



windata SEPA - API Basic / Pro

Dokumentation

10.08.2021

windata GmbH & Co. KG

Inhaltsverzeichnis

1	Vorabinformationen
2	Lastschriften bei DTAUS0-Dateien.....
3	Konverterprozess.....

1 Vorabinformationen

Die SEPA Converter Library ermöglicht die Konvertierung von herkömmlichen DTAUS0- und windata CSV 1.1-Dateien nach SEPA XML-Dateien. Die zugehörige Library unterstützt hierbei Überweisungen, sowie Lastschriften. Der Prozess arbeitet feldbasiert und ermöglicht den Zugriff auf sämtliche Felder eingelesener Dateien. Diese werden geprüft und auf semantische, sowie logische Fehler, untersucht. Durch einen Filteralgorithmus werden fehlerhafte Datensätze aussortiert, sowie intelligent sortiert und gruppiert. Die Ausgabe lässt sich als reine SEPA-Datei oder als SEPA-Container nach einem festgelegten Namensmuster definieren. Die erzeugte(n) Datei(en) können anschließend weiterverwendet werden.

2 Lastschriften bei DTAUS0-Dateien

Um Lastschriften zu konvertieren, sind einige zusätzliche Schritte erforderlich. Das windata CSV-Format v1.1 stellt hierfür bereits relevante Felder (z.B. Gläubiger-ID, Mandatsinformationen, etc.) bereit, welche während der Konvertierungsvorgangs automatisch übernommen werden. Da diese Felder jedoch bei DTAUS0-Dateien fehlen, ist das Verfahren hier nicht übertragbar. Aus diesem Grund müssen der Library zwei CSV-Zusatzdateien beige-steuert werden, um eine Verknüpfung von herkömmlichen Lastschriftdatensätzen und erforderlichen Lastschriftdaten zu ermöglichen:

1. Mandatsinformationen: Diese Datei enthält relevante Mandatsinformationen. Ein Datensatz ist jeweils mit einer KontoNr. und BLZ verknüpft.

Die Datei muss folgenden Pfad aufweisen:

[Ausführungspfad]\ext\DTAUS0Ext.csv

2. Gläubiger-Informationen: Diese Datei enthält relevante Gläubigerinformationen. Ein Datensatz ist jeweils mit einer KontoNr. und BLZ verknüpft.

[Ausführungspfad]\ext\InitPtyId.csv

3 Konverterprozess

Der Konverterprozess gliedert sich in vier Schritte, welche in nachfolgender Reihenfolge aufgerufen werden:

1. Es wird eine neue Instanz der CSEPACnv-Klasse erzeugt. Der Konstruktor überprüft sämtliche Eingabedateien auf semantische Fehler und erzeugt gegebenenfalls eine Exception. Zudem werden die Zusatzdateien, Datenbanken und Umgebungsvariablen auf Korrektheit verifiziert. Nach einer erfolgreichen Validierung ist das CSEPACnv-Objekt gültig und kann weiterverwendet werden.
2. Es wird eine neue Instanz der CDTAUS0Reader bzw. CCSVReader-Instanz erzeugt. Mittels der Methode ExtractTransactions() wird ein neues CSet-basiertes Objekt erzeugt. Dieses enthält eine feldbasierte Liste, welche die physikalische Datei als Objekt darstellt
3. Mittels der integrierten Methoden der CSet-Klasse kann die Liste auf Korrektheit überprüft werden, um fehlerhafte Datensätze zu erkennen. Mittels der Methode FilterTransactions() können sämtliche ungültige Datensätze entfernt werden.
4. Das gefilterte CSet-Objekt wird einem CSEPACreator-Objekt übergeben, welches mittels integrierter Methoden eine Liste von speicherbasierten Objekten erzeugt. Mittels der Methode SaveFiles() können diese Objekte physikalisch serialisiert werden, um sie auf einem Speicherort abzulegen.