch eine Service-Anwendung mit, welche einen einfachen REST-Service implementiert, mit dem Sie technologieunabhängig gruppieren können.

Sie starten den Service auf der Kommandozeile wie folgt:

```
.\service.exe start .\license.txt .\defs-demo.dat .\spitallisten-demo.dat
```

Dadurch wird per Default unter Port 8080 auf localhost ein HTTP-Server gestartet, über welchen gruppiert werden kann. Weitere Informationen dazu finden Sie im Dokument «SPLG-Grouper REST-Schnittstelle».