

# 4.8.1 SEPA: Nuclet-Schnittstellen

 Sprache: **Deutsch** · [English](#)

## 4.8.1 SEPA: Nuclet-Schnittstellen

Ziel dieses Schrittes ist, das tatsächlich im Zielnuclet verwendeten Businessobjekt an den Java-Code des SEPA-Nuclets anzubinden.

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

### Auf dieser Seite

- [Verwandte Seiten](#)

Ziel dieses Schrittes ist, das tatsächlich im Zielnuclet verwendeten Businessobjekt an den Java-Code des SEPA-Nuclets anzubinden. Dazu wird der Ansatz verfolgt, Nuclet-unabhängige Wrapper-Klassen zu verwenden – anstelle der zu den Businessobjekten generierten Businessobjekt-Klassen.

Zu den technischen Hintergründen lesen Sie dazu bitte auch den Artikel "Schnittstellen in Java-Regeln".



Dieser Integrationsschritt kann erst durchgeführt werden, sobald die [Vorbedingungen aus Abschnitt 3](#) erfüllt sind – d.h. sobald entsprechende Attribute in den genutzten Businessobjekten existieren.

Bei der Anbindung des SEPA-Nuclets sind Wrapper-Klassen für Kreditoren (Lieferanten, o.ä.), Debitoren (Kunden, o.ä.), Referenzen (Rechnungen, Belege, o.ä.) und Zahlungsreferenzen (Forderungen, Rechnungspositionen, o.ä.) anzupassen.

Die folgende Übersicht führt alle Wrapper-Klassen gruppiert nach Anwendungsfall auf. Alle Klassen befinden sich im Java-Package **org.nuclet.sepa.wrapper**.

Sollten Sie das SEPA-Nuclet entweder nur für Lastschriften oder nur für Überweisungen verwenden wollen, dann müssen nur die jeweiligen Klassen für Ihren Anwendungsfall angepasst werden. Die restlichen Klassen können Sie ignorieren.

| Wrapper-Klasse                     | Anmerkung                                                         | Funktion                                                                                            | Anwendungsfall           |
|------------------------------------|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------|
| CreditorReferenceWrapper           | neu ab v2.0.0                                                     | Anbindung an genutzte Referenzobjekte (Eingangsrechnungen, Gutschriften,o.ä.)                       | Überweisungen            |
| CreditorWrapper                    | neu ab v2.0.0                                                     | Anbindung an verwendete Kreditoren, Lieferanten, o.ä.                                               |                          |
| CreditTransferReferenceWrapper     | neu ab v2.0.0                                                     | Anbindung an verwendete Zahlungsreferenzobjekte (Forderungen, Rechnungspositionen, Buchungen, etc.) |                          |
| DebitorReferenceWrapper            | neu ab v2.0.0, ersetzt ReferenceWrapper                           | Anbindung an genutzte Referenzobjekte (Ausgangsrechnungen,o.ä.)                                     | Lastschriften            |
| DebitorWrapper                     |                                                                   | Anbindung an verwendete Debitoren, Kunden, o.ä.                                                     |                          |
| DirectDebitReferenceWrapper        | neu ab v2.0.0                                                     | Anbindung an verwendete Zahlungsreferenzobjekte (Forderungen, Rechnungspositionen, Buchungen, etc.) |                          |
| <del>ReferenceWrapper</del>        | obsolet ab v2.0.0, wird ersetzt durch DebitReferenceWrapper       | Anbindung an genutzte Referenzobjekte (Rechnungen,o.ä.)                                             | <del>Lastschriften</del> |
| <del>PaymentReferenceWrapper</del> | obsolet ab v2.0.0, wird ersetzt durch DirectDebitReferenceWrapper | Anbindung an verwendete Zahlungsreferenzobjekte (Forderungen, Rechnungspositionen, etc.)            | <del>Lastschriften</del> |

Tabelle 4.8.1: Übersicht Nuclet-Schnittstellen



Bitte beachten Sie, dass es in diesem Integrationsschritt nur darum geht, die Anbindung an die Attribute der tatsächlich genutzten Businessobjekte herzustellen.

Es ist von Ihnen an dieser Stelle im Normalfall keinerlei Prozesslogik zu implementieren.

### 4.8.1.1 CreditorReferenceWrapper

Die Klasse ReferenceWrapper dient als Nuclet-Schnittstelle für tatsächlich genutzte Referenzobjekte (Rechnungen, Forderungen, Belege, etc.).

Das sind jene Objekte, die bei SEPA-Exporten als Referenz für Zahlungsanweisungen verwendet werden.

| Methoden                     | Funktion                                                                                                                                                                                                                                     | Anpassung optional? |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| CreditorReferenceWrapper     | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                                                                                                                         | nein                |
| getCreditorId()              | liefert die Datenbank-ID des Kreditoren                                                                                                                                                                                                      | ja                  |
| getCreditTransferReference() | liefert das Identifizierungsmerkmal für Überweisungen (also z.B. die Rechnungsnummer)                                                                                                                                                        | ja                  |
| getReferenceDate()           | liefert ein Referenzdatum (bspw. das Rechnungsdatum)                                                                                                                                                                                         | ja                  |
| getTotalAmount()             | liefert den Gesamtbetrag einer Transaktion (bspw. den Rechnungsbetrag);<br><br>falls die Implementierung <i>null</i> zurückliefert, werden die Beträge aus den Einzelbeträgen ermittelt<br><br>(siehe unten: CreditTransferReferenceWrapper) | ja                  |

Tabelle 4.8.1.1: Anpassungen in CreditorReferenceWrapper

Beispiele dazu sind in Kommentarlöcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

#### org.nuclet.sepa.wrapper.CreditorReferenceWrapper

```
package org.nuclet.sepa.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

import org.nuclet.sepa.wrapper.AbstractCreditorReferenceWrapper;

// @replace!
//
// import org.nuclet.buinessemplate.SupplierBilling;

/**
 * Abstract wrapper class for debtors (supplier billings, invoices, etc.)
 *
 * @version 1.1
 * @date 11.01.2016
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.1.3
 * @sincenucletversion 1.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class CreditorReferenceWrapper extends AbstractCreditorReferenceWrapper
{
    public CreditorReferenceWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof SupplierBilling) {
        //     this.businessObject = reference;
        // }
    }
}
```

```

/**
 * Liefert die Datenbank-ID des Kreditoren
 *
 */
public Long getCreditorId()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((SupplierBilling)this.businessObject).getSupplierId();

    return null;
}

/**
 * Liefert das Identifizierungsmerkmal für Lastschriften (z.B. die Rechnungsnummer)
 *
 */
public String getCreditTransferReference()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((SupplierBilling)this.businessObject).getBillingNumber();

    return null;
}

/**
 * Liefert ein Referenzdatum (bspw. das Rechnungsdatum)
 *
 */
public Date getReferenceDate()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((SupplierBilling)this.businessObject).getBillingDate();

    return null;
}

/**
 * Liefert den Gesamtbetrag (bspw. den Rechnungsbetrag)
 *
 * @note Falls die Implementierung *null* zurückliefert, werden die Beträge aus den
 * Einzelbeträgen ermittelt (d.h. aus den CreditTransferReferenceWrapper-Objekten)
 */
public BigDecimal getTotalAmount() {
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((SupplierBilling)this.businessObject).getTotalAmountGross();

    return BigDecimal.ZERO;
}
}

```

#### 4.8.1.2 CreditorWrapper

Die Klasse CreditorWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Kreditoren-Businessobjekt.

Der Konstruktor der Klasse und die Methode getIso4217Code() sind mit anwendungsspezifischem Verhalten zu befüllen. Letztere Methode sollte den ISO-4217-Währungscode (siehe dazu [Wikipedia](#)) des Währungsobjektes zurückliefern.

| Method          | Funktion                                                                                                                                           | Anpassung optional? |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| CreditorWrapper | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                               | nein                |
| getBic()        | liefert die BIC des Debitoren, d.h. das Geschäftskennzeichen der Bank des Debitoren (BIC = Business Identifier Code, früher: Bank Identifier Code) | nein                |
| getIban()       | liefert die IBAN des Debitoren, d.h. die internationale Bankkontonummer (IBAN = International Bank Account Number)                                 | nein                |
| getName()       | liefert den Namen des Debitoren                                                                                                                    | nein                |

Tabelle 4.8.1.2: Anpassungen in CreditorWrapper

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an die tatsächlich genutzte Währungsbusinessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

#### org.nuclet.sepa.wrapper.CreditorWrapper

```
package org.nuclet.sepa.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

import org.nuclet.sepa.wrapper.AbstractCreditorWrapper;

// @replace!
//
// import org.nuclet.business-template.Supplier;

/**
 * Wrapper implementation for creditors
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class CreditorWrapper extends AbstractCreditorWrapper
{
    public CreditorWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof Supplier) {
        //     this.businessObject = reference;
        // }
    }
}

/**
```

```

* Liefert den Namen des Debitors
*
*/
public String getName()
{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// return ((Supplier)this.businessObject).getName();

return null;
}

/**
* Liefert den BIC für SEPA-Zahlungen
*
*/
public String getBic()
{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// return ((Supplier)this.businessObject).getBic();

return null;
}

/**
* Liefert den IBAN für SEPA-Zahlungen
*
*/
public String getIban()
{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// return ((Supplier)this.businessObject).getIban();

return null;
}
}

```

#### 4.8.1.3 CreditTransferReferenceWrapper

Die Klasse CreditTransferReferenceWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Businessobjekt für Zahlungsreferenzen.

Mit Zahlungsreferenzen sind jene Objekte gemeint, :

| Methode                          | Funktion                                                                                                                                               | Anpassung optional? |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| CreditTransferReferenceWrapper() | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                                   | nein                |
| getAmount()                      | liefert den zu zahlenden Betrag;<br>wird genutzt, wenn die Methode getTotalAmount() aus dem übergeordneten CreditorReferenceWrapper null zurückliefert | nein                |
| getReferenceId()                 | liefert die Datenbank-ID des übergeordneten Referenzobjektes (siehe 4.7.1.2)                                                                           | nein                |
|                                  |                                                                                                                                                        |                     |

|                               |                                                               |      |
|-------------------------------|---------------------------------------------------------------|------|
| setSEPAExportDate()           | setzt das Export-Datum nach erfolgtem SEPA-Export             | nein |
| setSEPAPaymentInformationId() | setzt die SEPA-Zahlungsinformation nach erfolgtem SEPA-Export | nein |

Tabelle 4.8.1.3: Anpassungen in CreditTransferReferenceWrapper

Beispiele dazu sind in Kommentarblöcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

#### org.nuclet.sepa.wrapper.CreditTransferReferenceWrapper

```
package org.nuclet.sepa.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

import org.nuclet.sepa.wrapper.AbstractPaymentReferenceWrapper;

// @replace!
//
// import org.nuclet.buinessemplate.SupplierBillingPosition;

/**
 * Wrapper implementation for SEPA credit transfer references (invoices, due payments, etc.)
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class CreditTransferReferenceWrapper extends AbstractCreditTransferReferenceWrapper
{
    public CreditTransferReferenceWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof SupplierBillingPosition) {
        //     this.businessObject = reference;
        // }
    }

    /**
     * Liefert die Datenbank-ID des übergeordneten Referenzobjektes
     * (also bspw. die ID der Rechnung, falls es sich bei den Zahlungsreferenzen um
     * Rechnungspositionen handelt)
     */
    public Long getReferenceId()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((SupplierBillingPosition)this.businessObject).getClientBillingId();

        return null;
    }
}
```

```

}

/**
 * Liefert den Zahlungsbetrag
 *
 */
public BigDecimal getAmount()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((SupplierBillingPosition)this.businessObject).getAmount();

    return null;
}

public void setSEPATransactionId(final Long lngSEPATransactionId)
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // ((SupplierBillingPosition)this.businessObject).setSEPATransactionId(lngSEPATransactionId);
}

public void setSEPAExportDate(final Date datSEPAExportDate)
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // ((SupplierBillingPosition)this.businessObject).setSEPAExportDate(datSEPAExportDate);
}
}

```

#### 4.8.1.4 DebitorReferenceWrapper

Die Klasse ReferenceWrapper dient als Nuclet-Schnittstelle für tatsächlich genutzte Referenzobjekte (Rechnungen, Forderungen, Belege, etc.).

Das sind jene Objekte, die bei SEPA-Exporten als Referenz für Zahlungsanweisungen verwendet werden.

| Methoden                  | Funktion                                                                                                                                                                                                                           | Anpassung optional? |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| DebitorReferenceWrapper   | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                                                                                                               | nein                |
|                           |                                                                                                                                                                                                                                    |                     |
| getDebitorId()            | liefert die Datenbank-ID des Debitoren                                                                                                                                                                                             | ja                  |
| getDirectDebitReference() | liefert das Identifizierungsmerkmal für Lastschriften (also z.B. die Rechnungsnummer)                                                                                                                                              | ja                  |
| getReferenceDate()        | liefert ein Referenzdatum (bspw. das Rechnungsdatum)                                                                                                                                                                               | ja                  |
| getTotalAmount()          | liefert den Gesamtbetrag einer Transaktion (bspw. den Rechnungsbetrag);<br><br>falls die Implementierung <i>null</i> zurückliefert, werden die Beträge aus den Einzelbeträgen ermittelt (siehe unten: DirectDebitReferenceWrapper) | ja                  |

Tabelle 4.8.1.4: Anpassungen in DebitorReferenceWrapper

Beispiele dazu sind in Kommentarkomblöcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

#### org.nuclet.sepa.wrapper.DebitorReferenceWrapper

```
package org.nuclet.sepa.wrapper;
```

```
import java.math.BigDecimal;
```

```

import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

import org.nuclet.sepa.wrapper.AbstractDebitorReferenceWrapper;

// @replace!
//
// import org.nuclet.businesstemplate.ClientBilling;

/**
 * Abstract wrapper class for debtors (client billings, invoices, etc.)
 *
 * @note replaces the original class ReferenceWrapper
 *
 * @version 2.1
 * @date 11.01.2016
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.1.3
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class DebitorReferenceWrapper extends AbstractDebitorReferenceWrapper
{
    public DebitorReferenceWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof ClientBilling) {
        //     this.businessObject = reference;
        // }
    }

    /**
     * Liefert die Datenbank-ID des Debitoren
     *
     */
    public Long getDebitorId()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getClientId();

        return null;
    }

    /**
     * Liefert das Identifizierungsmerkmal für Lastschriften (z.B. die Rechnungsnummer)
     *
     */
    public String getDirectDebitReference()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getBillingNumber();

        return null;
    }
}

```



```

    }

    /**
     * Liefert ein Referenzdatum (bspw. das Rechnungsdatum)
     *
     */
    public Date getReferenceDate()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getBillingDate();

        return null;
    }

    /**
     * Liefert den Gesamtbetrag (bspw. den Rechnungsbetrag)
     *
     * @note Falls die Implementierung *null* zurückliefert, werden die Beträge aus den
     * Einzelbeträgen ermittelt (d.h. aus den DirectDebitReferenceWrapper-Objekten)
     */
    public BigDecimal getTotalAmount() {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getTotalAmountGross();

        return BigDecimal.ZERO;
    }
}

```

#### 4.8.1.5 DebitorWrapper

Die Klasse DebitorWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Debitoren-Businessobjekt.

Der Konstruktor der Klasse und die Methode getIso4217Code() sind mit anwendungsspezifischem Verhalten zu befüllen. Letztere Methode sollte den ISO-4217-Währungscode (siehe dazu [Wikipedia](#)) des Währungsobjektes zurückliefern.

| Methode                         | Funktion                                                                                                                                           | Anpassung optional? |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| DebitorWrapper                  | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                               | nein                |
|                                 |                                                                                                                                                    |                     |
| getBic()                        | liefert die BIC des Debitoren, d.h. das Geschäftskennzeichen der Bank des Debitoren (BIC = Business Identifier Code, früher: Bank Identifier Code) | nein                |
| getDateOfSEPAMandateSignature() | liefert das Datum, an dem das SEPA-Mandat vom Debitoren ausgestellt/unterzeichnet wurde                                                            | nein                |
| getHasSEPAMandateChanged()      | liefert Informationen darüber, ob sich das SEPA-Mandat des Debitoren verändert hat (d.h. ob die Bankverbindung geändert wurde)                     | nein                |
| getIban()                       | liefert die IBAN des Debitoren, d.h. die internationale Bankkontonummer (IBAN = International Bank Account Number)                                 | nein                |
| getIsNewDebitorAgent()          | liefert Informationen darüber, ob der Debitor seit dem letzten SEPA-Export das Kreditinstitut gewechselt hat                                       | nein                |
| getName()                       | liefert den Namen des Debitoren                                                                                                                    | nein                |
| getOriginalDebitorAccount()     | liefert die letzte IBAN des Debitoren, falls sich diese seit dem letzten SEPA-Export geändert haben sollte                                         | nein                |
| getSEPAMandateIdentification()  | liefert die SEPA-Mandatskennung des Debitoren                                                                                                      | nein                |
| getSEPASequencetyped()          | liefert die SEPA-Laufzeit für den Debitoren (ist die SEPA-Lastschrift erstmals, einmalig oder wiederkehrend?)                                      | nein                |

|                             |                                                                                                                              |      |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------|------|
| setHasSEPA_MandateChanged() | setzt Informationen darüber, ob sich das SEPA-Mandat des Debitoren verändert hat (d.h. ob die Bankverbindung geändert wurde) | nein |
| setIsNewDebitorAgent()      | setzt Informationen darüber, ob der Debitor seit dem letzten SEPA-Export das Kreditinstitut gewechselt hat                   | nein |
| setOriginalDebitorAccount() | setzt die letzte IBAN des Debitoren, falls sich diese seit dem letzten SEPA-Export geändert haben sollte                     | nein |
| setSEPASequencetype()       | setzt die SEPA-Laufzeit für den Debitoren (ist die SEPA-Lastschrift erstmals, einmalig oder wiederkehrend?)                  | nein |

Tabelle 4.8.1.5: Anpassungen in DebitorWrapper

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an die tatsächlich genutzte Währungsbusinessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

#### org.nuclet.sepa.wrapper.DebitorWrapper

```
package org.nuclet.sepa.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

// @replace!
//
// import org.nuclet.buinessstemplate.Client;

/**
 * Wrapper implementation for debtors
 *
 * @version 1.0
 * @date 21.03.2014
 * @nuclet org.nuclet.SEPA
 * @nucletversion 1.0.0
 * @sincenucletversion 1.0.0
 * @since 21.03.2014
 *
 * @author frank.lehmann@nuclos.de
 */
public class DebitorWrapper extends AbstractDebitorWrapper
{
    public DebitorWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof Client) {
        //     this.businessObject = reference;
        // }
    }

    /**
     * Liefert den Namen des Debtors
     */
    public String getName()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    }
}
```

```

        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getName();

        return null;
    }

    /**
     * Liefert die SEPA-Mandatskennung
     */
    public String getSEPAMandateIdentification()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getSEPAMandateIdentification();

        return null;
    }

    /**
     * Liefert das Ausstellungsdatum des SEPA-Mandats
     */
    public Date getDateOfSEPAMandateSignature()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getDateOfSEPAMandateSignature();

        return null;
    }

    /**
     * Liefert Informationen darüber, ob sich das SEPA-Mandat auf Debitorenseite
     * geändert hat
     */
    public Boolean getHasSEPAMandateChanged()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getHasSEPAMandateChanged();

        return null;
    }

    /**
     * Setzt die Informationen darüber, ob sich das SEPA-Mandat auf Debitorenseite
     * geändert hat
     */
    public void setHasSEPAMandateChanged(final Boolean blnHasSEPAMandateChanged)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // ((Client)this.businessObject).setHasSEPAMandateChanged(blnHasSEPAMandateChanged);
    }

    /**
     * Liefert Informationen darüber, ob sich das Kreditinstitut auf Debitorenseite
     * gewechselt wurde
     */
    public Boolean getIsNewDebitorAgent()
    {

```

```

        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getIsNewDebitorAgent();

        return null;
    }

    /**
     * Setzt die Informationen darüber, ob sich das Kreditinstitut auf Debitorensseite
     * gewechselt wurde
     */
    public void setIsNewDebitorAgent(final Boolean blnIsNewDebitorAgent)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // ((Client)this.businessObject).setIsNewDebitorAgent(blnIsNewDebitorAgent);
    }

    /**
     * Liefert die Datenbank-ID der SEPA-Laufzeit
     */
    public Long getSEPASequenceTypeId()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getSEPASequenceTypeId();

        return null;
    }

    /**
     * Setzt die Datenbank-ID der SEPA-Laufzeit
     */
    public void setSEPASequenceTypeId(final Long lngSEPASequenceTypeId)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // ((Client)this.businessObject).setSEPASequenceTypeId(lngSEPASequenceTypeId);
    }

    /**
     * Liefert den BIC für SEPA-Zahlungen
     */
    public String getBic()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Client)this.businessObject).getBic();

        return null;
    }

    /**
     * Liefert den IBAN für SEPA-Zahlungen
     */

```

```

public String getIban()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((Client)this.businessObject).getIban();

    return null;
}

/**
 * Liefert die alte IBAN bei SEPA-Mandatsänderungen
 *
 */
public String getOriginalDebitorAccount()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((Client)this.businessObject).getOriginalDebitorAccount();

    return null;
}

/**
 * Setzt die alte IBAN bei SEPA-Mandatsänderungen
 *
 */
public void setOriginalDebitorAccount(final String strOriginalDebitorAccount)
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // ((Client)this.businessObject).setOriginalDebitorAccount(strOriginalDebitorAccount);
}
}

```

#### 4.8.1.6 DirectDebitReferenceWrapper

Die Klasse CreditTransferReferenceWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Businessobjekt für Zahlungsreferenzen.

Mit Zahlungsreferenzen sind jene Objekte gemeint, :

| Methode                       | Funktion                                                                                                                                                     | Anpassung optional? |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| DirectDebitReferenceWrapper() | Zuordnung des tatsächlich genutzten Businessobjektes                                                                                                         | nein                |
| getAmount()                   | liefert den zu zahlenden Betrag;<br>wird genutzt, wenn die Methode getTotalAmount() aus dem übergeordneten DebitorReferenceWrapper <i>null</i> zurückliefert | nein                |
| getReferenceId()              | liefert die Datenbank-ID des übergeordneten Referenzobjektes (siehe 4.7.1.2)                                                                                 | nein                |
| setSEPAExportDate()           | setzt das Export-Datum nach erfolgtem SEPA-Export                                                                                                            | nein                |
| setSEPAPaymentInformationId() | setzt die SEPA-Zahlungsinformation nach erfolgtem SEPA-Export                                                                                                | nein                |

Tabelle 4.8.1.3: Anpassungen in DirectDebitReferenceWrapper

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

## org.nuclet.sepa.wrapper.DirectDebitReferenceWrapper

```
package org.nuclet.sepa.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.facade.Stateful;

import org.nuclet.sepa.wrapper.AbstractDirectDebitReferenceWrapper;

// @replace!
//
// import org.nuclet.businesstemplate.ClientBillingPosition;

/**
 * Wrapper implementation for SEPA direct debit references (invoices, due payments, etc.)
 *
 * @note replaces the original class AbstractPaymentReferenceWrapper
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class DirectDebitReferenceWrapper extends AbstractDirectDebitReferenceWrapper
{
    public DirectDebitReferenceWrapper(final Modifiable reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof ClientBillingPosition) {
        //     this.businessObject = reference;
        // }
    }

    /**
     * Liefert die Datenbank-ID des übergeordneten Referenzobjektes
     * (also bspw. die ID der Rechnung, falls es sich bei den Zahlungsreferenzen um
     * Rechnungspositionen handelt)
     */
    public Long getReferenceId()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBillingPosition)this.businessObject).getClientBillingId();

        return null;
    }

    /**
     * Liefert den Zahlungsbetrag
     */
    public BigDecimal getAmount()
```

```

{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// return ((ClientBillingPosition)this.businessObject).getAmount();

return null;
}

public void setSEPATransactionId(final Long lngSEPATransactionId)
{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// ((ClientBillingPosition)this.businessObject).setSEPATransactionId(lngSEPATransactionId);
}

public void setSEPAExportDate(final Date datSEPAExportDate)
{
// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// ((ClientBillingPosition)this.businessObject).setSEPAExportDate(datSEPAExportDate);
}
}

```

## Verwandte Seiten

### **Nuclet: SEPA (sind im Handelsnuclet verfügbar)**

SEPA-Datenblatt

[Öffnen](#)

### **Schnittstellen**

Alle Integrationen

[Öffnen](#)