

Nuclet: cKonto (not yet available)

Language: [Deutsch](#) · [English](#)

Nuclet: cKonto (not yet available)

Data sheet and documentation entry point for the cKonto Nuclet (validating bank details via the cKonto interface).

CONCEPT

INTEGRATORS

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [Related pages](#)



Machine-translated – under review

This page was machine-translated from German as a first draft and is currently under review. The authoritative source remains the **German original** (see the language switch above).

- [Nuclet: cKonto \(not yet available\)](#)
- [Overview](#)
 - [Short description](#)
 - [Nuclet components](#)
 - [Java package structure](#)
- [Integration](#)
- [Version](#)
 - [Related pages](#)
 - [Interfaces](#)
 - [Nuclet Wiki](#)

Nuclet for connecting to the cKonto interface

Name:	cKonto
Package:	org.nuclet.ckonto
Version:	1.4.0
Date:	13.09.2017
Nuclos compatibility:	from Nuclos 4.19.3

Overview

Short description

The cKonto Nuclet offers the option to check bank details for validity via the cKonto interface.



Note: To use the cKonto interface, you need a paid account with NetSlave GmbH. Further information can be found at www.ckonto.de.

Nuclet components

The cKonto Nuclet comprises one Java package with three Java classes.

Java package structure

Java package	Java class	Short description
--------------	------------	-------------------

org.nuclet.ckonto	CkontoCheck	container class for processing the cKonto XML
	CkontoConstants	constants for interpreting the interface return values
	CkontoUtils	utility class with validation methods

Table 1: Java package structure

Integration

Necessary integration steps

The integration of the cKonto Nuclet takes place in 4 steps:

1. Download
2. Nuclet import
3. Configuration of the Nuclet parameters
4. Source code adjustments in your own Java rules

All integration steps are explained in detail below.

Step 1: Download

Download the .nuclet file „cKonto-v1.4.0-4.19.3.nuclet“ from the Nuclos web page under „Nuclos Services“ > „Download“ > „Nuclet Download“.

Step 2: Nuclet import

Import the cKonto Nuclet under „Configuration“ > „Nuclet Management“ > „Import“ into your existing Nuclos instance, selecting the file „cKonto-v1.4.0-4.19.3.nuclet“

Step 3: Configuration of the Nuclet parameters

The cKonto Nuclet can be parameterised via Nuclet parameters. The used Nuclet parameters are set up in the Nuclet management under „Konfiguration“.

Parameter	Short description
cKonto Check	indicates whether an account check should take place via the cKonto interface
cKonto Key	registration key
cKonto URL	URL of the cKonto interface

Table 2: Nuclet parameters



Caution: The validation methods `validiereIBAN()` and `validiereKontoverbindung()` of the class `CkontoUtils` always validate the passed parameters, regardless of the value of the Nuclet parameter **cKonto Check**. To query whether a validation should actually take place, the method `istKontovalidierungAktiviert()` in the same class is used.

Step 4: Source code adjustments in your own rules

If the validation of bank details via the cKonto interface is to be integrated, this is done by calling one of the two following methods of the class `CkontoUtils`:

- `validiereIBAN()`: performs a validation of an IBAN
- `validiereKontoverbindung()`: performs a validation of account number and bank sort code

The following example illustrates this integration.

1. Source code example for validation

```
(...)  
  
org.nuclet.ckonto.CkontoCheck check = null;  
org.nuclet.ckonto.CkontoConstants.Status status = null;  
  
try {  
    // Übergabe einer Kontonummer und einer Bankleitzahl an die cKonto-Validierung...  
    check = org.nuclet.ckonto.CkontoUtils.validiereKontoverbindung(strKontonummer, strBankleitzahl);  
  
} catch (javax.xml.bind.JAXBException jaxbe) {  
    // eigene Fehlerbehandlung einsetzen  
    // (...)  
} catch (java.net.MalformedURLException me) {  
    // eigene Fehlerbehandlung einsetzen  
    // (...)  
}  
  
status = org.nuclet.ckonto.CkontoConstants.Status.valueOf("STATUS_" + Check.getStatus().toString());  
  
if (!org.nuclet.ckonto.CkontoConstants.Status.STATUS_1.equals(status)) {  
    // eigener Code, zur Verarbeitung der Status-Informationen  
    // (...)  
}  
  
(...)
```

The enumeration class CkontoConstants comprises all possible return values of the cKonto interface, so that a simple classification via the enumeration comparison is possible.

org.nuclet.ckonto.CkontoConstants

```
public class CkontoConstants  
{  
    public enum Status  
    {  
        STATUS_0(0, "Die Kontonummer ist ungültig, sie kann von diesem Institut nicht vergeben werden."),  
        STATUS_1(1, "Die Kontonummer ist gültig, sie kann von diesem Institut vergeben werden."),  
        STATUS_2(2, "Fehler bei der Eingabe der Bankleitzahl. Sie ist entweder nicht 8-stellig oder enthält  
Buchstaben."),  
        STATUS_3(3, "Fehler bei der Eingabe der Kontonummer. Sie ist entweder zu lang oder enthält  
Buchstaben."),  
        STATUS_4(4, "Eingabefehler bei der Kontonummer und Bankleitzahl (siehe 2 und 3) oder der IBAN."),  
        STATUS_5(5, "Genereller Eingabefehler des Übergabeparameters - enthält evtl. Leerzeichen."),  
        STATUS_6(6, "Fehler im Format des Übergabeparameters."),  
        STATUS_7(7, "Die Bankleitzahl wurde in der Datenbank nicht gefunden (existiert nicht)."),  
        STATUS_8(8, "Die von der Bank verwendete Prüfmethode ist im Demonstrations-Modus nicht verfügbar."),  
        STATUS_9(9, "Die Kontonummer kann nicht geprüft werden, da die Bank entweder keine Prüfziffern  
verwendet oder es sich um eine spezielle Kontonummer handelt.");  
        (...)  
    }  
}
```

If the option to (temporarily) deactivate the validation is also to be used, it is advisable to combine the validation call with a call to the method istKontov
alidierungAktiviert() .

2. Source code example for validation

```
(...)  
  
if (org.nuclet.ckonto.CkontoUtils.istKontovalidierungAktiviert()) {  
    // Führe Validierung durch, wie im Beispiel zum Validierungaufruf beschrieben  
    // (...)  
}  
  
(...)
```

In this way, the validation can be easily switched on and off via the Nuclet parameter **ckonto Check** .

Version

Version	Date	Type	Changes
1.0.0	13.03.2013	initial version	-
1.1.0	13.12.2013	Migration to Nuclos 4.0	• Migration to Nuclos 4.0 • introduction of Nuclet parameters
1.2.0	24.03.2017	Migration to Nuclos 4.13	Migration to Nuclos 4.13
1.3.0	21.08.2017	Migration to Nuclos 4.18	Migration to Nuclos 4.18
1.4.0	13.09.2017	Migration to Nuclos 4.19	Migration to Nuclos 4.19

Related pages

Interfaces

Back to Interfaces

[Open](#)

Nuclet Wiki

Nuclet Wiki home

[Open](#)