

XIRR(noch nicht verfügbar)

 Sprache: **Deutsch** · [English](#)

XIRR(noch nicht verfügbar)

XIRR(noch nicht verfügbar) – Referenz zum Nuclet.

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Überblick](#)
 - [Integration](#)
 - [Versionshistorie](#)
 - [Verwandte Seiten](#)
-
- [XIRR\(noch nicht verfügbar\)](#)
 - [Überblick](#)
 - [Kurzbeschreibung](#)
 - [Nuclet-Bestandteile](#)
 - [Java-Package-Struktur](#)
 - [Integration](#)
 - [Notwendige Schritte zur Integration](#)
 - [Versionshistorie](#)
 - [Verwandte Seiten](#)
 - [Unterstützende Nuclets](#)
 - [Nuclet Wiki](#)

Name:	XIRR
Package:	org.nuclet.xirr
Version:	1.4.0
Datum:	27.09.2017
Nuclos-Kompatibilität:	ab Nuclos 4.19.3

Überblick

Kurzbeschreibung

Das XIRR-Nuclet bietet Funktionen zur Berechnung des internen Zinsfußes (Internal Rate of Return). Die Berechnung erfolgt auf der Grundlage einer Reihe von Zahlungen (Geldbeträge, gebunden an Datumswerte).

Die Berechnung entspricht den Excel-Funktionen XIRR (bzw. XINTZINSFUSS).



Die Berechnung fußt auf einem initialen Start bzw. Schätzwert. Für eine erfolgreiche Berechnung muss dieser innerhalb eines Konvergenzbereiches liegen.

Nuclet-Bestandteile

Das XIRR-Nuclet umfasst im Rahmen der .nuclet-Datei

- diverse Java-Regeln (verteilt auf Packages),
- eine Jobsteuerung (zur Steuerung eines Test-Jobs).

Typ	Name, englisch	Name, deutsch	Kurzbeschreibung
Java-Package	in.satpathy.financial		nuclet-intern genutzte XIRR-Funktionalität
	in.satpathy.math		nuclet-intern genutzte XIRR-Funktionalität

	org.nuclet.xirr.data		XIRR-Datenstrukturen
	org.nuclet.xirr.logic		Regelschnittstelle für XIRR-Funktionen
	org.nuclet.xirr.parameter		System-Parameter-Definitionen
	org.nuclet.xirr.test		Test-Routinen
Jobsteuerung	XIRR Test		Test-Job für den Aufruf von Test-Routinen

Tabelle 1: Nuclet-Bestandteile

Java-Package-Struktur

Die Java-Regeln sind in fünf Packages unterteilt. Von diesen Regeln sind im Normalfall nur vier Java-Klassen bei der Nuclet-Integration wirklich relevant:

- org.nuclet.xirr.data.XIRRSeries, als Datenstruktur für eine Serie von Zahlungen
- org.nuclet.xirr.logic.XIRRLogic, als Logik zum Aufruf der XIRR-Berechnungen
- org.nuclet.xirr.test.XIRRTTestJob, als Job zum Aufrufen von Tests
- org.nuclet.xirr.parameter.XIRRSysParameters, zur Konfiguration von System-Parametern

Integration

Notwendige Schritte zur Integration

Die Integration des XIRR-Nuclets erfolgt in 3 Schritten:

1. Download
2. Nuclet-Import
3. Konfiguration von System-Parametern
4. Anbindung des eigenen Java-Codes an die XIRR-Logik

Alle Integrationschritte werden im folgenden im Detail erläutert.

Schritt 1: Download

Download der ZIP-Datei „XIRR-v1.4.0-4.19.3.zip“ auf der Nuclos-Webpage unter „Nuclos Services“ > „Download“ > „Nuclet Download“. Das ZIP anschließend lokal entpacken.

Schritt 2: Nuclet-Import

Import des XIRR-Nuclets unter „Konfiguration“ > „Nuclet Management“ > „Importieren“ in Ihre bestehende Nuclos-Instanz, Auswahl der Datei „XIRR-v1.4.0-4.19.3.nuclet“

Schritt 3: Konfiguration der System-Parameter

Die Einrichtung des verwendeten Nuclos-Systemparamets erfolgt in der Version v1.0.0 über die Anpassung der Konstanten XIRR_DEFAULT_GUESS in der Klasse org.nuclet.xirr.parameter.XIRRSysParameters:

Parameter	Kurzbeschreibung
XIRR_DEFAULT_GUESS	Default-Startwert für die XIRR-Iterationen

Tabelle 2: System-Parameter



Anmerkung: Dieser Schritt wird in einer zukünftigen Version –d.h. sobald die neue Nuclos-API das Auslesen von System-Parametern unterstützt– nicht mehr über eine Anpassung des Source-Codes laufen.

Schritt 4: Anbindung des eigenen Java-Codes an die XIRR-Logik

Die Einbindung der XIRR-Logik erfolgt über die Initialisierung einer org.nuclet.xirr.data.XIRRSeries, die mit den Zahlungsbeträgen und Kalenderdaten der Zahlungen gefüllt wird. Optional ist ein spezifischer Startwert („guess“) mitzugeben. Falls kein Startwert angegeben ist, wird der Default-Startwert verwendet (s. Schritt 3). Einer Instanz der Klasse org.nuclet.xirr.logic.XIRRLogic wird sodann das Series-Objekt zur Berechnung übergeben.

Hier ein Beispiel zur Integration in einer UpdateRule:

```

@Rule(name="AktualisiereZahlungsverkehr", description="Aktualisiert den Zahlungsverkehr")
public class AktualisiereZahlungsverkehr implements UpdateRule
{
    public void update(UpdateContext context)
    {
        (...)

        XIRRLogic logic = new XIRRLogic(context); // Initialisierung der XIRR-Logik
        XIRRSeries series = new XIRRSeries(); // es ist möglich, den Startwert (guess) mizugegeben

        Zahlungsverkehr zvk = context.getBusinessObject(Zahlungsverkehr.class);
        List<Zahlung> lstZahlungen = zvk.getZahlung();

        for (Zahlung z : lstZahlungen) {
            series.add(z.getDatum(), z.getBetrag());
        }

        BigDecimal bgdXirrValue = logic.calculate(series); // ohne Konvergenz ist der Wert „null“!

        (...)
    }
}

```



Anmerkung: Siehe dazu auch die Methode test2() in der Klasse org.nuclet.xirr.test.XIRRTestJob.

Versionshistorie

Version	Datum	Typ	Änderungen
1.0.0	19.04.2013	initiale Version	-
1.1.0	13.12.2013	Umstellung auf Nuclos 4.0	• Umstellung auf Nuclos 4.0 • insbesondere: Umstellung auf Nuclet-Parameter
1.2.0	24.03.2017	Umstellung auf Nuclos 4.13	Umstellung auf Nuclos 4.13
1.3.0	22.08.2017	Umstellung auf Nuclos 4.18	Umstellung auf Nuclos 4.18
1.4.0	27.09.2017	Umstellung auf Nuclos 4.19	Umstellung auf Nuclos 4.19

Verwandte Seiten



Unterstützende Nuclets

Kategorie

[Öffnen](#)



Nuclet Wiki

Alle Kategorien

[Öffnen](#)