Es gibt die Tabellen ProjectTable und DynamicEntity (Stammdaten). Zur Änderungsverfolgung gibt es zusätzlich zu den beiden Tabellen jeweils eine Changeset Tabelle.

Master:

Slave:

ProjetTableChangeSet:

MaxRowersion: select max(rowversion) from ProjectTable

* SyncStatus: Datensatz ist synchron (1/true) oder nicht synchron (0/false) (muss gepushed werden)
* IsDeleted
* RowVersion: Wird auf (MaxRowversion + 1) gesetzt, wenn es zu Änderungen an den Stammdaten kommt.
* Json: Änderungen der Spalten

Ablauf:

Zunächst werden die Stammdaten, bei denen MaxRowversion größer ist, als lokal heruntergeladen und nötigenfalls überschrieben.

Es werden alle Änderungen vom Server geholt und mit den lokalen Changeset verglichen. Die Änderungen werden in die Stammdaten integriert, bei Konflikten werden die behandelt. Bei den Stammdaten wird SyncStatus auf false gesetzt. Es werden alle Stammdaten mit SyncStatus false eins nach dem anderen an den Server gepusht. Dieser gibt die neue RowId zurück, welche lokal gespeichert wird.

DynamicEntityChangeSet:

MaxRowersion: select max(rowversion) from DynamicEntity

* SyncStatus: Datensatz ist synchron (1/true) oder nicht synchron (0/false) (muss gepushed werden)
* IsDeleted
* RowVersion: Wird auf (MaxRowversion + 1) gesetzt, wenn es zu Änderungen an den Stammdaten kommt.
* Json: Änderungen der Spalten

Ablauf:

Zunächst werden alle lokalen Änderungen im ChangeSet an den Server gepusht. Das ChangeSet kann anschließend geleert werden. Anschließend werden Stammdaten vom Server runtergeladen, deren MaxRowversion größer ist, als die lokale MaxRowVersion.

Slave:

ProjetTableChangeSet:

MaxRowersion: select max(rowversion) from ProjectTable

* SyncStatus: Datensatz ist synchron (1/true) oder nicht synchron (0/false) (muss gepushed werden)
* IsDeleted
* RowVersion: Wird auf (MaxRowversion + 1) gesetzt, wenn es zu Änderungen an den Stammdaten kommt.
* Json: Änderungen der Spalten

Ablauf:

Zunächst werden alle lokalen Änderungen im ChangeSet an den Server gepusht. Das ChangeSet kann anschließend geleert werden. Anschließend werden Stammdaten vom Server runtergeladen, deren MaxRowversion größer ist, als die lokale MaxRowVersion.

DynamicEntityChangeSet:

MaxRowersion: select max(rowversion) from DynamicEntity

* SyncStatus: Datensatz ist synchron (1/true) oder nicht synchron (0/false) (muss gepushed werden)
* IsDeleted
* RowVersion: Wird auf (MaxRowversion + 1) gesetzt, wenn es zu Änderungen an den Stammdaten kommt.
* Json: Änderungen der Spalten

Ablauf:

Zunächst werden alle lokalen Änderungen im ChangeSet an den Server gepusht. Das ChangeSet kann anschließend geleert werden. Anschließend werden Stammdaten vom Server runtergeladen, deren MaxRowversion größer ist, als die lokale MaxRowVersion.

Felder

Zur Synchronisation werden folgende Felder benötigt:

1. sync\_status – Zeigt an, ob Datensatz synchronisiert ist (true/1) oder noch synchronisiert werden muss(false/0)
2. is\_deleted -
3. guid –
4. row\_version – Wird vom Server zurückgegeben, und bei jedem update um 1 erhöht. Stimmt bei PUT row\_version nicht, so kommt es zu einem Konflikt

Grober Synchronisationsablauf:

1. Finde das maximale last\_modified Datum der zu synchronisierenden Tabelle
2. Finde alle Datensätze der zu synchronisierenden Tabelle, bei denen sync\_status 0/false ist
3. Sende das Datum und die Daten an den Server
4. Der Server Insterted/Updated die Datensätze und Setzt bei jeden Datensatz last\_modified auf DateTime.Now
5. Der Server sendet alle Datensätze der zu synchronisierenden Tabelle, bei denen last\_modified größer ist, als das vom Client gesendete Datum
6. Der Client setzt bei jeden erhaltenen Datensatz sync\_status auf 1/true und Inserted/Updated die Datensätze