Spezifische Anpassung von Datenklassen zwischen DiversityCollection und DiversityPhone

DiversityPhoneOnly-KLassen

# UserProfile

Die Tabelle Userprofile existiert nur clientseitig und dient der Nutzerverwaltung. Aus dieser werden folgende Informationen benötigt:

* ProjectID: ID des aktuellen Projekts. Nur zum Synchronisationszeitpunkt veränderbar.
* AgentName: Name des Nutzers, wie dieser angezeigt werden soll
* AgentURI: UIR aus DiversityAgents
* Loginname: Loginname in der Datenbank
* RecordGeoPosition: Usereingabe, ob Geodaten erfasst werden sollen.

# TaxonName

# PropertyName

DiversityCollectionOnly-KLassen

# Entity

# Entity-Usage

# Entity-Representation

# EntityContext\_Enum

DiversityCollection-Phone-Klassen

Die Klassen werden gemäß der DiversityCollection\_Monitoring.htm – Doku gebildet. Auf Anpassungen wird explizit eingegangen.

Für die Synchronisation erhalten alle editierbaren Klassen die Felder isModified und LogUpdatedWhen.

Readonly-Klassen erhalten nur das Feld isModified und es wird im Kommentar darauf hingewiesen, dass es sich um eine ReadOnly-Klasse handelt.

# Analysis

Die Tabelle ist clientseitig Read-only.

Analysis wird clientseitig in einer eigenen Klasse Analysis geführt.

Folgende Anpassungen werden ausgeführt:

* Die Spalte OnlyHierarchie wird nicht übernommen.
* Tabelleneinträge mit OnlyHierarchie=True werden vom Service nicht geholt
* Die Spalte AnalysisParentID wird nicht übernommen
* Sie Spalte Notes wird nicht übernommen
* Die Spalte AnalysisURI wird nicht übernommen, da die Analysis durch die AnalysisID bereits eindeutig identifiziert ist.

# AnalysisResult

Die Tabelle ist clientseitig Read-only.

AnalysisResult wird folgenderweise auf Term abgebildet:

* AnalysisResult->Code
* AnalysisID->SourceID
* Sie Spalte Notes wird nicht übernommen.

# AnalysisTexonomicGroup

AnalysisTaxonomicGroup wird folgenderweise auf Term angebildet:

* AnalysisID -> SourceID
* TaxonomicGroup -> Code

# CollectionAgent

Die Tabelle existiert clientseitig nicht.

Korrespondierende Einträge werden bei der Synchronisation beim Übertragen der CollectionSpecimen in die Serverdatenbank eingefügt.

# CollectionEvent

CollectionEvent wird clientseitig in einer eigenen Klasse Event geführt.

Folgende Anpassungen finden statt:

* Die Spalte Version wird nicht übernommen, sondern bei der Synchronisation mit dem Wert 1 befüllt
* Das Datum wird separat in den Feldern CollectionDay, CollectionMoth und CollectionYear gespeichert
* CollectionDateCategory wird nicht geführt und bei der Synchronisation auf den Wert actual gesetzt.
* Zur Geokodierung werden die Felder Latitude, Longitude und Altitude eingeführt.

# CollectionEventImage

ColllectionEventImage wird auf die Klasse MultimediaObject abgebildet.

Die Klasse enthält in Uri einen Schlüssel zum File im Dateisystem. Dieses muss bei der Synchronisation mit übertragen und auf den neuen Pfad angepasst werden.

Folgende Anpassungen finden statt:

* EventID -> SourceID
* ImageType->MediaType

# CollectionEventLocalisation

Die Tabelle existiert clientseitig nicht.

Korrespondierende Einträge werden bei der Synchronisation beim Übertragen der Tabelle CollectionEvent in die Serverdatenbank eingefügt. Dazu werden je für die Altitude und die WGS84-Daten ein CollectionEventLocalisation angelegt.

Folgende Anpassungen finden statt:

* LocalisationSystemID ist 4 für Altitude und 8 für WGS84
* Location1 enthält Altitude bzw. Longitude
* Location2 enthät Null bzw. Latitude
* DeterminationDate wird auf Event berechnet
* ResponsibleName und ResponsibleURI werden aus dem Userprofile übernommen
* In Geography wird für WGS84 ein Transact-SQL kompatibler String für einen Punkt eingetragen, die Koordinaten bestimmen sich aus latitude, altitude und longitude des Events.

# CollectionEventProperty

Diese Tabelle wird in einer eigenen Klasse CollectionEventProperty geführt.

Folgende Anpassungen finden statt:

* ResponsibleName und RsponsibleAgentUri werden bei der Synchronisation eingefügt und sonst nicht geführt.

# CollectionEventSeries

Diese Tabelle wird in einer eigenen Klasse EventSeries geführt.

Beim Setzen des Endzeitpunkts werden Geodaten in einem Transact-SQL-String in das Feld Geography gespeichert.

# CollectionEventSeriesImage

ColllectionEventSeriesImage wird auf die Klasse MultimediaObject abgebildet.

Die Klasse enthält in Uri einen Schlüssel zum File im Dateisystem. Dieses muss bei der Synchronisation mit übertragen und auf den neuen Pfad angepasst werden.

Folgende Anpassungen finden statt:

* SeriesID -> SourceID
* ImageType->MediaType

# CollectionProject

Die Tabelle existiert clientseitig nicht.

Korrespondierende Einträge werden bei der Synchronisation beim Übertragen der CollectionSpecimen in die Serverdatenbank eingefügt.

# CollectionSpecimen

CollectionSpecimen wird clientseitig in einer eigenen Klasse Specimen geführt.

Folgende Anpassungen finden statt:

* Die Spalte Version wird nicht übernommen, sondern bei der Synchronisation mit dem Wert 1 befüllt

# CollectionSpecimenImage

ColllectionSpecimenImage wird auf die Klasse MultimediaObject abgebildet.

Die Klasse enthält in Uri einen Schlüssel zum File im Dateisystem. Dieses muss bei der Synchronisation mit übertragen und auf den neuen Pfad angepasst werden.

Folgende Anpassungen finden statt:

* SpecimenID -> SourceID

Die Tabelle wird auch für IdentificationUnits verwendet. Folgende Anpassungen finden statt:

* IdentificationUnitID -> SourceID

Für diese müssen bei der Synchronisation fogende Anpassungen vorgenommen werden:

* ID korrepondierend IdentificationUnit ->IdentificationUnitID
* ID des korrespondierenden Specimen der korrespondierenden IdentificationUnit -> CollectionSpecimenID

# Identification

Die Tabelle existiert clientseitig nicht.

Korrespondierende Einträge werden bei der Synchronisation beim Übertragen der IdentificationUnit in die Serverdatenbank eingefügt.

Für die Zeitpunkte werden die Zeitpunkte des zugehörigen Events verwendet.

# IdentificationUnit

CollectionSpecimen wird clientseitig in einer eigenen Klasse Specimen geführt.

IdentificationUnit wird clientseitig in einer eigenen Klasse IdentificationUnit geführt.

Folgende Anpassungen finden statt:

* In den Feldern LastIdentificationCache, FamiliyCache und OrderCache werden die Informationen zur Identifikation gespeichert. Da diese beim Übertrage überschrieben werden, müssen die Werte in die Tabelle Identification geschrieben werden.
* Es existiert zusätzlich ein Feld IdentificationUri, welche die URI der Identification enthält.
* Zur Geokodierung werden die Felder Latitude, Longitude und Altitude eingeführt.

# IdentificationUnitAnalysis

IdentificationUnitAnalysis wird clientseitig in einer eigenen Klasse IdentificationUnitAnalysis geführt.

Diese hat folgende Struktur:

* IdentificationUnitID
* AnalysisID
* IdentificationUnitAnalysisID
* AnalysisResult

Die Klasse wird bei der Synchronisation an die Serverklasse angepasst.

# IdentificationUnitGeoAnalysis

Die Klasse wird clientseitig nicht geführt, sondern analog zur CollectionEventLocalisation bei der Synchronisation erzeugt. Die Geodaten sind aus der korrespondierenden identificatioUnit zu entnehmen.

# LocalisationSystem

Die Klasse wird clientseitig nicht geführt. Bei der Synchronisation von referenzierenden Klassen wird bei Altitude 4 und bei WGS84 der Wert 8 eingetragen.

# ProjectAnalysis

Die Klasse wird clientseitig nicht geführt. Da das Projekt nur nach einer Synchronisation verwendet werden kann, werden die Analysen zu diesem Zeitpunkt für das neu gewählte Projekt übertragen.

# Property

Property wird folgenderweise auf Term abgebildet:

* Property -> SourceID
* PropertyName-> Code
* PropertyParentID, DefaultAccuracyOfProperty, DefaultMeasurementUnit, ParsingMethodName werden nicht geführt.