

Mithilfe des *Webservice Marktdaten* kann eine lokal gehostete Marktdaten-Datenbank aktualisiert werden. Änderungen an den Daten werden paketweise in sogenannten „Deltas“ zur Verfügung gestellt. Diese können über REST-Webservices abgerufen werden. Zur Kennzeichnung der Versionierung verfügen diese Deltas über eine eindeutige `Delta_ID`.

Eine Übersicht und Beschreibung der erforderlichen Webservices für die Aktualisierung der Marktdaten ist online verfügbar:

<https://ws.enet-navigator.de/marktdaten/swagger-ui/>

Verwendung von enet Navigator® Webservices

Um REST-Webservices des enet Navigator® verwenden zu können, benötigen sie ein gültiges Maschinenkonto, dem die notwendige Lizenz für den jeweiligen Webservice zugewiesen ist.

Jeder Webservice-Aufruf muss mittels Basic-Authentication gegenüber dem Server authentifiziert werden. Hierfür muss das Request um einen Authentication-Header erweitert werden, der aus dem Prefix „BASIC“ und dem Base64-kodierten String „maschinenkontoname:maschinenkontopasswort“ (mit gültigen Zugangsdaten) besteht.

Ablauf Deltamigration

Um eine lokale MarktdatenDB per *Webservice Marktdaten* zu aktualisieren, sind die folgenden Schritte notwendig:

1. Abfrage der nächsten verfügbaren `Delta_ID`
2. Anfrage und Download des entsprechenden Deltas
3. Einspielen des Inhalts in die lokale DB innerhalb einer Transaktion
4. Quittieren der `Delta_ID` nach erfolgreichem Einspielen

1. Abfrage der nächsten verfügbaren `Delta_ID`

Der aktuelle Stand der lokalen Marktdaten-Datenbank sowie der Verlauf der eingespielten Deltas kann in der Tabelle „ndbd“.„Info“ abgefragt werden. In der Regel ist die höchste vorhandene `Delta_ID` auch die ID mit dem jüngsten „Ende“-Datum. Ist das nicht der Fall, deutet dies auf einen Fehler hin. Die Liste der `Delta_ID` muss dabei nicht lückenlos sein.

Mit der so ermittelten `Delta_ID` kann per REST-Get das nächste verfügbare Delta angefragt werden:

Beispiel: <https://ws.enet-navigator.de/marktdaten/naechste-delta-id?deltald=4711>

Sollen pro Aufruf direkt mehrere verfügbare `Delta_ID` zurückgegeben werden, kann der optionale Parameter „limit“ verwendet werden, um bis zu 15 Folge-ID zu erhalten. Da jede Minute neue Deltas erzeugt werden können, ist dieses Vorgehen sinnvoll, sobald neue Deltas in längeren Intervallen abgefragt werden sollen.

Beispiel: <https://ws.enet-navigator.de/marktdaten/naechste-delta-id?deltald=4711&limit=15>

Ist die als Ergebnis zurückgegebene Liste leer, sind keine aktuelleren Deltas vorhanden. Ansonsten enthält die Liste zwischen einer und bis zu durch „limit“ definierten Anzahl von `Delta_ID`.

2. Anfrage und Download des entsprechenden Deltas

Nachdem angefragt wurde, welche Deltas verfügbar sind, können diese mittels ihrer `Delta_ID` ebenfalls per REST-Get angefragt werden

Beispiel: <https://ws.enet-navigator.de/marktdaten/download?deltald=4712>
(die nachfolgende `Delta_ID` zu „4711“)

Der Endpunkt liefert als Ergebnis das Delta als ZIP-Datei. Diese enthalten zwei zum Archiv gleichnamige Dateien:

- eine .SQL-Datei, welche die im Delta enthaltenen Änderungen als SQL-DML-Skript enthält
- eine .MD5-Datei, welche die Prüfsumme der .SQL-Datei beinhaltet

Anhand der Prüfsumme kann die Integrität der .SQL-Datei verifiziert werden.

3. Einspielen des Inhalts in die lokale DB innerhalb einer Transaktion

Um den Inhalt des Deltas in die lokale Datenbank zu übertragen, muss das aus der zuvor angefragten ZIP-Datei extrahierte SQL-DML-Skript auf der lokalen Datenbank ausgeführt werden. Das Skript ist sowohl mit PostgreSQL als auch MSSQL-Server kompatibel.

Wurde ein Delta-Skript erfolgreich ausgeführt, ist die zugehörige `Delta_ID` anschließend in der Tabelle „ndbd“ „Info“ aufgeführt.

Wichtig:

- Deltas müssen in exakt der Reihenfolge eingespielt werden, wie sie der Endpunkt „naechste-delta-id“ vorgibt
- Das SQL-DML-Skript jedes Deltas sollte für sich alleine und innerhalb einer Transaktion ausgeführt werden
- **Sollte es beim Ausführen eines Skripts in einer Transaktion zu einem Fehler kommen, darf dieses Delta nicht übersprungen und mit dem Nächsten fortgefahren werden!** Generiert ein Delta einen Fehler während des Einspielens, muss dieser korrigiert werden, bevor mit weiteren Deltas fortgefahren wird.

4. Quittieren der Delta_ID nach erfolgreichem Einspielen

Nachdem ein Delta erfolgreich in die lokale Datenbank eingespielt wurde, muss es per REST-Post quittiert werden.

<https://ws.enet-navigator.de/marktdaten/delta-quittieren?deltald=4712>

Sollten hierbei Fehler im Prozess auftreten, z. B. ein Delta quittiert werden, während für ein älteres noch die Quittierung aussteht, wird der Vorgang verweigert.

Auch wenn zuvor mehrere `Delta_ID` angefragt worden sind, sollten Deltas immer einzeln eingespielt und zunächst quittiert werden, bevor mit dem nächsten Delta fortgefahren wird. Dadurch wird gewährleistet, dass der Prozess im Fehlerfall wieder an der korrekten `Delta_ID` anknüpfen kann.