



OFCKS.NET Webdienst

Schnittstellendokumentation

Alle Rechte vorbehalten, auch die der photomechanischen Wiedergabe und der Speicherung in elektronischen Medien.

Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Weder die Firma enventa financial solutions GmbH noch die Autoren können für fehlerhafte Angaben und deren Folgen eine juristische Verantwortung oder sonstige Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir dankbar.

© Copyright enventa financial solutions GmbH, Stuttgart. Alle Rechte vorbehalten.

9202511



enventa financial solutions GmbH
Tränkestraße 11
70597 Stuttgart

Zentrale
+49 711 727246-0
Info.financial@enventa-group.com
www.enventa-group.com

enventa ProFI Support
+49 721 98593-53
profisupport.financial@enventa-group.com
www.enventa-group.com/produkt/profi

Inhalt

OFCKS.NET Interoperable Dienst-Schnittstelle	1
Übersicht zur interoperablen Schnittstelle des OFCKS.NET	1
Verwendungsmöglichkeiten	1
Installation, Konfiguration und Test	3
Installation	3
Voraussetzungen	3
Installation	3
Konfiguration Standard (ein Dienst)	3
Application.config	4
OFCKSNET.Dienst.exe.config	4
Dienst starten	4
Problem: Der Dienst startet nicht	5
Konfiguration mit mehreren Diensten	5
Zusätzlichen Dienst anlegen	5
Zusätzlichen Dienst verwalten	5
Zusätzlichen Dienst deinstallieren	6
Konfigurationsdateien sichern	6
Installation auf einem IIS	6
Grundeinrichtung als 32 Bit Anwendung	6
Einrichtung als 64 Bit Anwendung	7
Bedienung	7
Aufruf	7
Test einer lesenden Methode	8
Test einer speichernden Methode	8
Verwendung mit Sicherheitsschlüssel in einem Schnittstellen- Programm	10
Aufbau der Schnittstelle	11
Methoden	11
Datentypen	11
Fehler	12
Referenz der Methoden	13
AnzahlungGet	13
Aufrufparameter	13
Rückgabe	13
AnzahlungGetAZDebitorzahlung	13
Aufrufparameter	13
Rückgabe	14
AnzahlungGetAll	14
Aufrufparameter	14
Rückgabe	14
AnzahlungSave	14
Aufrufparameter	14
Rückgabe	15

AuftragGet	15
Aufrufparameter	15
Rückgabe	15
AuftragGetAll	16
Aufrufparameter	16
Rückgabe	16
AuftragSaveAll	16
Aufrufparameter	16
Rückgabe	17
AuswertungskreisGet	17
Aufrufparameter	17
Rückgabe	17
AuswertungskreisGetAll	17
Aufrufparameter	17
Rückgabe	17
BestellungGet	18
Aufrufparameter	18
Rückgabe	18
BestellungGetAll	18
Aufrufparameter	19
Rückgabe	19
BestellungSaveAll	19
Aufrufparameter	19
Rückgabe	20
BuchungskreisExistsNummer	20
Aufrufparameter	20
Rückgabe	20
BuchungskreisGet	20
Aufrufparameter	20
Rückgabe	20
BuchungskreisGetAll	20
Aufrufparameter	21
Rückgabe	21
DatenbankGetAll	21
Aufrufparameter	21
Rückgabe	21
DIIVersion	21
Aufrufparameter	21
Rückgabe	21
FremdwährungExistsKuerzel	21
Aufrufparameter	22
Rückgabe	22
FremdwährungGetAll	22
Aufrufparameter	22
Rückgabe	22
FremdwährungGetKurs	22
Aufrufparameter	22
Rückgabe	22
GetStatus	23
Aufrufparameter	23
Rückgabe	23
GultigesBuchDatum	23
Aufrufparameter	23
Rückgabe	23
HelloWorld	23
Aufrufparameter	23
Rückgabe	24

IsKoreEingerichtet	24
Aufrufparameter	24
Rückgabe	24
KontoBankGetAll	24
Aufrufparameter	24
Rückgabe	24
KontoBankGetSelected	25
Aufrufparameter	25
Rückgabe	25
KontoBankGetNewSepaMandatName	25
Aufrufparameter	25
Rückgabe	26
KontoBankSaveAll	26
Aufrufparameter	26
Rückgabe	26
KontoBankSaveThis	27
Aufrufparameter	27
Rückgabe	27
KontoBankSaveSepaMandat	27
Aufrufparameter	27
Rückgabe	28
KontoBankSepaMandatGetAll	28
Aufrufparameter	28
Rückgabe	28
KontoDelete	28
Aufrufparameter	29
Rückgabe	29
KontoExistsKontoKuerzel	29
Aufrufparameter	29
Rückgabe	29
KontoExistsKontoNummer	29
Aufrufparameter	29
Rückgabe	29
KontoGet	30
Aufrufparameter	30
Rückgabe	30
KontoGetAll	33
Aufrufparameter	33
Rückgabe	33
KontoGetAllKennung	33
Aufrufparameter	33
Rückgabe	34
KontoGetAllStatus	34
Aufrufparameter	34
Rückgabe	34
KontoGetBelegzeilen	34
Aufrufparameter	34
Rückgabe	35
KontoGetBelegzeilenAbBuchNr	35
Aufrufparameter	35
Rückgabe	36
KontoGetKennung	36
Aufrufparameter	36
Rückgabe	36
KontoGetListe	36
Aufrufparameter	36
Rückgabe	36

KontoGetNextFreeNo	37
Aufrufparameter	37
Rückgabe	37
KontoGetPeriodenSaldo	37
Aufrufparameter	37
Rückgabe	37
KontoGetSaldoMitStapel	38
Aufrufparameter	38
Rückgabe	38
KontoGetWaWiKennung	38
Aufrufparameter	38
Rückgabe	38
KontoGetZahlverhalten.....	38
Aufrufparameter	38
Rückgabe	39
KontoSave.....	39
Aufrufparameter	39
KontoSetStatus	42
Aufrufparameter	42
KostenArtExistsKontoKuerzel	43
Aufrufparameter	43
Rückgabe	43
KostenArtExistsKontoNummer	43
Aufrufparameter	43
Rückgabe	43
KostenArtGet	43
Aufrufparameter	43
Rückgabe	44
KostenArtGetOE	44
KostenArtGetAll.....	44
Aufrufparameter	44
Rückgabe	44
KostenArtSave.....	44
Aufrufparameter	44
Rückgabe	45
KostenBuchungSave.....	45
Aufrufparameter	45
Rückgabe	46
KostenBuchungSaveAndBook	46
Aufrufparameter	46
Rückgabe	46
KostenDimensionGetAll	46
Aufrufparameter	46
Rückgabe	47
KostenStelleExistsKuerzel.....	47
Aufrufparameter	47
Rückgabe	47
KostenStelleExistsNummer	47
Aufrufparameter	48
Rückgabe	48
KostenStelleGet.....	48
Aufrufparameter	48
Rückgabe	48
KostenStelleGetOE.....	48
KostenStelleGetAll	49
Aufrufparameter	49
Rückgabe	49

KostenStelleGetPeriodenSaldo	49
Aufrufparameter	49
Rückgabe	49
KostenStelleSave	50
Aufrufparameter	50
Rückgabe	50
KostenSchabloneGet	50
Aufrufparameter	50
Rückgabe	50
KostenSchabloneGetAll	51
Aufrufparameter	51
Rückgabe	51
KostenTraegerExistsKuerzel	51
Aufrufparameter	51
Rückgabe	51
KostenTraegerExistsNummer	51
Aufrufparameter	52
Rückgabe	52
KostenTraegerGet	52
Aufrufparameter	52
Rückgabe	52
KostenTraegerGetAll	53
Aufrufparameter	53
Rückgabe	53
KostenTraegerGetPeriodenSaldo	53
Aufrufparameter	53
Rückgabe	53
KostenTraegerSave	54
Aufrufparameter	54
Rückgabe	54
KostenTraegerNExistsNummer	54
Aufrufparameter	54
Rückgabe	55
KostenTraegerNGet	55
Aufrufparameter	55
Rückgabe	55
KostenTraegerNGetAll	55
Aufrufparameter	55
Rückgabe	56
MandantGet	56
Aufrufparameter	56
Rückgabe	56
MandantGetAdressDaten	57
Aufrufparameter	57
Rückgabe	57
MandantGetAll	57
Aufrufparameter	57
Rückgabe	58
MandantSave	58
Aufrufparameter	58
Rückgabe	58
MandantSaveAdressDaten	58
Aufrufparameter	58
Rückgabe	59
OPGetDeb	59
Aufrufparameter	59
Rückgabe	59

OPGetDebBeleg.....	60
Aufrufparameter	60
Rückgabe	60
OPGetDebOPBeleg	60
Aufrufparameter	60
Rückgabe	61
OPGetKred	61
Aufrufparameter	61
Rückgabe	61
OPGetKredBeleg	62
Aufrufparameter	62
Rückgabe	62
OPGetKredOPBeleg	62
Aufrufparameter	62
Rückgabe	62
OPGetVorgaenge	62
Aufrufparameter	63
Rückgabe	63
OPStatistikDebTop100	64
Aufrufparameter	64
Rückgabe	64
OPStatistikDebTop100Faellig	64
Aufrufparameter	64
Rückgabe	64
OPStatistikDebTop100FaelligMinWert	64
Aufrufparameter	64
Rückgabe	65
OPStatistikDebTop100Gemahnt	65
Aufrufparameter	65
Rückgabe	65
OPStatistikDebTop100GemahntStufe	65
Aufrufparameter	65
Rückgabe	65
OPStatistikDebTop100MinWert	65
Aufrufparameter	66
Rückgabe	66
OPStatistikDebTop100Posten	66
Aufrufparameter	66
Rückgabe	66
OPStatistikSaldoFristen	66
Aufrufparameter	66
Rückgabe	67
OPStatistikSaldoGesamt	67
Aufrufparameter	67
Rückgabe	67
OPStatistikSaldoMahnstufen	67
Aufrufparameter	67
Rückgabe	68
OPStatistikSaldoWochen	68
Aufrufparameter	68
Rückgabe	68
RechnungGet	68
Aufrufparameter	68
Rückgabe	69
RechnungGetAll	70
Aufrufparameter	70
Rückgabe	70

RechnungGetOpen	70
Aufrufparameter	70
Rückgabe	70
RechnungGetBelegNr	70
Aufrufparameter	70
Rückgabe	71
RechnungGetBelegNrDok.....	71
Aufrufparameter	71
Rückgabe	71
RechnungGetBelegKonto	71
Aufrufparameter	71
Rückgabe	71
RechnungSetBelegNrGeloesch	72
Aufrufparameter	72
Rückgabe	72
RechnungSetGeloesch	72
Aufrufparameter	72
Rückgabe	72
SpracheExistsNummer	72
Aufrufparameter	72
Rückgabe	73
SpracheGet	73
Aufrufparameter	73
Rückgabe	73
SpracheGetAll	73
Aufrufparameter	73
Rückgabe	73
StapelBuchungGetAll	73
Aufrufparameter	74
Rückgabe	74
StapelBuchungSave	74
Aufrufparameter	74
Rückgabe	79
StapelBuchungSaveAndBook	79
Aufrufparameter	79
Rückgabe	79
StapelDelete	80
Aufrufparameter	80
Rückgabe	80
StapelExistsBelegNr.....	80
Aufrufparameter	80
Rückgabe	80
StapelExistsNummer	80
Aufrufparameter	81
Rückgabe	81
StapelGet	81
Aufrufparameter	81
Rückgabe	81
StapelGetAll	81
Aufrufparameter	82
Rückgabe	82
StapelSave	82
Aufrufparameter	82
Rückgabe	82
StapelCheckAndBook	83
Aufrufparameter	83
Rückgabe	83

SteuerArtExistsNummer	83
Aufrufparameter	83
Rückgabe	83
SteuerArtGetAll.....	83
Aufrufparameter	84
Rückgabe	84
SteuerArtGetAllSatz	84
Aufrufparameter	84
Rückgabe	84
SteuerSatzExistsNummer.....	84
Aufrufparameter	85
Rückgabe	85
Version	85
Aufrufparameter	85
Rückgabe	85
ZahlzielDelete	85
Aufrufparameter	85
Rückgabe	85
ZahlzielExistsKuerzel	85
Aufrufparameter	86
Rückgabe	86
ZahlzielExistsNummer.....	86
Aufrufparameter	86
Rückgabe	86
ZahlzielGet	86
Aufrufparameter	86
Rückgabe	86
ZahlzielGetAll	87
Aufrufparameter	87
Rückgabe	87
ZahlzielSave.....	87
Aufrufparameter	88
Rückgabe	88

Index

89

OFCKS.NET Interoperable Dienst-Schnittstelle

Übersicht zur interoperablen Schnittstelle des OFCKS.NET

Das **Open Finance Connectivity Kit** OFCK ist die primäre Schnittstelle zum Programm enventa ProFI (auch unter anderen Namen bei OEM-Herstellern bekannt) und dessen Daten. Integratoren können damit Daten aus der Buchhaltung abfragen (z.B. Sachkonten, Steuerschlüssel, Zahlungsbedingungen), Stammdaten mit denen in der Finanzbuchhaltung abgleichen (z.B. Kunden – Debitoren, Lieferanten – Kreditoren) oder Daten an die Buchhaltung übergeben (vor allem Buchungen).

OFCKS.NET stellt 2 Schichten der Schnittstelle zur Verfügung: es kann als .NET Assembly direkt oder als COM-Server angesprochen werden. Die Verwendungslogik ist auf direkte Einzelzugriffe über Objekte und ihre Eigenschaften ausgerichtet. Für eine interoperable, lose Kopplung eignen sich diese Schnittstellen nicht.

Die **Webdienst-Schnittstelle** setzt auf OFCKS.NET auf und stellt eine interoperable Kopplung derzeit mittels SOAP-Zugriffen zur Verfügung. Der Dienst besteht aus der ASP.NET Webseite mit Code-Behind und einem Windows Dienst als Host.

Verwendungsmöglichkeiten

Der Zugriff erfolgt über SOAP (http) wie bei Webdiensten üblich und ist plattformübergreifend möglich. Es werden SOAP 1.1 und 1.2 unterstützt. Die WS*-Services werden nicht unterstützt.

Ab Version 5.1.4 kann der Zugriff mit einem Sicherheitsschlüssel abgesichert werden.

Installation, Konfiguration und Test

Installation

Voraussetzungen

Der Webdienst kann auf einer Arbeitsstation installiert werden. Der typische Anwendungsbereich liegt jedoch in einer Server-Installation.

Voraussetzungen sind:

- .NET Framework 4.8 32Bit.
- OFCKS.NET
- Für die unterstützten Datenbanken Oracle und Informix deren Treiber in der 32Bit-Version. Installiert sein muss der .NET Provider.

Installation

Die Webdienst-Schnittstelle ist Bestandteil des Setups für OFCKS.NET OFCKSNETInstall.zip (mit vorangestellter Versionsnummer). Um die Webdienst-Schnittstelle zu installieren, gehen Sie in die benutzerdefinierte Installation und wählen das Paket zur Installation an.

Konfiguration Standard (ein Dienst)

Dieser Abschnitt beschreibt die Konfiguration der Standardinstallation mit nur einem (Windows-)Dienst. Die Konfiguration geschieht in mehreren Config-Dateien:

- sql.config: Konfiguration des Datenbankzugriffs
- Application.config: Konfiguration der Anwendung
- OFCKSNET.Dienst.exe.config: Konfiguration des Webdienstes
- web.config: Konfiguration aller Webdienste

Die Datei sql.config benötigen Sie auch für die Oberflächenprogramme von enventa ProFI. Kopieren Sie diese aus dem enventa ProFI Installationsverzeichnis ins Installationsverzeichnis des Webdienstes.

Dort stehen auch die anderen Konfigurationsdateien. Sie können mit einem Editor bearbeitet werden, es steht teilweise jedoch auch ein spezielles Konfigurationsprogramm zur Verfügung.

An der Datei web.config müssen üblicherweise keine Änderungen vorgenommen werden.

Application.config

Nutzen Sie zur Bearbeitung die Anwendung **ServiceConfig.exe**. Öffnen Sie die Konfigurationsdatei im Menüpunkt Datei – Öffnen. Tragen Sie folgendes ein:

- Datenbankname: Der logische Name der Datenbank, wie in der Datei sql.config im Tag <database name="..."> eingetragen.
- Benutzer: Username in der Datenbank, unter dem die Anmeldung für den Webdienst erfolgen soll.
- Kennwort: Passwort des Benutzers in der Datenbank.
- Trace-Level: Im Normalfall ist kein Tracing notwendig, Fehler werden trotzdem in die Trace-Datei geschrieben. Es können bei Bedarf andere Trace-Level gewählt werden.
- Tracedatei: Tragen Sie hier den Pfad und Dateinamen der Trace-datei ein.
- Verbindungstyp: Hier ist der Typ der Verbindung zu wählen. Zur Auswahl stehen „Einmal“ (es wird beim Start des Service einmalig eine DB-Verbindung hergestellt) und „Immer“ (es wird für jede Anfrage eine DB-Verbindung hergestellt).
- Security Schlüssel (ab Version 5.1.4.0): Wenn der Zugriff abgesichert werden soll, tragen Sie hier ein "Geheimniswort" ein. Die Eingabe wird verschlüsselt in die Datei Application.config eingetragen. Der Aufruf ist dann nur noch mit dem verschlüsselten Security Schlüssel möglich, den Sie abspeichern können. Er wird auch beim nächsten Öffnen im Feld darüber angezeigt.

Die Datei Application.config wird leer ausgeliefert und muss immer konfiguriert werden.

OFCKSNET.Dienst.exe.config

Falls die Standardwerte geändert werden sollen, kann dies in der Konfigurationsanwendung **ServiceConfig.exe** im Menü Service – Dienstkonfiguration geschehen.

- Virtuelles Verzeichnis: Tragen Sie hier den Namen des virtuellen Verzeichnisses ein, auf welchem die Anwendungsdomäne ausgeführt wird. Dieser Verzeichnisname ist Bestandteil des Aufruf-Links.
- Port: Tragen Sie hier den Port ein. Im Standard wird Port 82 vorgeschlagen.
- Nur diese IP: In einigen Konstellationen (z.B. mehrere Netzwerkkarten) muss gezielt die IP-Adresse gewählt werden, über die der Webdienst angesprochen werden soll.
- Event-Level: Hier ist der Grad der Ereignisaufzeichnung in der Windows-Ereignisanzeige zu wählen. Zur Auswahl stehen „Keine Ereignisse“ (es werden keine Ereignisse protokolliert), „Nur Fehler-Ereignisse“ (es werden nur Ereignisse im Falle eines Fehlers protokolliert) und „Alle Ereignisse“ (es werden Ereignisse jeglicher Art protokolliert). Die empfohlene Einstellung lautet „Nur Fehler-Ereignisse“.

Dienst starten

In der Windows Dienstverwaltung trägt sich der Dienst zum automatischen Start ein.

Falls der Dienst manuell gesteuert werden soll, gibt es 2 Möglichkeiten:

- Sie starten und beenden den Dienst in der Windows-Systemsteuerung - Verwaltung – Dienste. Der Dienst heißt OFCKS.NET Dienst.

- Sie verwenden die Konfigurationsanwendung **ServiceConfig.exe** mit dem Menü Service – Dienststeuerung. Im Normalfall muss die Anwendung als Administrator gestartet werden, damit eine Steuerung des Dienstes zur Wirkung kommt.

Problem: Der Dienst startet nicht

Nach Erstinstallation kann es vorkommen, dass der Dienst nicht startet. Im Ereignisprotokoll findet sich u.U. ein Fehler FileNotFoundException. Grund ist, dass Windows ein notwendiges Verzeichnis für temporäre Dateien des Webdienstes nicht in jedem Fall anlegt.

In der Anwendung **ServiceConfig** gibt es (ab Version 5.1.0.2) den Menüpunkt Service -> Temporärpfad überprüfen. Das Verzeichnis wird bei Bedarf angelegt. Wenn es in Ihrer Konstellation so ist, dass die Anwendungssteuerung das Anlegen eines Verzeichnisses unter Windows nicht ohne weiteres zulässt, gelingt dies nur, wenn Sie die Anwendung **ServiceConfig** "als Administrator" starten. Darauf weist ggf. die Fehlermeldung hin.

Konfiguration mit mehreren Diensten

Dieser Abschnitt beschreibt die Konfiguration, wenn mehrere Dienste parallel konfiguriert werden sollen. Das kann z.B. gewünscht sein, um einen Dienst für die produktive Datenbank und einen Dienst für die Testdatenbank zu führen. Oder zwei Dienste für zwei Datenbanken von rechtlich unabhängigen Tochterfirmen.

Das Prinzip ist:

- Alle Dienste werden aus nur einem Installationsverzeichnis administriert.
- Jeder Dienst bekommt einen Namen.
- Es gibt neben den Standard-Konfigurationsdateien getrennte Konfigurationsdateien Application.*name*.config und OFCKSNET.Dienst.*name*.exe.config

Zusätzlichen Dienst anlegen

Öffnen Sie die Anwendung **ServiceConfig** als Administrator. Öffnen Sie den Menüpunkt Service -> Dienstkonfiguration. Öffnen Sie mit der Schaltfläche "Neu" die Maske für einen neuen Dienst. Geben Sie diesem Dienst einen Namen: Sie sehen, welche Datei- und Dienstnamen sich daraus ergeben.

Wählen Sie die Schaltfläche "Service installieren". Der Dienst wird in der Windows-Dienstverwaltung eingetragen. Sie können anschließend ein Protokoll dieses Prozesses öffnen. Starten Sie den Dienst jetzt noch nicht, falls Sie danach gefragt werden: Sie müssen zunächst die Konfigurationsparameter abändern (siehe Folgeabschnitt).

Zusätzlichen Dienst verwalten

In Masken zur Verwaltung der Dienste sehen Sie, welche Dienste in der Windows Dienstverwaltung eingetragen sind. Wählen Sie jeweils den passenden Dienst aus.

Mit den Menüpunkten Datei -> öffnen und Service -> Dienstkonfiguration verwalten Sie die Einträge in Config-Dateien. Die benannten Config-Dateien müssen im Aufrufverzeichnis liegen.

Sollten Sie mehrere Installationsverzeichnisse haben, können Sie nur jeweils die Dateien des aktuellen Aufverzeichnisses bearbeiten.

Im Menüpunkt Test -> Service testen müssen Sie, wenn Sie mehrere Dienste führen, die richtige URL auswählen.

Zusätzlichen Dienst deinstallieren

Falls Sie einen Dienst nicht mehr benötigen: Gehen Sie in den Menüpunkt Service -> Dienststeuerung. Klicken Sie auf den Dienst, den Sie deinstallieren möchten. Wählen Sie die Schaltfläche "Deinstallieren".

Konfigurationsdateien sichern

Um Ihre Einstellungen auf Dateiebene zu sichern (z.B. vor einem Update der Schnittstelle), kopieren Sie alle Dateien Application*config und alle Dateien OFCKSNET.Dienst*exe.config in ein Sicherungsverzeichnis.

Installation auf einem IIS

Grundeinrichtung als 32 Bit Anwendung

Der OFCKS.NET Webdienst ist im Standard dafür vorgesehen, als Windows-Dienst zu laufen. Er nutzt dabei einen in Windows eingebauten HTTP-Server. Es ist jedoch auch möglich, ihn auf einem IIS-Server (Microsoft Windows Internetinformationsdienste) zu installieren (ab Version 5.1.4.0). Da dies nicht die Standardinstallation ist, sind hier die wenigen manuellen Schritte beschrieben, die dazu notwendig sind.

1. Installieren und konfigurieren Sie den Webdienst als Windows-Dienst wie oben beschrieben auf einem beliebigen Rechner. Dies geschieht nur, um ein fertig eingerichtetes Verzeichnis zu erhalten.
2. Richten Sie im IIS-Manager eine Website ein. Die Bindung der Website können Sie mit http und Port 82 einrichten wie für den Windows-Dienst im Standard, aber auch nach Belieben anders.
3. Die Website sollte über ihren eigenen Anwendungspool verfügen, das schlägt der IIS-Manager im Standard so vor.
4. Gehen Sie in die erweiterten Einstellungen des Anwendungspools. Setzen Sie "32-Bit-Anwendungen aktivieren" auf True.
5. Kopieren Sie alles aus dem in Schritt 1 eingerichteten Verzeichnis einschließlich Unterverzeichnis bin in das Verzeichnis dieser Website. Einige Dinge sind dabei eigentlich nicht notwendig, wie vor allem die OFCKSNET.Dienst.exe und ihre Config-Datei.
6. Starten Sie den Anwendungspool (neu) und die Website.

Der Testaufruf in einem Browser erfolgt, anders als unten für den Windows-Dienst beschrieben, ohne den Teil "/syskasoap3". Wenn für Port 82 eingerichtet, lautet der Testaufruf

<http://127.0.0.1:82/OFCKSNETWebDienst.asmx>

Oder Sie verwenden den Servernamen statt der IP-Adresse 127.0.0.1.

Einrichtung als 64 Bit Anwendung

Der Webdienst kann auch als 64 Bit Anwendung laufen, benötigt dann jedoch von einer Datei die 64 Bit Variante. Es handelt sich um die Oracle.DataAccess.dll, die Sie ggf. in der passenden Ausführung erhalten können. Im Anwendungspool ist dann die Einstellung "32-Bit-Anwendungen aktivieren" wieder auf false zu setzen.

Bedienung

Der Webdienst wird grundsätzlich von einer anderen Anwendung genutzt. Er kann lokal auf dem Computer, auf dem er installiert ist, mit einer Testoberfläche des .NET Framework getestet werden. Dies geschieht in einem Internet Browser wie dem Internet Explorer oder Firefox.

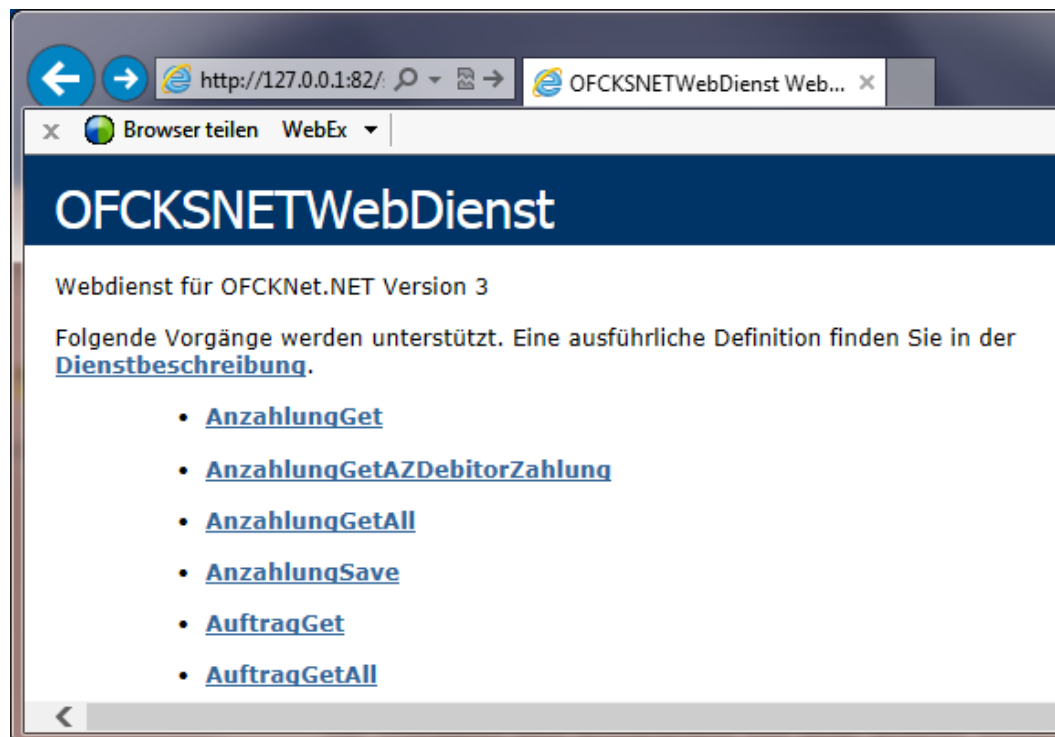
Aufruf

Der Aufruf geschieht mit einer Adresse, die Sie sich für lokale Verwendung von der Konfigurationsanwendung **ServiceConfig.exe** geben lassen können: Rufen Sie den Menüpunkt Test – Service aufrufen auf und klicken Sie auf die zusammengestellte Adresse. Diese können sie sich auch als Lesezeichen im Browser hinterlegen.

Die Standardadresse für lokale Verwendung lautet:

`http://127.0.0.1:82/syskasoap3/OFCKSNETWebDienst.asmx`

Es erscheint eine Oberfläche, die die möglichen Funktionen auflistet:

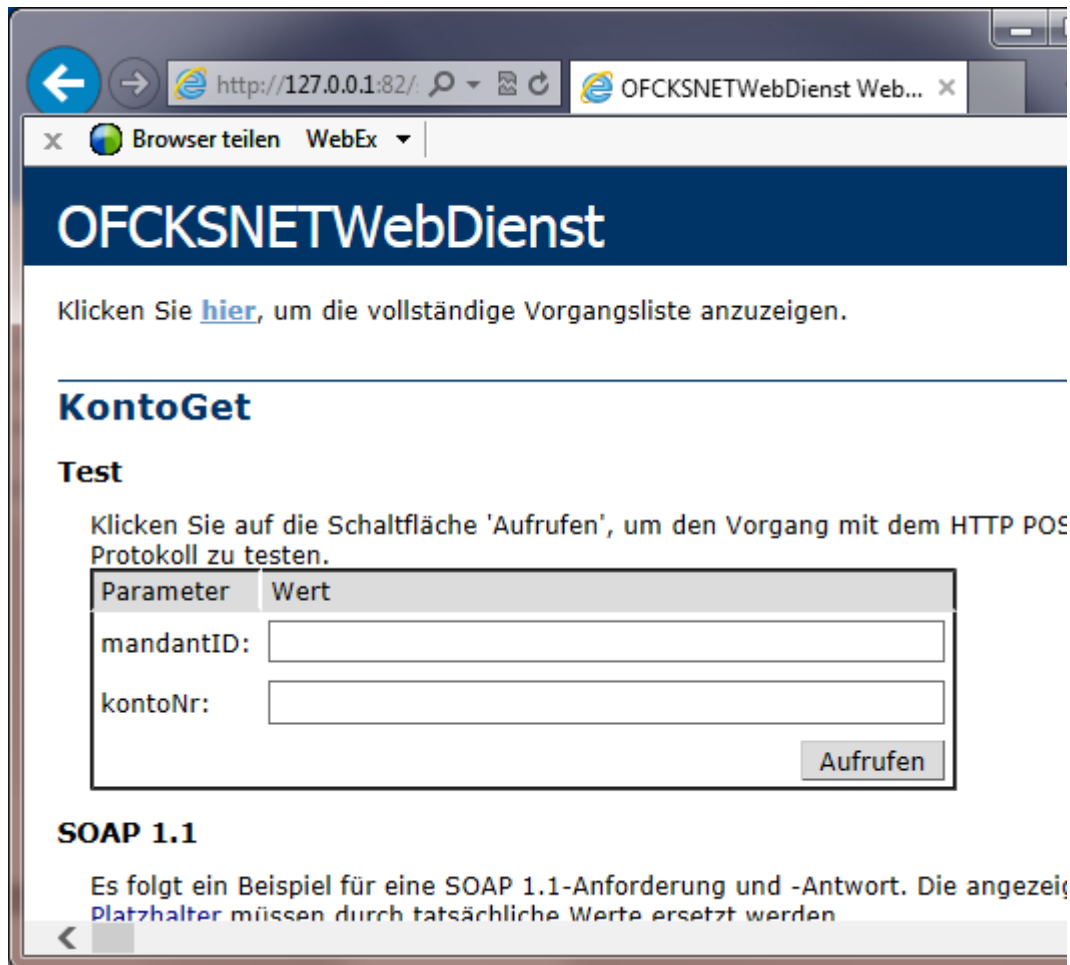


Unter den Funktionen kann die Funktion HelloWorld ohne weitere Aufrufparameter und ohne Datenbankverbindung als Funktionstest des Dienstes verwendet werden.

Test einer lesenden Methode

Funktionen, die mit wenigen Parametern mit Standard-Datentypen aufgerufen werden, können direkt getestet werden.

Beispiel KontoGet:



The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1:82/'. The page title is 'OFCKSNETWebDienst Web...'. The main heading is 'OFCKSNETWebDienst'. Below it, a message says 'Klicken Sie [hier](#), um die vollständige Vorgangsliste anzuzeigen.' The section 'KontoGet' is highlighted. Under 'Test', it says 'Klicken Sie auf die Schaltfläche 'Aufrufen', um den Vorgang mit dem HTTP POST Protokoll zu testen.' Below this is a form with two input fields: 'mandantID:' and 'kontoNr:'. A button labeled 'Aufrufen' is at the bottom right of the form. Below the form, the section 'SOAP 1.1' is shown, with a note: 'Es folgt ein Beispiel für eine SOAP 1.1-Anforderung und -Antwort. Die angezeigten Platzhalter müssen durch tatsächliche Werte ersetzt werden'.

Parameter	Wert
mandantID:	
kontoNr:	

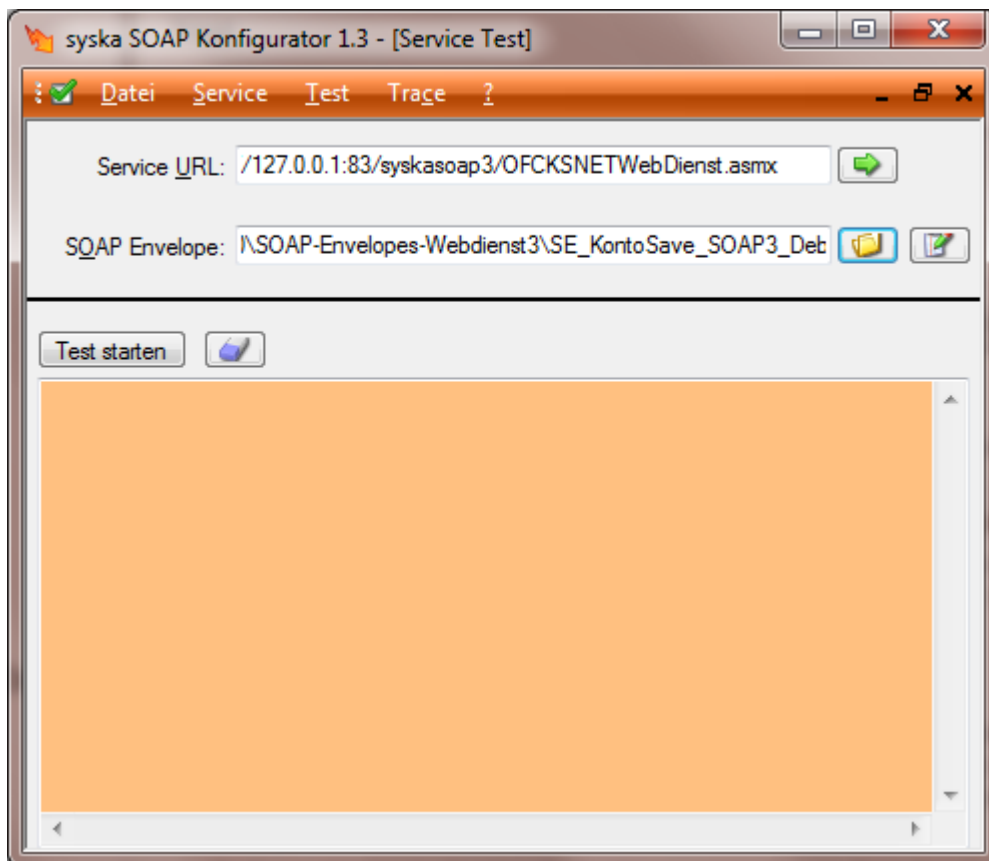
Aufrufen

Geben Sie die Mandantennummer und die Kontonummer ein, um sich die Daten eines Kontos anzeigen zu lassen. Das Ergebnis erscheint im XML-Format.

Test einer speichernden Methode

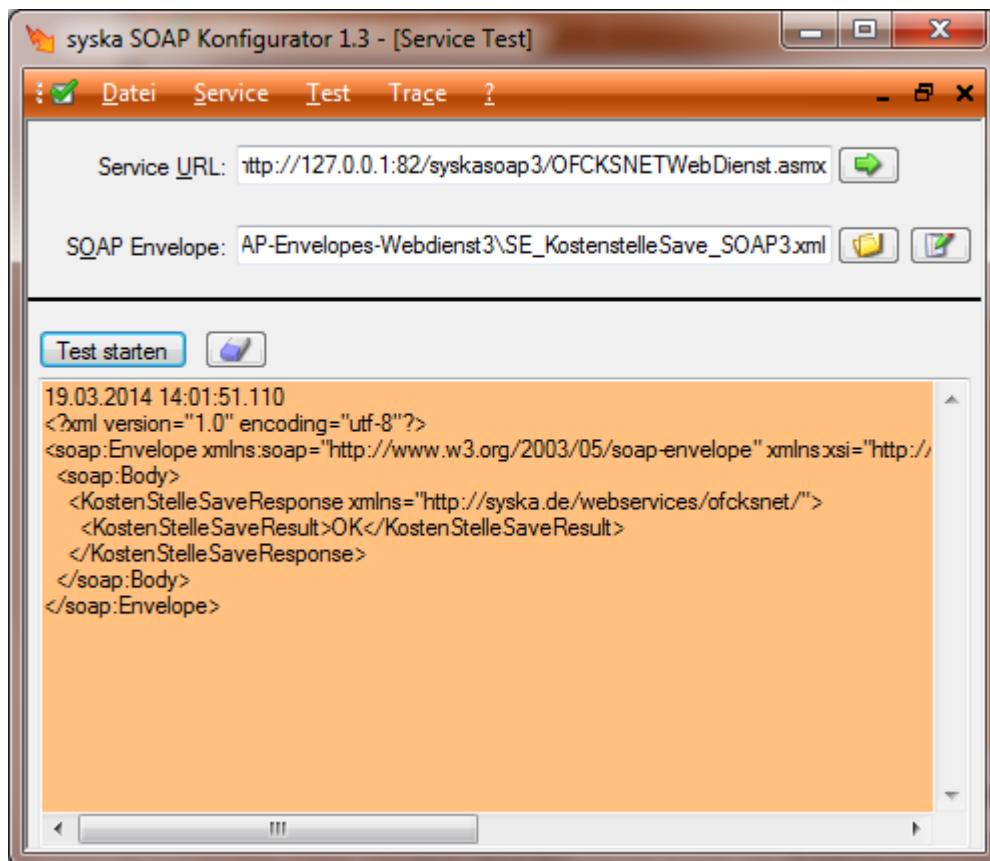
Bei entsprechender Sachkenntnis können Datenstrukturen für die Speicherung z.B. eines Kontos oder eines Buchungsbelegs in XML-Dateien, so genannten SOAP-Envelopes aufgebaut werden. Mit der Installation werden einige Envelopes mit ausgeliefert.

Der Aufruf geschieht in der Konfigurationsanwendung **ServiceConfig.exe** unter dem Menüpunkt Test – Service testen.



Die Service-URL sollte schon automatisch vorbelegt sein. Ein SOAP-Envelope muss ausgewählt werden. Der Envelope kann über ein Icon auch in einem Editor bearbeitet werden.

Benutzen Sie anschließend die Schaltfläche "Test starten". Sie erhalten das Ergebnis im Ausgabefenster in XML-Darstellung.



Verwendung mit Sicherheitsschlüssel in einem Schnittstellen-Programm

Wurde ein Sicherheitsschlüssel konfiguriert wie im Abschnitt "[Application.config](#)" beschrieben, gilt für ein verwendendes Schnittstellenprogramm:

- Der WSDL-Importer generiert eine class SecurityHeader : System.Web.Services.Protocols.SoapHeader.
- Legen Sie eine Instanz der Klasse an.
- Füllen Sie deren Element Key mit dem verschlüsselten Geheimniswort.
- Weisen Sie diese Instanz der Eigenschaft SecurityHeaderValue zu.

Aufbau der Schnittstelle

Methoden

Der OFCKS.NET Webdienst bietet den Zugriff auf die Objekte des Rechnungswesens über Methodenaufrufe an, die einzelne Parameter sowie in den meisten Fällen auch eine komplexere Datenstruktur erwarten oder zurückgeben. Die Collections des OFCKS.NET werden in der Regel mit folgenden Standardmethoden angeboten:

- Einen GetAll-Zugriff auf alle Datensätze oder gefilterte Untermengen.
- Ein Get-Zugriff für einen einzelnen Datensatz.
- Ein Save-Zugriff für Anlegen und Ändern eines Datensatzes.
- Ein Delete-Zugriff, wo erlaubt.

Bei den Save-Zugriffen für Stammdaten ist in der Regel das Tag IsNew zu beachten. Darüber wird gesteuert, ob aus Sicht des verwendenden Programms eine Neuanlage oder eine Änderung gewünscht ist.

Es wird nicht die vollständige Funktionalität des OFCKS.NET angeboten. Die Schnittstelle sollte jedoch ausreichend umfangreich sein.

Datentypen

Die Datentypen sind grundsätzlich die Standard-XML-Datentypen. Es gibt jedoch einige Besonderheiten:

Integer statt Bool: In der früheren Version des Webdienstes wurden keine Bool-Felder verwendet, sondern stattdessen Int-Felder. Nicht alle Felder, die in Wirklichkeit Bool-Inhalte repräsentieren, wurden umgestellt.

Nullable-Felder: In manchen Fällen ist die Information relevant, ob ein Feldinhalt 0 (bei numerischem) Feld ist oder kein Inhalt gesetzt ist. Bei Datumsfeldern wurde in der früheren Version ein C++-Datum 0 (30.12.1899) als leeres Datum betrachtet. Stattdessen werden jetzt in der Regel Nullable-Felder genutzt. Dies gilt jedoch nicht durchgängig. Insbesondere bei numerischen Feldern ist es häufig auch so, dass kein Bedeutungs-Unterschied zwischen einem Inhalt 0 und NULL besteht und deshalb die Schnittstelle kein Nullable-Feld nutzt.

Enums und Integer: Die Schnittstelle verwendet an etlichen Stellen Enum-Felder. Dies kann jedoch zu Problemen in der verwendenden Software führen, wenn ein Enum-Wert später hinzu kam und zum Zeitpunkt der Erstellung der Schnittstelle noch nicht bekannt war. An einigen Stellen gibt es deshalb zu Get-Funktionen parallele Funktionen mit der Endung OE (Ohne Enum). In dieser Funktion werden dann der Integer- statt dem Enum-Wert dargestellt.

Datumsfelder: In der früheren Version wurden alle Datumsfelder als string-Typen definiert. Sie können je nach Situation leer bleiben oder werden mit einem Datum gemäß Ländereinstellungen gefüllt.

Viele Felder sind auf den XML-Datentyp DateTime umgestellt und erwarten ein Datum im XML-Format YYYY-MM-TT, u.U. auch mit Zeitstempel.

Fehler

Fehler werden als SOAPException zurückgegeben. Im Tag <soap:Reason> ist die Fehlermeldung enthalten einschließlich Stacktrace.

Referenz der Methoden

Die Referenz der Methoden geht die Methoden in alphabetischer Reihenfolge durch.

AnzahlungGet

Gibt die Daten eines Anzahlungsvorfalles zurück.

Anzahlungsvorfälle organisieren Buchungen im Anzahlungsverfahren von der Abschlagsrechnung bis zur Bezahlung der Schlussrechnung. Anzahlungsvorfälle können auch automatisch mit der ersten Buchung auf einen Vorfall angelegt werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.
- <anzahlungID>int</anzahlungID>: Nummer des Anzahlungsvorfalles.

Rückgabe

- anmerkung: Ausführliche Anmerkung zum Anzahlungsvorfall.
- anzahlungID: Vom System vergebene numerische ID des Anzahlungsvorfalles.
- bezeichnung: Bezeichnung des Anzahlungsvorfalles, der vom Vorsystem vergebene Schlüssel.
- buchungskreisID: Buchungskreisnummer

AnzahlungGetAZDebitorZahlung

Gibt die Zahlungen zu einem Anzahlungsvorfall zurück. Dies dient als Rückmeldung für das ERP-System über die geleisteten Anzahlungen. Das ERP-System will diese evtl. bei der Erstellung der Schlussrechnung berücksichtigen.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.
- <anzahlungID>int</anzahlungID>: Nummer des Anzahlungsvorfalles.

Rückgabe

- anzahlungID: Vom System vergebene numerische ID des Anzahlungsvorfalls.
- belegNr: Fibu-Belegnummer
- belegRefID: (Interne fortlaufende) Buchungsnummer
- bruttoBetrag: Betrag der Anzahlung (inkl. USt) in OP-Währung
- eigenBetrag: Betrag in Mandantenwährung
- eigenSteuerBetrag: Steuerbetrag in Mandantenwährung
- opBelegNr: Rechnungsnummer des OPs
- steuerBetrag: Steuerbetrag in OP-Währung
- zahlungsDatum: Belegdatum der Zahlung

AnzahlungGetAll

Liefert die Daten aller Anzahlungsvorfälle eines Mandanten.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.

Rückgabe

Array von Anzahlungen wie unter AnzahlungGet beschrieben.

AnzahlungSave

Speichert einen Anzahlungsvorfall – Anlegen oder Ändern.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.
- <data>-Struktur mit den Feldern:
- <anzahlungID>int</anzahlungID>: Vom System vergebene numerische ID des Anzahlungsvorfalls. Zur Neuanlage 0 mitgeben.
- <anmerkung>string</anmerkung>: Ausführliche Anmerkung zum Anzahlungsvorfall.
- <bezeichnung>string</bezeichnung>: Bezeichnung des Anzahlungsvorfalls, der vom Vorsystem vergebene Schlüssel.
- <buchungskreisID>int</buchungskreisID>: Buchungskreisnummer, im Standardfall 1.

Rückgabe

- `returnCode`: Enum der möglichen Returncodes.

AuftragGet

Stellt die Daten eines Auftrags zur Verfügung.

Zahlungsdaten eines Warenwirtschaft-Auftrags für die Liquiditätsplanung. Zur Übergabe von Auftragsdaten an die Liquiditätsplanung der Chefinfo (falls nicht ein View in das ERP-System genutzt wird). Die Verwendung dieser Funktion setzt voraus, dass eine Tabelle „cif_li_auftrag“ vorhanden ist, vgl. Schnittstellendokumentation zur Liquiditätsplanung. Informationen zur Übergabe der einzelnen Felder siehe ebenfalls Schnittstellendokumentation zur Liquiditätsplanung.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer.
- `<auftragID>int</auftragID>`: Interne Nummer des Auftrags-Datensatzes.

Rückgabe

- `fremdID`: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- `belegDatum`: Belegdatum, voraussichtliches Rechnungsdatum
- `nettoFaellig`: Fälligkeitsdatum
- `steuerFaellig`: Fälligkeit der Umsatzsteuer
- `zahlZiel`: Zahlungsziel
- `eigenBrutto`: Bruttobetrag in Eigenwährung
- `fremdBrutto`: Bruttobetrag in Fremdwährung
- `waehrung`: (Fremd-)Währung der Buchung
- `kurs`: Umrechnungskurs
- `einheit`: Einheit zum Kurs, falls nicht 1
- `eigenSteuer`: Steuerbetrag in Eigenwährung
- `fremdSteuer`: Steuerbetrag in Fremdwährung
- `steuerProzent`: Umsatzsteuer-Prozentsatz
- `herkunft`: Herkunftskennung Ihrer Anwendung
- `bemerkung`: Buchungstext, Vorgangstext
- `linkInfo`: Pfad zum PDF mit dem Auftrag
- `konto`: Personenkonto

AuftragGetAll

Liefert alle Aufträge aus cif_li_auftrag

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.

Rückgabe

Array von `<AuftragData>` wie unter `AuftragGet` beschrieben.

AuftragSaveAll

Speichert das Array von Aufträgen und löscht ggf. zuvor. Gedacht für eine paketweise Übergabe, beim 1. Aufruf mit Löschen

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.
- `<deleteExistent>boolean</deleteExistent>`: Vor dem Speichern sollen alle Schnittstellen-Auftragsdaten gelöscht werden.
- Array von `<AuftragData>` mit diesen Feldern:
- `<fremdID>string</fremdID>`: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- `<belegDatum>string</belegDatum>`: Belegdatum, voraussichtliches Rechnungsdatum
- `<nettoFaellig>string</nettoFaellig>`: Fälligkeitsdatum
- `<steuerFaellig>string</steuerFaellig>`: Fälligkeit der Umsatzsteuer
- `<zahlZiel>string</zahlZiel>`: Zahlungsziel
- `<eigenBrutto>double</eigenBrutto>`: Bruttobetrag in Eigenwährung
- `<fremdBrutto>double</fremdBrutto>`: Bruttobetrag in Fremdwährung
- `<waehrung>string</waehrung>`: (Fremd-)Währung der Buchung
- `<kurs>double</kurs>`: Umrechnungskurs
- `<einheit>int</einheit>`: Einheit zum Kurs, falls nicht 1
- `<eigenSteuer>double</eigenSteuer>`: Steuerbetrag in Eigenwährung
- `<fremdSteuer>double</fremdSteuer>`: Steuerbetrag in Fremdwährung
- `<steuerProzent>double</steuerProzent>`: Umsatzsteuer-Prozentsatz
- `<herkunft>string</herkunft>`: Herkunfts-Kennung Ihrer Anwendung
- `<bemerkung>string</bemerkung>`: Buchungstext, Vorgangstext

- <linkInfo>string</linkInfo>: Pfad zum PDF mit dem Auftrag
- <konto>int</konto>: Personenkonto

Rückgabe

- AuftragSaveAllResult: Enum der möglichen Returncodes.

AuswertungskreisGet

Gibt die Daten eines Auswertungskreises zurück.

Die Auswertungskreise müssen in der Fibu gepflegt und mit Buchungskreiszuordnungen und Benutzerrechten versehen werden. Sie können gelesen werden, um Daten für einen Auswertungskreis abrufen zu können.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.
- <auswertungskreisID>int</auswertungskreisID>: Nummer des Auswertungskreises.

Rückgabe

auswertungskreisID: Nummer des Auswertungskreises.

name: Bezeichnung des Auswertungskreises.

AuswertungskreisGetAll

Gibt alle Auswertungskreise eine Mandanten zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.

Rückgabe

Array von <AuswertungskreisData> mit den unter AuswertungskreisGet beschriebenen Feldern.

BestellungGet

Stellt die Daten einer Bestellung zur Verfügung.

Zahlungsdaten einer Warenwirtschaft-Bestellung für die Liquiditätsplanung. Zur Übergabe von Bestellungsdaten an die Liquiditätsplanung der Chefinfo (falls nicht ein View in das ERP-System genutzt wird). Die Verwendung dieser Funktion setzt voraus, dass eine Tabelle „cif_li_bestellung“ vorhanden ist, vgl. Schnittstellendokumentation zur Liquiditätsplanung. Informationen zur Übergabe der einzelnen Felder siehe ebenfalls Schnittstellendokumentation zur Liquiditätsplanung.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.
- <bestellungID>int</bestellungID>: Interne Nummer des Bestellungs-Datensatzes.

Rückgabe

- fremdID: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- belegDatum: Belegdatum, voraussichtliches Rechnungsdatum
- nettoFaellig: Fälligkeitsdatum
- steuerFaellig: Fälligkeit der Umsatzsteuer
- zahlZiel: Zahlungsziel
- eigenBrutto: Bruttobetrag in Eigenwährung
- fremdBrutto: Bruttobetrag in Fremdwährung
- waehrung: (Fremd-)Währung der Buchung
- kurs: Umrechnungskurs
- einheit: Einheit zum Kurs, falls nicht 1
- eigenSteuer: Steuerbetrag in Eigenwährung
- fremdSteuer: Steuerbetrag in Fremdwährung
- steuerProzent: Umsatzsteuer-Prozentsatz
- herkunft: Herkunftskennung Ihrer Anwendung
- bemerkung: Buchungstext, Vorgangstext
- linkInfo: Pfad zum PDF mit dem Bestellung
- konto: Personenkonto

BestellungGetAll

Liefert alle Bestellungen aus cif_li_bestellung.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.

Rückgabe

Array von `<BestellungData>` wie unter `BestellungGet` beschrieben.

BestellungSaveAll

Speichert das Array von Bestellungen und löscht ggf. zuvor. Gedacht für eine paketweise Übergabe, beim 1. Aufruf mit Löschen

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.
- `<deleteExistent>boolean</deleteExistent>`: Vor dem Speichern sollen alle Schnittstellen-Bestellungsdaten gelöscht werden.
- Array von `<BestellungData>` mit diesen Feldern:
- `<fremdID>string</fremdID>`: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- `<belegDatum>string</belegDatum>`: Belegdatum, voraussichtliches Rechnungsdatum
- `<nettoFaellig>string</nettoFaellig>`: Fälligkeitsdatum
- `<steuerFaellig>string</steuerFaellig>`: Fälligkeit der Umsatzsteuer
- `<zahlZiel>string</zahlZiel>`: Zahlungsziel
- `<eigenBrutto>double</eigenBrutto>`: Bruttobetrag in Eigenwährung
- `<fremdBrutto>double</fremdBrutto>`: Bruttobetrag in Fremdwährung
- `<waehrung>string</waehrung>`: (Fremd-)Währung der Buchung
- `<kurs>double</kurs>`: Umrechnungskurs
- `<einheit>int</einheit>`: Einheit zum Kurs, falls nicht 1
- `<eigenSteuer>double</eigenSteuer>`: Steuerbetrag in Eigenwährung
- `<fremdSteuer>double</fremdSteuer>`: Steuerbetrag in Fremdwährung
- `<steuerProzent>double</steuerProzent>`: Umsatzsteuer-Prozentsatz
- `<herkunft>string</herkunft>`: Herkunftskennung Ihrer Anwendung
- `<bemerkung>string</bemerkung>`: Buchungstext, Vorgangstext
- `<linkInfo>string</linkInfo>`: Pfad zum PDF mit dem Bestellung
- `<konto>int</konto>`: Personenkonto

Rückgabe

- BestellungSaveAllResult: Enum der möglichen Returncodes.

BuchungskreisExistsNummer

Ist eine bestimmte Buchungskreisnummer schon im Mandant vergeben?

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer
- <nummer>int</nummer>: Die gewünschte Buchungskreisnummer.

Rückgabe

Ergebnis True oder False.

BuchungskreisGet

Gibt die Daten eines Buchungskreises zurück.

Die Buchungskreise, mit denen mehrere Rechnungslegungen organisiert werden, müssen in der Fibu gepflegt und mit Benutzerrechten versehen werden. Sie können gelesen werden, um Daten für einen Buchungskreis abrufen zu können.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer.
- <buchungskreisID>int</buchungskreisID>: Nummer des Buchungskreises.

Rückgabe

BuchungskreisID: Nummer des Buchungskreises.

name: Bezeichnung des Buchungskreises.

BuchungskreisGetAll

Gibt alle Buchungskreise eines Mandanten zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer.

Rückgabe

Array von <BuchungskreisData> mit den unter BuchungskreisGet beschriebenen Feldern.

DatenbankGetAll

Ab Version 5.1.7: gibt eine Auflistung der Datenbanken lt. sql.config zurück.

Aufrufparameter

Keine.

Rückgabe

Liste der in der sql.config im OFCKS.NET-Installationsverzeichnis genannten Datenbank-Namen. Der zuerst genannte Datenbank-Name bezieht sich auf die Datenbank, die aktuell mit dem Webdienst verbunden ist (s. [Application.config](#)).

DllVersion

Gibt die Version der OFCKS.NET Assembly zurück.

Aufrufparameter

Keine.

Rückgabe

Versionsnummer der DLL als Zahl.

FremdwaehrungExistsKuerzel

Ist ein bestimmtes Fremdwährungskürzel im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kuerzel>string</kuerzel>`: Das gewünschte Fremdwährungskürzel.

Rückgabe

Ergebnis True oder False.

FremdwaehrungGetAll

Gibt alle (Fremd)Währungen eines Mandanten zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.

Rückgabe

Array von `<FremdwaehrungData>`, das lediglich das Feld `<waehrungID>` mit dem Währungskürzel enthält.

FremdwaehrungGetKurs

Gibt den Umrechnungskurs einer Währung an einem Datum zurück.

Wenn die Methoden den Fehler „Zu diesem Index konnte kein Datensatz gefunden“ wirft, kann dies 2 Gründe haben: Entweder wurde eine nicht definierte Währung abgefragt, oder es gibt zu diesem Tag keinen Kurs. Ein ermittelter Kurs kann ein Tageskurs oder ein Monats-Durchschnittskurs sein.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.
- `<waehrungID>string</waehrungID>`: Kürzel der Währung.
- `<kursDatum>dateTime</kursDatum>`: der Tag, für den der Kurs abgefragt wird.

Rückgabe

`waehrungID`: Kürzel der Währung.

`gueltigDatum`: Datum des abgefragten Kurses.

kurs: Der Umrechnungskurs.

einheit: ggf. 10er Potenz, um große Kurszahlen zu vermeiden.

isMengennotiz: Handelt es sich bei der Kursangabe um Mengen- oder Preisnotiz?

isMonatsDurchschnitt: Handelt es sich um einen Monats-Durchschnittskurs?

GetStatus

Stellt die Verbindung zur Datenbank her und gibt den Erfolg dieser Aktion zurück.

Aufrufparameter

Keine.

Rückgabe

Enum von OK oder verschiedenen möglichen Fehlern.

GueltigesBuchDatum

Prüft, ob Buchungen zu einem Datum in dem Mandant möglich sind.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer.
- <bookDate>string</bookDate>: Datum, für das geprüft werden soll, ob es bebuchbar ist.

Rückgabe

<GueltigesBuchDatumResult>boolean</GueltigesBuchDatumResult>: Zeigt an, ob bebuchbar.

HelloWorld

Testfunktion zur Prüfung der ordnungsgemäßen Installation.

Aufrufparameter

Keine

Rückgabe

Begrüßungstext.

IsKoreEingerichtet

Ist in dem Mandant die Kore zu dem Datum aktiv?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.
- `<koreDate>string</koreDate>`: Datum, für das geprüft werden soll, ob die Kostenrechnung aktiv ist.

Rückgabe

`<IsKoreEingerichtetResult>boolean</IsKoreEingerichtetResult>`: Zeigt an, ob aktiv.

KontoBankGetAll

Liefert die Bankverbindungen eines Kontos.

Ein Personenkonto kann mehrere Bankverbindungen haben. Unter diesen gibt es die präferierte Bankverbindung. Diese ist vom Konto aus in der Eigenschaft Bank zugänglich. Ihre Banknummer kann im Feld Clearing ausgelesen werden.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer.
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.

Rückgabe

Array von `<BankData>` mit diesen Feldern:

- `bankKontoNr`: Bank-Kontonummer der Bankverbindung
- `bankName`
- `blz`: Bankleitzahl
- `iban`: International Bank Account Number.
- `inhaber`: Inhaber des Bankkontos
- `land`: Land der Bank. Nicht nötig für Inlandsbankverbindungen.

- name1: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- name2: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- ort: optional, nur für Auslandsbankverbindungen ohne SWIFT-Code
- plz: Postleitzahl der Bank. Nur für manche Auslandsbankverbindungen notwendig
- strasse: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- swiftCode: Swift-Code oder BIC der Bank, im Inland nicht mehr erforderlich.
- zahlungskennung: Zahlungskennungen(0=, 1=, bei CH)
- bankID: Eindeutige interne Banknummer

KontoBankGetSelected

Liefert bestimmte Bankverbindungen eines Kontos.

Ein Personenkonto kann mehrere Bankverbindungen haben. Die Methode KontoBankGetAll liefert sie vollständig, die Methode KontoBankGetSelected erlaubt Einschränkungen.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer.
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <ohneABI>boolean</ohneABI>: Wenn true, werden im Kontoauszugsimport ABI automatisch angelegte Bankverbindungen nicht berücksichtigt.
- <zeitFilter>enum</zeitFilter>: Ein Enum, ob nur die Neueste Bankverbindung nach Anlagedatum oder nach dem Datum der letzten Änderung geliefert werden soll. Der Wert Ungefiltert schränkt nicht ein.

Rückgabe

Array von <BankData> mit demselben Aufbau wie für die Methode KontoBankGetAll.

KontoBankGetNewSepaMandatName

Liefert die nächste Mandatsnummer (Mandatsname). Voraussetzung ist, dass die automatische Vergabe von SEPA-Mandatsnummern konfiguriert ist.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer.
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <bankID>int</bankID>: Interne Nummer der Bankverbindung.

Rückgabe

Neue Mandatsnummer / Mandatsname.

KontoBankSaveAll

Aktualisiert die Bankverbindungen eines Kontos.

Die Zuordnung geschieht entweder exakt anhand der BankID. Falls diese nicht vorhanden ist, wird anhand IBAN / BLZ / Kontonummer verglichen, ob die Bankverbindung schon vorhanden war.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer.
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.
- `<username>string</username>`: Anwendername im Vorsystem. Wird als Ändernder in der Bankverbindung eingetragen (optional, ab Version 3.0.5.1).

Array von `<BankData>` mit diesen Feldern:

- `<bankKontoNr>string</bankKontoNr>`: Bank-Kontonummer der Bankverbindung
- `<bankName>string</bankName>`: Name der Bank
- `<blz>string</blz>`: Bankleitzahl
- `<iban>string</iban>`: International Bank Account Number.
- `<inhaber>string</inhaber>`: Inhaber des Bankkontos
- `<land>string</land>`: Land der Bank. Nicht nötig für Inlandsbankverbindungen.
- `<name1>string</name1>`: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- `<name2>string</name2>`: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- `<ort>string</ort>`: optional, nur für Auslandsbankverbindungen ohne SWIFT-Code
- `<plz>string</plz>`: Postleitzahl der Bank. Nur für manche Auslandsbankverbindungen notwendig
- `<strasse>string</strasse>`: optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- `<swiftCode>string</swiftCode>`: Swift-Code oder BIC der Bank, im Inland nicht mehr erforderlich.
- `<zahlungsKennung>int</zahlungsKennung>`: Zahlungskennungen (0=, 1=, bei CH)
- `<bankID>int</bankID>`: Eindeutige interne Banknummer

Rückgabe

Enum von OK oder möglichen Fehlerursachen.

KontoBankSaveThis

Aktualisiert eine Bankverbindung eines Kontos.

Falls per BankID, oder IBAN oder BLZ / Kontonummer eine vorhandene Bankverbindung eindeutig identifiziert wird, wird diese aktualisiert. Falls nicht, wird die Bankverbindung neu angelegt. Es werden keine Bankverbindungen gelöscht.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer.
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.
- `<username>string</username>`: Anwendername im VORSYSTEM. Wird als Ändernder in der Bankverbindung eingetragen (optional, ab Version 3.0.5.1).

Nur 1 Block `<date>` mit dem Aufbau wie `<BankData>` der Methode `KontoBankSaveAll` mit denselben Feldern wie dort beschrieben.

Rückgabe

Enum von OK oder möglichen Fehlerursachen.

KontoBankSaveSepsaMandat

Speichert ein SEPA-Mandat - Anlegen oder Aktualisieren.

Die SEPA-Mandate gehören zu einer konkreten Bankverbindung. Eine Mandatsnummer darf aktiv nur zu einer Bankverbindung vorhanden sein. Es sind nur bestimmte Status-Übergänge zulässig.

In der Dokumentation zu OFCKS.NET sind die Modelle der Zusammenarbeit zwischen ERP und Fibu im Hinblick auf SEPA-Mandate beschrieben.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.
- `<bankID>int</bankID>`: Interne Banknummer
- `<sepaMandat>string</sepaMandat>`: SEPA-Mandatsnummer / Mandatsname.
- `<unterschDatum>dateTime</unterschDatum>`: Unterschriftsdatum, kann NULL sein. Wenn geliefert, ist das Mandat im Status "Aktiv", ohne Unterschrift im Status "Ausgestellt".
- `<mandatVorgang>Unbekannt or Einzel or Dauer or Widerruf or Inaktiv or FirmenEinzel or FirmenDauer</mandatVorgang>`: Enum, das die Anlage als Einzel- oder Dauermandat Basis- oder Firmenlastschrift kennzeichnet. Ebenso sind Widerruf und Inaktiv-Setzen möglich.

- `<username>string</username>`: Anwendername im Vorsystem. Wird als Ändernder im SEPA-Mandat eingetragen (optional, ab Version 3.0.5.1).

Rückgabe

Enum von OK oder möglichen Fehlerursachen.

KontoBankSepaMandatGetAll

Gibt die SEPA-Mandate einer Bankverbindung zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.
- `<bankID>int</bankID>`: Interne Banknummer
- `<nurAktivDauer>boolean</nurAktivDauer>`: Nur aktive Dauermandate sollen geholt werden.

Rückgabe

Array von `<SepaMandatData>` mit diesen Feldern:

- `mandatName`: Externe Mandatsnummer
- `bemerkung`: Hinweis zum Mandat
- `uciComplete`: Vollständige Gläubiger-ID der Firma, die zum Zeitpunkt der Ausstellung des Mandats gültig war.
- `art`: Art des Mandats (Enum): Einzel- oder Dauermandat als Basis- oder Firmenlastschrift.
- `status`: Zahl, die den Status des Mandats wiedergibt.
- `unterschrift`: Unterschriftsdatum. Kann NULL sein.
- `benutzt`: Letztes Nutzungsdatum. Kann NULL sein.
- `bankID`: Nummer der Bankverbindung, zu der das SEPA-Mandat gehört.
- `kontoNr`: Personenkonto, zu dem das SEPA-Mandat gehört.
- `version`: Neueste Versionsnummer des Mandats.

KontoDelete

Löscht ein Personenkonto, falls es gelöscht werden kann. Wenn ein Konto bebucht ist, kann es nicht mehr gelöscht werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.

Rückgabe

Enum von OK oder möglichen Fehlerursachen.

KontoExistsKontoKuerzel

Ist ein bestimmtes Kontokürzel im Mandant schon vergeben?

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kuerzel>string</kuerzel>: Die gewünschte Kontokurzbezeichnung.

Rückgabe

Ergebnis True oder False.

KontoExistsKontoNummer

Ist eine bestimmte Kontonummer im Mandant schon vergeben?

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoID>int</kontoID>: Die gewünschte Kontonummer.

Rückgabe

Ergebnis True oder False.

KontoGet

Gibt die Daten eines Kontos zurück.

Konten sind grundsätzlich alle Kontoarten: die Personenkonten und die Sachkonten. Eine Kontonummer kann nur als eine Kontoart vorhanden sein. Die Sachkonten sind in der Fibu zu pflegen, stehen der Voranwendung jedoch zur Abfrage zur Verfügung.

Das Konto kann ein Sachkonto, ein Personenkonto (Debitor oder Kreditor) oder eine Einmaladresse (eines Debitoren-Einmalkontos oder eines Kreditoren-Einmalkontos) sein. Einmaladressen bekommen vom System eine fortlaufende negative "Kontonummer" zugewiesen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kontoNr>int</kontoNr>`: Fibu-Kontonummer des gewünschten Kontos.

Rückgabe

- **AdressFlags**: Steuerflags für die Adressbildung für Mahnungen und andere Briefreports (Enum). Der Enum-Wert `EinstellungKontoAdressFlags` kann im Konto selbst nicht verwendet werden.
- **aHWSaldo**: Saldo des Kontos in alter Hauswährung
- **ansprechpartner**: Name des Ansprechpartners
- **bankRefID**: Nummer der präferierten Bankverbindung des Kontos
- **clearingFlag**: Zahlungsart (Enum). Mögliche Zahlungsarten sind je nach Kontoart:
 - Debitoren:
 - `cfDEBI_Abbuchung`
 - `cfDEBI_Einzug`
 - `cfDEBI_Rueckzahlung`
 - `cfDEBI_SEPALastschrift`
 - Kreditoren:
 - `cfKRED_ESR` (Schweiz)
 - `cfKRED_Gutschrift` (Überweisung im Sinne einer Bankgutschrift)
 - `cfKRED_Scheck`
 - beide:
 - `cfKein` (gleichbedeutend mit Rechnung)
 - `cfUseDefault` ist im Konto selbst nicht zulässig sondern in Buchungen ggf. zu verwenden.
- **datev**: Datevautomatik: bei Sachkonten: `true` = Automatikkonto, `false` = sonst, bei Personenkonten: `true` = exportiert, `false` = nicht exportiert
- **eMail**
- **eurSaldo**: Saldo des Kontos in Euro

- fax
- forderungskonto: (Debitoren) falls Leer gilt das Standardforderungskonto welches beim Mandanten abgefragt werden kann.
- homepage
- kontoArt: Art des Kontos. Möglichkeiten:
 - kaSachkonto
 - kaDebitorkonto
 - kaKreditorkonto
 - kaDebitoradresse
 - kaKreditoradresse
- kontoKlasse: Nur Sachkonten (Enum). Aktiv-, Passiv- oder GuV-Konto.
- kontoKuerzel: Kurzbezeichnung des Kontos. Nur Grossbuchstaben zugelassen. Übergebene Strings werden automatisch umgewandelt.
- kontoName
- kontoNummer: Eindeutige Kontonummer. Einmaladressen haben eine negative, vom System vergebene Kontonummer.
- kostenart: Nummer der Kostenart falls Kostenkonto
- kundenID: (Debitoren) Kundennummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.
- kundenRefID: (Debitoren) interne Referenznummer bei Kunden. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- landKennung: Landes Kennung bei Personenkonten
- letztesBuchDatum: Das Datum der letzten Buchung auf das Konto ist abrufbar.
- lieferantenID: (Kreditoren) Lieferantenummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.
- lieferantenKundenNr: Die Kundennummer welche der Lieferant dem Mandanten vergeben hat. Also die eigene Kunden Nummer, welche man beim Lieferanten hat.
- lieferantenRefID: (Kreditoren) interne Referenznummer. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- mahnbar: (Debitoren) Ist Kunde mahnbar oder nicht?
- maxMahnstufe: (Debitoren) maximale Mahnstufe, die die OPs des Kunden erreicht haben
- name2: Felder für zusätzliche Namen
- name3
- ort
- ort2: Postfach-Ort
- plz: Postleitzahl
- plz2: Postfach-Postleitzahl
- saldo: Kontosaldo. In Mandanten, die vor der Euro-Umstellung begannen, ist das der Saldo in der früheren Währung.
- sprache: (Debitoren) Mahnsprache
- steuerArtID: Steuerart die dem Konto zugeordnet ist

- strasse
- strasse2: Feld für das Postfach.
- telefonNr
- titel
- uStIDNr: Umsatzsteuer ID Nummer
- uStVAKz: Umsatzsteuer Voranmeldungs-Kennziffer. Nur Sachkonten
- verbindlichkeitenKonto: Falls leer gilt das Standardverbindlichkeitenkonto welches beim Mandanten hinterlegt ist.
- waehrung: Kontowährung
- wwsKennung: Kennung, ob das Konto ein Aufwands- oder Erlöskonto der Warenwirtschaft ist.
- zahlzielID: Nummer des zum Konto gehörenden Zahlziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- zusatzText: Textfeld für Zusatztext bzw. alternativer Kontoname bei Sachkonten.
- text1 / text2 / text3: Frei verwendbare Text-Zusatzfelder für Personenkonten.
- option1 / option2 / option3: Frei verwendbare Ja/Nein-Zusatzfelder für Personenkonten.
- wert1 / wert2 / wert3: Frei verwendbare Wert-Zusatzfelder für Personenkonten.
- zahlzielName. Name des zum Konto gehörenden Zahlungsziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- einmal: Personenkonto ist ein Einmalkonto.
- steuernummer: Steuernummer des Kunden / Lieferanten.
- eMailAnsprechpartner: Der Email-Ansprechpartner
- eMail2
- zessionsvermerk
- zahlungsAvis: Enum Avise gem. Zahlungslauf (Standard), kein Avis, immer Avis
- verbandID: -1 = Verbandsmutter, positive Zahl = Verbandsmitglied bei diesem Konto. Nur für Debitoren und Kreditoren.
- kontoStatus: Macht den Inaktiv-Status eines Kontos sichtbar. In inaktiven Konten steht ein "I" als Status. Die Eigenschaft kann nicht gesetzt werden, Konten müssen in der Fibu deaktiviert werden.
- PruefstatusUstid (ab Version 3.0.3.0): Ergebnisstatus der Prüfung der Umsatzsteuer-Identnummer (Enum).
- PruefungUstid (ab Version 3.0.3.0): Datum der Prüfung der Umsatzsteuer-Identnummer oder NULL.
- anmerkung (ab Version 5.1.0.1): Das lange Anmerkungsfeld in den Personenkonten.
- Kostenschablone (ab Version 5.1.2.1, nur Lesen): Nummer der Kostenschablone falls Kostenkonto

KontoGetAll

Gibt die Daten einer KontoCollection mit Filtereinschränkungen zurück. Diese Methode listet nur aktive Konten auf.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kontoVon>int</kontoVon>`: Beginn-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- `<kontoBis>int</kontoBis>`: Ende-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- `<kontoArt>Enum</kontoArt>`: nur Konten dieser Kontoart sind gewünscht, sonst kaAll.
- `<sortOrder>Enum</sortOrder>`: gewünschte Sortierung. Kann NULL sein.
- `<einmalKonto>int</einmalKonto>`: Nummer des gewünschten Einmalkontos. Kann NULL sein.

Rückgabe

Array von `<KontoData>` mit den Feldern wie unter KontoGet beschrieben.

KontoGetAllKennung

Gibt die Daten einer KontoCollection mit Filtereinschränkungen zurück. Mit dieser Methode kann man auf Konten mit einer bestimmten Kennung einschränken.

Die Aufrufparameter sind mit einer Ausnahme derzeit identisch zur Methode KontoGetAll. Anstelle des Parameters `kontoArt` steht der Parameter `kontoKennung` zur Verfügung.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kontoVon>int</kontoVon>`: Beginn-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- `<kontoBis>int</kontoBis>`: Ende-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- `<kontoKennung>Enum</kontoKennung>`: nur Konten mit dieser Kennung sind gewünscht.
- `<sortOrder>Enum</sortOrder>`: gewünschte Sortierung. Kann NULL sein.
- `<einmalKonto>int</einmalKonto>`: Nummer des gewünschten Einmalkontos. Kann NULL sein.

Rückgabe

Array von <KontoData> mit den Feldern wie unter KontoGet beschrieben.

KontoGetAllStatus

Gibt die Daten einer KontoCollection mit Filtereinschränkungen zurück. Diese Methode listet auch inaktive Konten auf.

Die Aufrufparameter sind derzeit identisch zur Methode KontoGetAll. Falls künftig mehr Kontostatūs (Mehrzahl von Kontostatus) möglich sind, kann sich die Signatur ändern.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoVon>int</kontoVon>: Beginn-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- <kontoBis>int</kontoBis>: Ende-Kontonummer eines Abfrage-Bereichs. Kann NULL sein.
- <kontoArt>Enum</kontoArt>: nur Konten dieser Kontoart sind gewünscht, sonst kaAll.
- <sortOrder>Enum</sortOrder>: gewünschte Sortierung. Kann NULL sein.
- <einmalKonto>int</einmalKonto>: Nummer des gewünschten Einmalkontos. Kann NULL sein.

Rückgabe

Array von <KontoData> mit den Feldern wie unter KontoGet beschrieben.

KontoGetBelegzeilen

Gibt alle ab dem angegebenen Datum auf das Konto gebuchten Belege zurück. Ab Version 5.1.2.2

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <belegDatumVon>dateTime</belegDatumVon>: Ab welchem Datum sollen die Belegzeilen zurückgegeben werden?

Rückgabe

Array von <BelegZeileData> mit diesen Feldern:

- <auftragID>int</auftragID>: AuftragID der Belegzeile (aus dem Belegkopf)
- <belegDatum>dateTime</belegDatum>: Belegdatum
- <belegNr>string</belegNr>: Belegnummer
- <belegRefID>int</belegRefID>: (Interne fortlaufende) Buchungsnummer
- <Betrag>double</betrag>: Betrag in Belegwährung
- <buchungsart>Enum</buchungsart>: Enum der Belegart, Standard ist OFCKBuchungsart.baLfdBuchung
- <buchungsDatum>dateTime< buchungsDatum>: Buchungsdatum
- <buchungskreisID>int</buchungskreisID>: Buchungskreisnummer
- <buchungsText>string</buchungsText>: Buchungstext
- <eigenBetrag>double</eigenBetrag>: Betrag in Mandantenwährung.
- <eigenSteuerBetrag>double</eigenSteuerBetrag>: Steuerbetrag in Mandantenwährung.
- <gegenkonto>int</gegenkonto>: Gegenkonto (informativ)
- <sollHabenKennzeichen>Enum</sollHabenKennzeichen>: Soll- oder Habenbuchungszeile.
- <geschaeftsJahr>int</geschaeftsJahr>: Geschäftsjahr in das gebucht wurde
- <kontoArt>Enum</kontoArt>: Art des Kontos in dieser Zeile
- <konto>int</konto>: gebuchtes Konto
- <periode>Enum</periode>: Absolute Periode in die gebucht wurde (oder 13. Periode).
- <relPeriode>Enum</relPeriode>: Relative Periode in die gebucht wurde (oder 13. Periode)
- <steuerArtID>int</steuerArtID>: Steuerart
- <steuerSatz>double</steuerSatz>: Steuersatz
- <waehrung>string</waehrung>: Währung
- <zeileNr>int</zeileNr>: Belegzeilennummer in der Buchung

KontoGetBelegzeilenAbBuchNr

Gibt alle ab der angegebenen internen Belegnummer auf das Konto gebuchten Belege zurück. Ab Version 5.1.2.2.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <buchNrVon>int</ buchNrVon >: Ab welcher internen Belegnummer sollen die Belegzeilen zurückgegeben werden?

Rückgabe

Array von <BelegZeileData> mit den Feldern wie unter KontoGetBelegzeilen beschrieben.

KontoGetKennung

Gibt die Daten einer Kontocollection mit Einschränkung nach Kontokennung zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoKennung>Enum</kontoKennung>: Filterung auf die gewünschte Kontenkennung.
- <sortOrder>Enum</sortOrder>: gewünschte Sortierung. Kann NULL sein.

Rückgabe

Array von <KontoData> mit den Feldern wie unter KontoGet beschrieben.

KontoGetListe

Gibt die Daten einer KontoListeCollection mit Filtereinschränkungen zurück. Ab Version 3.0.6.0.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoArt>Enum</kontoArt>: nur Konten dieser Kontoart sind gewünscht, sonst kaAll.
- <suchName>string</suchName>: Suchbegriff für die Suche im Kontoname. Der Suchbegriff wird für eine Like-Suche verwendet und kann Wildcards enthalten.
- <suchKuerzel>string</suchKuerzel>: Suchbegriff für die Suche im Kontenkürzel. Der Suchbegriff wird für eine Like-Suche verwendet und kann Wildcards enthalten.
- <anzahl>int</anzahl>: Begrenzung des Ergebnisses auf eine gewünschte Anzahl. Kann NULL sein.
- <sortOrder>Enum</sortOrder>: gewünschte Sortierung. Kann NULL sein.

Rückgabe

Array von <KontoListe> mit den wichtigsten Feldern eines Kontos:

<kontoNummer>int</kontoNummer>: Kontonummer.

<kontoKuerzel>string</kontoKuerzel>: Kontokürzel.

<kontoName>string</kontoName>: Kontenname, Bezeichnung.

KontoGetNextFreeNo

Gibt die nächste freie Kontonummer in einem Nummernbereich zurück

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoArt>Enum</kontoArt>: Filterung auf die gewünschte Kontoart.
- <von>int</von>: Nummernbereichseinschränkung Beginn.
- <bis>int</bis>: Nummernbereichseinschränkung Ende.

Rückgabe

Kontonummer

KontoGetPeriodenSaldo

Gibt den Saldo eines Kontos für eine Periode aus.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Konto, zu dem Auskunft gewünscht wird
- <buchungsKreis>int</buchungsKreis>: Buchungskreis, aus dem der Saldo gewünscht wird. Wird 0 mitgegeben, wird für Standardbuchungskreis 1 abgerufen.
- <kalJahr>int</kalJahr>: (Kalender-)Jahr, für das abgerufen wird.
- <periode>Enum</periode>: (Absolute) Periode, für die der Saldo gewünscht ist. Es stehen die Perioden 1 bis 12 und 13 zur Verfügung.

Rückgabe

- <SaldoSoll>double</SaldoSoll>: Saldo der Periode auf der Sollseite.
- <SaldoHaben>double</SaldoHaben>: Saldo der Periode auf der Habenseite.
- <SaldoGesamt>double</SaldoGesamt>: Gesamtsaldo (Soll – Haben) der Periode.

KontoGetSaldoMitStapel

Gibt den aktuellen Saldo eines Kontos inklusive Stapelbuchungen (bis zur aktuellen Kalender-Periode) aus. Dieser Saldo kann für eine Kreditlimitprüfung einen genaueren Wert liefern als das Feld Saldo in KontoGet, wenn Schnittstellen-Buchungen noch im Buchungsstapel stehen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kontoNr>int</kontoNr>`: Konto, zu dem Auskunft gewünscht wird

Rückgabe

- `<KontoGetSaldoMitStapelResult>double</KontoGetSaldoMitStapelResult>`: Aktueller Kontosaldo inkl. Stapelbuchungen.

KontoGetWaWiKennung

Gibt die Daten einer Kontocollection mit Einschränkung nach Wawikennung zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<wawiKennung>Enum</wawiKennung>`: Filterung auf die gewünschte Kontenkennung für die WaWi: Aufwands- oder Erlöskonto.
- `<sortOrder>Enum</sortOrder>`: gewünschte Sortierung.

Rückgabe

Array von `<KontoData>` mit den Feldern wie unter KontoGet beschrieben.

KontoGetZahlverhalten

Gibt die wichtigsten Zahlungsverhalten-Kennziffern zu einem Personenkonto (i.a. Debitor) aus.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

- `<kontoNr>int</kontoNr>`: Konto, zu dem Auskunft gewünscht wird
- `<datumsArt>Enum</datumsArt>` Die Datumsabgrenzung kann nach Zahlungsdatum oder Valutadatum geschehen. Im ersteren Fall sind nur OPs in der Auswertung, die mindestens teilbezahlt sind.
- `<datumVon>dateTime</datumVon>`: Ab welchem Datum gemäß Datumsart sollen OPs in die Auswertung einbezogen werden? Falls leer, dann mit unbegrenztem Alter.
- `<datumBis>dateTime</datumBis>`: Bis zu welchem Datum gemäß Datumsart sollen OPs in die Auswertung einbezogen werden? Falls leer, dann auch neueste OPs.

Rückgabe

- `<schnellsteZahlung>int</schnellsteZahlung>`: Ausgabe der schnellsten Zahlung in Tagen.
- `<langsamsteZahlung>int</langsamsteZahlung>`: Ausgabe der langsamsten Zahlung in Tagen.
- `<durchschnZahlung>int</durchschnZahlung>`: Ausgabe der durchschnittlichen Zahlung in Tagen.
- `<durchschnVerzug>int</durchschnVerzug>`: Ausgabe der durchschnittlichen Zahlung ab Fälligkeit in Tagen.
- `<anzahlPosten>int</anzahlPosten>`: Ausgabe der Anzahl Posten, die in die Auswertung eingegangen sind.

KontoSave

Speichert ein Konto - Neuanlegen oder Update.

Konten sind grundsätzlich alle Kontoarten: die Personenkonten und die Sachkonten. Eine Kontonummer kann nur als eine Kontoart vorhanden sein. Die Sachkonten sind in der Fibu zu pflegen, sie können mit der Schnittstelle nicht gespeichert werden.

Das Konto kann ein Personenkonto (Debitor oder Kreditor) oder eine Einmaladresse (eines Debitoren-Einmalkontos oder eines Kreditoren-Einmalkontos) sein. Einmaladressen bekommen vom System eine fortlaufende negative "Kontonummer" zugewiesen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<isKontoNew>boolean</isKontoNew>`: Konto Neuanlage oder Änderung.
- `<username>string</username>`: Anwendername im VORSYSTEM. Wird als Ändernder im Konto eingetragen (optional, ab Version 3.0.5.1).

Eine Struktur `<data>` mit diesen Feldern:

- `<AdressFlags>int</AdressFlags>`: Steuerflags für die Adressbildung für Mahnungen und andere Briefreports (Enum). Der Enum-Wert EinstellungKontoAdressflags kann im Konto selbst nicht verwendet werden.
- `<aHWSaldo>double</aHWSaldo>`: Saldo des Kontos in alter Hauswährung
- `<ansprechpartner>string</ansprechpartner>`: Name des Ansprechpartners

- <bankRefID>int</bankRefID>: Nummer der präferierten Bankverbindung des Kontos
- <clearingFlag>Enum</clearingFlag>: Zahlungsart (Enum). Mögliche Zahlungsarten sind je nach Kontoart:
 - Debitoren:
 - cfDEBI_Abbuchung
 - cfDEBI_Einzug
 - cfDEBI_Rueckzahlung
 - cfDEBI_SEPALastschrift
 - Kreditoren:
 - cfKRED_ESR (Schweiz)
 - cfKRED_Gutschrift (Überweisung im Sinne einer Bankgutschrift)
 - cfKRED_Scheck
 - beide:
 - cfKein (gleichbedeutend mit Rechnung)
 - cfUseDefault ist im Konto selbst nicht zulässig sondern in Buchungen ggf. zu verwenden.
- <datev>boolean</datev>: Datevautomatik: bei Sachkonten: true = Automatikkonto, false = sonst, bei Personenkonten: true = exportiert, false = nicht exportiert
- <eMail>string</eMail>
- <eurSaldo>double</eurSaldo>: Saldo des Kontos in Euro
- <fax>string</fax>
- <forderungskonto>int</forderungskonto>: (Debitoren) falls Leer gilt das Standardforderungskonto welches beim Mandanten abgefragt werden kann.
- <homepage>string</homepage>
- <kontoArt>Enum</kontoArt>: Art des Kontos. Möglichkeiten:
 - kaSachkonto
 - kaDebitorkonto
 - kaKreditorkonto
 - kaDebitoradresse
 - kaKreditoradresse
- <kontoKlasse>Enum<kontoKlasse>: Nur Sachkonten (Enum) zum Auslesen, kann nicht gesetzt werden.
- <kontoKuerzel>string</kontoKuerzel>: Kurzbezeichnung des Kontos. Nur Grossbuchstaben zugelassen. Übergebene Strings werden automatisch umgewandelt.
- <kontoName>string</kontoName>
- <kontoNummer>int</kontoNummer>: Eindeutige Kontonummer. Einmaladressen haben eine negative, vom System vergebene Kontonummer.
- <kostenart>int</kostenart>: Nummer der Kostenart falls Kostenkonto
- <kundenID>int</kundenID>: (Debitoren) Kundennummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.

- <kundenRefID>int</kundenRefID>: (Debitoren) interne Referenznummer bei Kunden. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- <landKennung>string</landKennung>: Landes Kennung bei Personenkonten
- <letztesBuchDatum>string</letztesBuchDatum>: Das Datum der letzten Buchung auf das Konto ist abrufbar.
- <lieferantenID>int</lieferantenID>: (Kreditoren) Lieferantenummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.
- <lieferantenKundenNr>string</lieferantenKundenNr>: Die Kundennummer welche der Lieferant dem Mandanten vergeben hat. Also die eigene Kunden Nummer, welche man beim Lieferanten hat.
- <lieferantenRefID>int</lieferantenRefID>: (Kreditoren) interne Referenznummer. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- <mahnbar>boolean</mahnbar>: (Debitoren) Ist Kunde mahnbar oder nicht?
- <maxMahnstufe>int</maxMahnstufe>: (Debitoren) maximale Mahnstufe, die die OPs des Kunden erreicht haben
- <name2>string</name2>: Felder für zusätzliche Namen
- <name3>string</name3>
- <ort>string</ort>
- <ort2>string</ort2>: Postfach-Ort
- <plz>string</plz>: Postleitzahl
- <plz2>string</plz2>: Postfach-Postleitzahl
- <saldo>double</saldo>: Kontosaldo. In Mandanten, die vor der Euro-Umstellung begannen, ist das der Saldo in der früheren Währung.
- <spracheID>string</spracheID>: (Debitoren) Mahnsprache
- <steuerArtID>int</steuerArtID>: Steuerart die dem Konto zugeordnet ist
- <strasse>string</strasse>
- <strasse2>string</strasse2>: Feld für das Postfach.
- <telefonNr>string</telefonNr>
- <titel>string</titel>
- <uStIDNr>string</uStIDNr>: Umsatzsteuer ID Nummer
- <uStVAKz>int</uStVAKz>: Umsatzsteuer Voranmeldungs-Kennziffer. Nur Sachkonten
- <verbindlichkeitenKonto>: Falls leer gilt das Standardverbindlichkeitenkonto welches beim Mandanten hinterlegt ist.
- <waehrung>string</waehrung>: Kontowährung
- <wwsKennung>int</wwsKennung>: Kennung, ob das Konto ein Aufwands- oder Erlöskonto der Warenwirtschaft ist.
- <zahlzielID>int</zahlzielID>: Nummer des zum Konto gehörenden Zahlziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- <zusatzText>string</zusatzText>: Textfeld für Zusatztext bzw. alternativer Kontoname bei Sachkonten.

- `<text1>string</text1>` / `text2` / `text3`: Frei verwendbare Text-Zusatzfelder für Personenkonten.
- `<option1>boolean</option1>` / `option2` / `option3`: Frei verwendbare Ja/Nein-Zusatzfelder für Personenkonten.
- `<wert1>double</wert1>` / `wert2` / `wert3`: Frei verwendbare Wert-Zusatzfelder für Personenkonten.
- `<zahlzielName>string</zahlzielName>`: Name des zum Konto gehörenden Zahlungsziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- `<einmal>boolean</einmal>`: Personenkonto ist ein Einmalkonto.
- `<steuernummer>string</steuernummer>`: Steuernummer des Kunden / Lieferanten.
- `<eMailAnsprechpartner>string</eMailAnsprechpartner>`: Der Email-Ansprechpartner
- `<eMail2>string</eMail2>`
- `<zessionsvermerk>int</zessionsvermerk>`
- `<zahlungsAvis>Enum</zahlungsAvis>`: Enum Avise gem. Zahlungslauf (Standard), kein Avis, immer Avis
- `<verbandID>int</verbandID>`: -1 = Verbandsmutter, positive Zahl = Verbandsmitglied bei diesem Konto. Nur für Debitoren und Kreditoren.
- `<pruefstatusUstid>Enum</pruefstatusUstid` (ab Version 3.0.3.0): Falls die Prüfung der Umsatzsteuer-Identnummern im Vorsystem vorgenommen wird, kann hier der Geprüft-Status gesetzt werden (Enum). Zum Zurücksetzen auf Ungeprüft diesen Status setzen und evtl. das Prüfdatum mit NULL übergeben, um dieses zu löschen.
- `<pruefungUstid>dateTime</pruefungUstid>` (ab Version 3.0.3.0): Datum der Prüfung der Umsatzsteuer-Identnummer. Wird eingetragen, falls auch ein Prüfstatus übergeben wird. Die Übergabe von NULL löscht das Datum.
- `<anmerkung>string</anmerkung>` (ab Version 5.1.0.1): Das lange Anmerkungsfield in den Personenkonten.

KontoSetStatus

Setzt ein Personenkonto inaktiv oder auch wieder aktiv.

Darf auf Sachkonten nicht angewendet werden, diese werden in der Finanzbuchhaltung gepflegt.

Ein inaktives Personenkonto darf über Schnittstellen oder im normalen Buchungsdialog nicht mehr bebucht werden. In Opos ist es jedoch noch möglich, ggf. eintreffende Zahlungen zu buchen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kontoNr>int</kontoNr>`: Eindeutige Kontonummer.
- `<status>enum</status>`: Der neue Status. Invalid ist hier unzulässig.
- `<username>string</username>`: Anwendername im Vorsystem. Wird als Ändernder im Konto eingetragen (optional, ab Version 3.0.5.1).

KostenArtExistsKontoKuerzel

Ist ein bestimmtes Kostenartkürzel im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kuerzel>string</kuerzel>`: Das gewünschte Kostenartkürzel.

Rückgabe

Ergebnis True oder False.

KostenArtExistsKontoNummer

Ist eine bestimmte Kostenartnummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<nummer>int</nummer>`: Die gewünschte Kostenartnummer.

Rückgabe

Ergebnis True oder False.

KostenArtGet

Gibt die Daten einer Kostenart zurück.

Kostenarten können Primärkostenarten sein (verbunden mit Fibu-Konten), Sekundärkostenarten (nur Kore), Leistungsarten (nur Kore) oder Umlagekostenarten (nur Kore für die Umlage).

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kostenArtNr>int</kostenArtNr>`: Nummer der gewünschten Kostenart.

Rückgabe

- einheit: Bezugseinheit. Sie wird auch dann gespeichert, wenn sie in den Kore-Einstellungen nicht angelegt ist.
- erloesart: Handelt es sich um eine Erlös-Kostenart ? Nur für Primärkostenarten.
- fixkostenart: Handelt es sich um eine Fixkosten-Kostenart ?
- art: Art der Kostenart (Enum).
- kuerzel: Kürzel für die Kostenart.
- name: Name der Kostenart.
- kostenArtID: Kostenarten-Nummer.

KostenArtGetOE

Dieselbe Methode wie KostenArtGet, jedoch werden statt Enums die zugehörigen Integer-Zahlen zurückgegeben.

KostenArtGetAll

Gibt alle Kostenarten eines Mandanten zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Rückgabe

Array von <KostenArtData> mit den Feldern wie unter KostenArtGet beschrieben.

KostenArtSave

Speichert eine Kostenart – Anlegen oder Ändern.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <isNew>boolean</isNew>: Neuanlage oder Ändern?

Eine Struktur <data> mit diesen Feldern:

- `<einheit>string</einheit>`: Bezugseinheit. Sie wird auch dann gespeichert, wenn sie in den Kore-Einstellungen nicht angelegt ist.
- `<erloesart>boolean</erloesart>`: Handelt es sich um eine Erlös-Kostenart ? Nur für Primärkostenarten.
- `<fixkostenart>boolean</fixkostenart>`: Handelt es sich um eine Fixkosten-Kostenart ?
- `<art>enum</art>`: Art der Kostenart (Enum).
- `<kuerzel>string</kuerzel>`: Kürzel für die Kostenart.
- `<name>string</name>`: Name der Kostenart.
- `<kostenArtID>int</kostenArtID>`: Kostenarten-Nummer.

Rückgabe

Enum der möglichen Returncodes.

KostenBuchungSave

Speichert eine reine Kostenbuchung in den kr_stapel.

In einem Kostenrechnungs-Stapel können Sekundärkostenbuchungen übergeben werden.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<stapelID>int</stapelID>`: Stapelnummer

Ein Array von `<kostenBuchung>` für die Buchungen je Kostenart mit diesen Feldern:

- `<belegDatum>dateTime</belegDatum>`: Belegdatum
- `<belegNr>string</belegNr>`: Belegnummer
- `<einheit>string</einheit>`: Mengeneinheit – sofern mit Mengeneinheiten gebucht wird.
- `<geschaeftsJahr>int</geschaeftsJahr>`: Geschäftsjahr, bei abweichendem Geschäftsjahr das Beginn-Kalenderjahr des Geschäftsjahres.
- `<kostenArtID>int</kostenArtID>` Nummer der Kostenart
- `<periode>Enum</periode>` Enum der Buchungsperiode

Ein Array von `<KostenZeileData>`. Jede Buchung auf eine Kostenart muss auf mindestens eine, kann aber auf mehrere Kostenstellen und ggf. Kostenträger aufgeteilt werden.

- `<nettoEigenBetrag>double</nettoEigenBetrag>`: Buchungsbetrag in die Kostenrechnung, in Mandantenwährung. In die Kostenrechnung wird netto ohne Umsatzsteuer gebucht.
- `<kostenStelle>string</kostenStelle>`: Kostenstelle
- `<kostenTraeger1>string</kostenTraeger1>`: Kostenträger – falls mit Kostenträger gearbeitet wird.
- `<kostenTraeger2>` bis `<kostenTraeger5>` wie `<kostenTraeger1>`. In der Kostenrechnung können bis zu 5 Kostenträger-Kreise definiert sein.

- `<fixVariabel>Enum</fixVariabel>`: Die Angabe, ob es sich um Fixkosten oder variable Kosten handelt – falls die Einstellung diese Unterscheidung vorsieht.
- `<menge>double</menge>`: Mengenangabe – falls mit Mengen gebucht wird.
- `<buchungsText>string</buchungsText>`: Kostenrechnungs-Buchungstext.

Rückgabe

Enum der möglichen Returncodes.

KostenBuchungSaveAndBook

Speichert eine reine Kostenbuchung in den `kr_stapel`, prüft und bucht ihn (sofern alle Schritte erfolgreich sind).

In einem Kostenrechnungs-Stapel können Sekundärkostenbuchungen übergeben werden.

Die Hinweise zur Methode `StapelBuchungSaveAndBook` gelten analog.

Aufrufparameter

Die Aufrufparameter sind dieselben wie für die Methode `KostenBuchungSave`.

Rückgabe

- `<ergebnis>Enum</ergebnis>`: Enum des Erfolgs oder der möglichen Fehlerschritte.
- `<fehlermeldung>string</fehlermeldung>`: Die Fehlermeldung im Fehlerfall.
- `<protokoll>string</protokoll>`: Das Protokoll im Erfolgsfall.
- `<buchungsnummer>int</buchungsnummer>`: Gibt im Erfolgsfall die Buchungsnummer (`i_beleg_refid`) der verbuchten Buchung zurück, sonst 0.

KostenDimensionGetAll

Gibt die maximal 5 Kostendimensionen oder Kostenträgerdefinitionen eines Mandanten zurück. Kostenträgerdefinitionen, die in der Oberfläche gelöscht wurden, sind noch vorhanden und zeigen in der Oberfläche den Namen `<keine Auswahl>`. Diese werden von der Methode nicht geliefert.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von <KostenDimensionData> mit diesen Feldern:

- DimensionID: Laufende Nummer der Kostendimension.
- Name: Bezeichnung dieser Kostenträgerdefinition.
- Pflicht: Muss dieser Kostenträger bei einer Kostenbuchung angegeben werden oder nicht?
- DBTabelle: Name der Tabelle in der die Kostenträger geführt werden.
- DBSpalte: Spaltenname aus der Tabelle welcher die Kostenträger-Nummer enthält
- AnzeigeSpalte: Spaltenname der Spalte, welche als Bezeichnung des Kostenträgers angezeigt wird
- FirmenSpalte: Spaltenname der Spalte welche die Mandantennummer enthält (i_firm_refid oder i_firmid)
- NrVon: Falls Nummernbereiche angegeben sind die kleinste zulässige „Kostenträgernummer“ welche für diesen Kostenträgerkreis benutzt werden darf. Alphanumerisch, nicht numerisch.
- NrBis: Falls Nummernbereiche angegeben sind die größte zulässige „Kostenträgernummer“ welche für diesen Kostenträgerkreis benutzt werden darf. Alphanumerisch, nicht numerisch.
- Numerisch: Information ob es sich um ein numerisches Feld bei DBSpalte handelt (ob die "Kostenträgernummer" numerisch ist).
- Validieren: Wenn true, werden die eingegebenen Werte validiert. Wenn false, erfolgt keine Prüfung der eingegebenen Werte gegen die vorhandenen Stammdaten – stattdessen wird ein solcher Datensatz angelegt.

KostenStelleExistsKuerzel

Ist ein bestimmtes Kostenstellenkürzel im Mandant schon vergeben?

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kuerzel>string</kuerzel>: Das gewünschte Kostenstellenkürzel.

Rückgabe

Ergebnis True oder False.

KostenStelleExistsNummer

Ist eine bestimmte Kostenstellennummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummer>string</nummer>`: Die gewünschte Kostenstellennummer.

Rückgabe

Ergebnis True oder False.

KostenStelleGet

Gibt die Daten einer Kostenstelle zurück.

In der Kore kann festgelegt werden, ob die Kostenstellen rein numerisch geführt werden sollen. OFCKS.NET behandelt die Kostenstelle jedoch immer alphanumerisch.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kostenStelleID>string</kostenStelleID>`: "Nummer" der gewünschten Kostenstelle, kann alphanumerisch oder numerisch sein.

Rückgabe

- `budget`: Jährliches Budget
- `gueltigBis`: Gültigkeitszeitraum von/bis
- `gueltigVon`: Gültigkeitszeitraum von/bis
- `art`: Kostenstellenart (Enum). Hauptkostenstellen werden von Hilfskostenstellen und allgemeinen Kostenstellen unterschieden.
- `kuerzel`: Kurzbezeichnung
- `name`: Bezeichnung der Kostenstelle
- `nummerID`: alphanumerische eindeutige Kennung der Kostenstelle
- `leiter`: Name des Leiters der Kostenstelle
- `saldo`: Saldo der Kostenstelle (nur Lesen)
- `saldoReset`: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird.

KostenStelleGetOE

Dieselbe Methode wie `KostenStelleGet`, jedoch werden statt Enums die zugehörigen Integer-Zahlen zurückgegeben.

KostenStelleGetAll

Gibt alle Kostenstellen eines Mandanten oder einen eingeschränkten Bereich zurück, mit festlegbarer Sortierung.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummerIDVon>string</nummerIDVon>`: Beginn des Kostenstellenbereichs, falls eingeschränkt werden soll.
- `<nummerIDBis>string</nummerIDBis>`: Ende des Kostenstellenbereichs, falls eingeschränkt werden soll.
- `<sortOrder>Enum</sortOrder>`: Gewünschte Sortierung der Ausgabe.

Rückgabe

Array von `<KostenStelleData>` mit den Feldern wie unter `KostenStelleGet` beschrieben.

KostenStelleGetPeriodenSaldo

Gibt den Saldo einer Kostenstelle für eine Periode aus. Der Saldo wird je Kostenart ausgegeben. Ab Version 5.1.6.0.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kostenStelleID>string</kostenStelleID>`: Kostenstelle, zu der Auskunft gewünscht wird
- `<kalJahr>int</kalJahr>`: (Kalender-)Jahr, für das abgerufen wird.
- `<periode>Enum</periode>`: (Absolute) Periode, für die die Salden gewünscht sind. Es stehen die Perioden 1 bis 12 und 13 zur Verfügung.
- `<kumuliert>bool</kumuliert>`: Wenn true, werden die Werte von Periode 1 bis zur gewünschten Periode kumuliert.

Rückgabe

Array von `<KorePeriodenSaldo>` mit diesen Feldern:

- `<KoArtNummer>int</KoArtNummer>`: Nummer der Kostenart
- `<Fixkosten>double</Fixkosten>`: Saldo der Fixkosten der Periode.
- `<VariableKosten>double</VariableKosten>`: Saldo der variablen Kosten der Periode.

KostenStelleSave

Speichert eine Kostenstelle – Anlegen oder Ändern.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<isNew>boolean</isNew>`: Neuanlage oder Ändern?

Eine Struktur `<data>` mit diesen Feldern:

- `<budget>double</budget>`: Jährliches Budget
- `<gueltigBis>DateTime</gueltigBis>`: Gültigkeitszeitraum von/bis. Kann NULL sein.
- `<gueltigVon>DateTime</gueltigVon>`: Gültigkeitszeitraum von/bis. Kann NULL sein.
- `<art>Enum</art>`: Kostenstellenart (Enum). Hauptkostenstellen werden von Hilfskostenstellen und allgemeinen Kostenstellen unterschieden.
- `<kuerzel>string</kuerzel>`: Kurzbezeichnung
- `<name>string</name>`: Bezeichnung der Kostenstelle
- `<nummerID>string</nummerID>`: alphanumerische eindeutige Kennung der Kostenstelle
- `<leiter>string</leiter>`: Name des Leiters der Kostenstelle
- `<saldoReset>boolean</saldoReset>`: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird. Kann NULL sein.

Rückgabe

Enum der möglichen Returncodes.

KostenSchabloneGet

Gibt die Kopfdaten einer Kostenschablone zurück. Die Verteilungsdaten einer Kostenschablone können nicht ausgelesen werden. Ab 5.1.2.1.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kostenSchabloneID>string</kostenSchabloneID>`: "Nummer" der gewünschten Kostenschablone, kann alphanumerisch oder numerisch sein.

Rückgabe

- `einheit`: Bezugseinheit für die Verteilung

- **gesamtBetrag:** Gesamtbetrag der Verteilung
- **gueltigBis:** Gültigkeitszeitraum von/bis
- **gueltigVon:** Gültigkeitszeitraum von/bis
- **kuerzel:** Kürzel der Kostenschablone
- **name:** Bezeichnung der Kostenschablone
- **nummerID:** Alphanumerische eindeutige Kennung der Kostenschablone

KostenSchabloneGetAll

Gibt die Kopfdaten aller Kostenschablonen eines Mandanten zurück. Von den Kostenschablonen können nur die Kopfdaten gelesen werden. Die Pflege muss in der Kore erfolgen. Ab 5.1.2.1.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von `<KostenSchabloneData>` mit den Feldern wie unter `KostenSchabloneGet` beschrieben.

KostenTraegerExistsKuerzel

Ist ein bestimmtes Kostenträgerkürzel im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kuerzel>string</kuerzel>`: Das gewünschte Kostenträgerkürzel.

Rückgabe

Ergebnis True oder False.

KostenTraegerExistsNummer

Ist eine bestimmte Kostenträgernummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummer>string</nummer>`: Die gewünschte Kostenträgernummer.

Rückgabe

Ergebnis True oder False.

KostenTraegerGet

Gibt die Daten eines Kostenträgers in der Tabelle `kr_kostentraeger` zurück. Anders definierte Kostenträgerkeise sind nicht zugänglich.

In der Kore kann für bis zu 5 Kostenträger-Dimensionen festgelegt werden, in welcher Tabelle die Daten dafür gehalten werden. OFCKS.NET erlaubt die Stammdatenpflege jedoch nur auf der Tabelle `kr_kostentraeger` unabhängig von den Kostenträger-Definitionen. Andere Tabellen liegen in der Datenhoheit anderer Anwendungen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kostenTraegerID>string</kostenTraegerID>`: "Nummer" des gewünschten Kostenträgers, kann alphanumerisch oder numerisch sein.

Rückgabe

- `budget`: Jahres-Budget.
- `gueltigBis`: Gültigkeitszeitraum von/bis
- `gueltigVon`: Gültigkeitszeitraum von/bis
- `kuerzel`: Kurzbezeichnung
- `name`: Bezeichnung
- `nummerID`: alphanumerische eindeutige Kennung des Kostenträgers
- `leiter`: Zuständiger für den Kostenträger
- `saldo`: Saldo des Kostenträgers (nur Lesen)
- `saldoReset`: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird.

KostenTraegerGetAll

Gibt alle Kostenträger eines Mandanten in Tabelle kr_kostentraeger oder einen eingeschränkten Bereich zurück, mit festlegbarer Sortierung. Im Unterschied zur Methode KostenTraegerNGetAll werden die Definitionen der Kostendimensionen nicht beachtet.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<nummerIDVon>string</nummerIDVon>`: Beginn des Kostenträgerbereichs, falls eingeschränkt werden soll.
- `<nummerIDBis>string</nummerIDBis>`: Ende des Kostenträgerbereichs, falls eingeschränkt werden soll.
- `<sortOrder>Enum</sortOrder>`: Gewünschte Sortierung der Ausgabe.

Rückgabe

Array von `<KostenTraegerData>` mit den Feldern wie unter KostenTraegerGet beschrieben.

KostenTraegerGetPeriodenSaldo

Gibt den Saldo eines Kostenträgers für eine Periode aus. Der Saldo wird je Kostenart ausgegeben. Ab Version 5.1.6.0.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kostenTraegerID>string</kostenTraegerID>`: Kostenträger, zu dem Auskunft gewünscht wird
- `<kalJahr>int</kalJahr>`: (Kalender-)Jahr, für das abgerufen wird.
- `<periode>Enum</periode>`: (Absolute) Periode, für die die Salden gewünscht sind. Es stehen die Perioden 1 bis 12 und 13 zur Verfügung.
- `<kumuliert>bool</kumuliert>`: Wenn true, werden die Werte von Periode 1 bis zur gewünschten Periode kumuliert.

Rückgabe

Array von `<KorePeriodenSaldo>` mit diesen Feldern:

- `<KoArtNummer>int</KoArtNummer>`: Nummer der Kostenart
- `<Fixkosten>double</Fixkosten>`: Saldo der Fixkosten der Periode.

- `<VariableKosten>double</VariableKosten>`: Saldo der variablen Kosten der Periode.

KostenTraegerSave

Speichert einen Kostenträger Kostenträgers in der Tabelle `kr_kostentraeger` – Anlegen oder Ändern.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<isNew>boolean</isNew>`: Neuanlage oder Ändern?

Eine Struktur `<data>` mit diesen Feldern:

- `<budget>double</budget>`: Jährliches Budget
- `<gueltigBis>DateTime</gueltigBis>`: Gültigkeitszeitraum von/bis. Kann NULL sein.
- `<gueltigVon>DateTime</gueltigVon>`: Gültigkeitszeitraum von/bis. Kann NULL sein.
- `<kuerzel>string</kuerzel>`: Kurzbezeichnung
- `<name>string</name>`: Bezeichnung des Kostenträgers
- `<nummerID>string</nummerID>`: alphanumerische eindeutige Kennung der Kostenstelle
- `<leiter>string</leiter>`: Name des Leiters der Kostenstelle
- `<saldoReset>boolean</saldoReset>`: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird. Kann NULL sein.

Rückgabe

Enum der möglichen Returncodes.

KostenTraegerNExistsNummer

Ist eine bestimmte Kostenträgernummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<dimension>int</dimension>`: Die Nummer der Kostendimension.
- `<nummer>string</nummer>`: "Nummer" bzw. Schlüssel des gewünschten Kostenträgers.

Rückgabe

Ergebnis True oder False.

KostenTraegerNGet

Gibt die Daten eines Kostenträgers der definierten Kostenträger-Dimension N zurück. Ab Version 5.1.0.0.

In der Kore kann für bis zu 5 Kostenträger-Dimensionen festgelegt werden, in welcher Tabelle die Daten dafür gehalten werden, welches das Suchfeld und welches das Anzeigefeld ist. Details können mit der Methode KostenDimensionGetAll abgerufen werden.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<dimension>int</dimension>`: Die Nummer der Kostendimension.
- `<kostenTraegerID>string</kostenTraegerID>`: "Nummer" bzw. Schlüssel des gewünschten Kostenträgers.

Rückgabe

- `name`: Bezeichnung aus der Anzeigespalte
- `nummerID`: Schlüsselfeld des Kostenträgers

KostenTraegerNGetAll

Gibt die Kostenträger eines Mandanten einer definierten Kostenträger-Dimension N oder einen eingeschränkten Bereich zurück, mit festlegbarer Sortierung. Ab Version 5.1.0.0.

Im Unterschied zur Methode KostenTraegerGetAll müssen diese nicht auf Tabelle `kr_kostentraeger` beruhen. Falls eine Kostendimension auf Tabelle `kr_kostentraeger` beruht, aber nur einen gewissen Nummernkreis umfasst, zeigt sich das in dieser Methode.

Die Daten können nur gelesen werden. Unter Umständen liegen die Tabellen in der Hoheit einer anderen Anwendung.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<dimension>int</dimension>`: Die Nummer der Kostendimension.
- `<nummerIDVon>string</nummerIDVon>`: Beginn des Kostenträgerbereichs, falls eingeschränkt werden soll.

- <nummerIDbis>string</nummerIDbis>: Ende des Kostenträgerbereichs, falls eingeschränkt werden soll.
- <sortOrder>Enum</sortOrder>: Gewünschte Sortierung der Ausgabe. Eine Sortierung nach Kürzel ist nicht möglich, die Kostendimensionen definieren kein Feld Kürzel.

Rückgabe

Array von <KostenTraegerNData> mit den Feldern wie unter KostenTraegerNGet beschrieben.

MandantGet

Gibt die Fibu-Daten eines Mandanten aus. Fast alle Fibu-Daten sind in Mandanten (Firmen) organisiert.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Rückgabe

- alteHauswaehrung: Bei EWS Teilnehmerländern die Währung vor der Umstellung auf den Euro.
- anbuStart: Startdatum der Anbu falls diese installiert ist.
- fibuStart: Startdatum der Fibu.
- geJahrErsteAbsPeriode: Erste absolute Periode eines Geschäftsjahres. Bsp: Fibustart ist der 1.8.2001. GeJahrErsteAbsPeriode = 8
- isKostenstellePflicht: Bei installierter Kore Angabe ob die Kostenstelle eine Pflichtangabe ist bei der Buchung von Kostenkonten.
- kontenrahmen: Name des installierten Kontenrahmens
- land: Landkürzel des Mandanten (der Firma)
- letztesGeJahr: Das neueste Geschäftsjahr das im Mandanten eröffnet wurde
- mahnBetrag (R/W): Betrag ab dem Mahnungen geschrieben werden
- mandantID: Eindeutige Mandantennummer.
- name (R/W): Name des Mandanten.
- periodenbestimmt: Periodenfreier oder periodenbestimmter Buchungsmandanten (Enum).
- steuerNr (R/W): Steuernummer des Mandanten

MandantGetAdressDaten

Gibt die Adressdaten eines Mandanten aus.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

- `branche`
- `faBundesland`: Bundesland des Finanzamts.
- `faOrt`: Ort des Finanzamts.
- `faPLZ`: Postleitzahl des Finanzamts.
- `faStrasse`: Straße des Finanzamts.
- `fax`
- `finanzamt`
- `name`
- `ort`
- `plz`
- `postfach`
- `postfachPLZ`
- `referat` (Österreich)
- `strasse`
- `telefon`
- `titel`
- `uStIDNr`

MandantGetAll

Gibt die Daten aller Mandanten aus.

Aufrufparameter

Keine

Rückgabe

Array von <MandantData> mit den Feldern wie unter MandantGet beschrieben.

MandantSave

Speichert ausgewählte Felder eines Mandanten.

Mandanten müssen in der Fibu angelegt werden. Einzelne Felder können von außen geändert werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Eine Struktur <data> mit den Feldern:

- <mahnbetrag>double</mahnbetrag>: Betrag ab dem Mahnungen geschrieben werden
- <name>string</name>: Name des Mandanten.
- <steuerNr>string</steuerNr>: Steuernummer des Mandanten

Rückgabe

Enum der möglichen Returncodes.

MandantSaveAdressDaten

Speichert die Adressdaten eines Mandanten.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Eine Struktur <data> mit den Feldern:

- <branche>string</branche>: Branche
- <faBundesland>string</faBundesland>: Bundesland des Finanzamts.
- <faOrt>string</faOrt>: Ort des Finanzamts.
- <faPLZ>string</faPLZ>: Postleitzahl des Finanzamts.
- <faStrasse>string</faStrasse>: Straße des Finanzamts.
- <fax>string</fax>: Faxnummer
- <finanzamt>string</finanzamt>: Finanzamt
- <name>string</name>: Mandantenname

- <ort>string</ort>: Ort
- <plz>string</plz>: PLZ
- <postfach>string</postfach>: Postfach
- <postfachPLZ>string</postfachPLZ>: PLZ der Postfach-Adresse
- <referat>string</referat>: Referat (Österreich)
- <strasse>string</strasse>: Strasse
- <telefon>string</telefon>: Telefon
- <titel>string</titel>: Titel
- <uStIDNr>string</uStIDNr>: Umsatzsteuer-Identnummer

Rückgabe

Enum der möglichen Returncodes.

OPGetDeb

Gibt die Daten aller OPs eines Debitorenkontos zurück.

Die Offenen Posten entstehen beim Buchen und können gelesen werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.

Rückgabe

Eine Struktur <OPDataDeb> mit diesen Feldern:

- anmerkung: Anmerkung zum Offenen Posten.
- auftragID: Auftragsnummer
- ausstand: offener Betrag in Belegwährung
- belegDatum
- belegNr: Belegnummer der Fibu
- bruttoBetrag: Gesamtbetrag des OP in OP-Währung
- eigenAusstand: Ausstand des OPs in Mandantenwährung
- eigenBetrag: Bruttobetrag in Mandantenwährung
- historischSeit: Datum seitdem der OP bezahlt ist
- isHistorisch: Handelt es sich um einen historischen oder aktiven OP?
- kategorie

- kontoNr: Debitorenkonto
- mahnDatum: Datum der letzten Mahnung
- mahnstufe: aktuelle Mahnstufe des OPs
- nettoDatum: Frist bis zu dem OP netto fällig ist, ab dann wird gemahnt
- opBelegNr: Rechnungsnummer
- opBemerkung: Buchungstext des OPs
- opRefID: interne OP-Nummer
- opTyp: Typ des OPs (Rechnung, Gutschrift, etc., Enum)
- sprache: Mahnsprache
- valutaDatum
- waehrung: Belegwährung
- wiedervorlageDatum: Datum der Wiedervorlage
- zahlungsSperre: Datum, vor dem der OP nicht bezahlt bzw. eingezogen werden darf.
- zahlzielNr: Nummer des Zahlungszieles

OPGetDebBeleg

Gibt die Daten der OPs einer Belegnummer zurück (Debitoren).

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <belegNr>string</belegNr>: Fibu-Belegnummer.

Rückgabe

Eine Collection der Struktur <OPDataDeb> wie unter OpGetDeb beschrieben. Zu einer Belegnummer kann es ggf. mehrere OPs geben.

OPGetDebOPBeleg

Gibt die Daten der OPs einer Rechnungsnummer – auch OP-Belegnummer genannt – zurück (Debitoren).

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer

- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <opBelegNr>string</opBelegNr>: OP-Belegnummer = Rechnungsnummer.

Rückgabe

Eine Collection der Struktur <OPDataDeb> wie unter OpGetDeb beschrieben. Zu einer Belegnummer kann es ggf. mehrere OPs geben.

OPGetKred

Gibt die Daten aller OPs eines Kreditorenkontos zurück.

Die Offenen Posten entstehen beim Buchen und können gelesen werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.

Rückgabe

Eine Struktur <OPDataKred> mit diesen Feldern:

- anmerkung: Anmerkung zum Offenen Posten.
- ausstand: offener Betrag in Belegwährung
- belegDatum
- belegNr: Belegnummer der Fibu
- bruttoBetrag: Gesamtbetrag des OP in OP-Währung
- eigenAusstand: Ausstand des OPs in Mandantenwährung
- eigenBetrag: Bruttobetrag in Mandantenwährung
- historischSeit: Datum seitdem der OP bezahlt ist
- isHistorisch: Handelt es sich um einen historischen oder aktiven OP?
- kategorie
- kontoNr: Debitorenkonto
- nettoDatum: Frist bis zu dem OP netto fällig ist, ab dann wird gemahnt
- opBelegNr: Rechnungsnummer
- opBemerkung: Buchungstext des OPs
- opRefID: interne OP-Nummer
- opTyp: Typ des OPs (Rechnung, Gutschrift, etc., Enum)
- valutaDatum

- waehrung: Belegwährung
- zahlungsSperre: Datum, vor dem der OP nicht bezahlt bzw. eingezogen werden darf.
- zahlzielNr: Nummer des Zahlungszieles

OPGetKredBeleg

Gibt die Daten der OPs einer Belegnummer zurück (Kreditoren).

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <belegNr>string</belegNr>: Fibu-Belegnummer.

Rückgabe

Eine Collection der Struktur <OPDataKred> wie unter OpGetKred beschrieben. Zu einer Belegnummer kann es ggf. mehrere OPs geben.

OPGetKredOPBeleg

Gibt die Daten der OPs einer Rechnungsnummer – auch OP-Belegnummer genannt – zurück (Kreditoren).

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <kontoNr>int</kontoNr>: Fibu-Kontonummer des gewünschten Kontos.
- <opBelegNr>string</opBelegNr>: OP-Belegnummer = Rechnungsnummer.

Rückgabe

Eine Collection der Struktur <OPDataKred> wie unter OpGetKred beschrieben. Zu einer Rechnungsnummer kann es ggf. mehrere OPs geben.

OPGetVorgaenge

Gibt die Daten aller Vorgänge eines OPs zurück. Ab Version 5.1.4.0

Um diese Methode zu nutzen rufen Sie zuerst eine der OPGet...-Methoden auf. Dadurch erhalten Sie die opRefID. Diese können Sie dann nutzen um mit OPGetVorgaenge die Vorgänge eines OPs abzurufen.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <opRefID>int</ opRefID>: interne OP-Nummer

Rückgabe

Eine Collection der Struktur <OPDataVorgang> mit diesen Feldern:

- typ: Typ des Vorgangs (Kreditor, Debitor)
- auftragID: Auftragsnummer des ERP-Systems.
- azKonto: bei Debitoren das Konto für erhaltene Anzahlungen. Bei Kreditoren das Konto für geleistete Anzahlungen.
- belegDatum: (Fibu-)Belegdatum
- belegRefID: RefID des zugehörigen Buchungsbeleges.
- belegZeileRefID: Zeilennummer der Zeile des zugehörigen Buchungsbeleges.
- bemerkung: Buchungstext
- eigenNettobetrag: Nettobetrag des Vorgangs in Eigenwährung
- eigenSteuerbetrag: Steuerbetrag des Vorgangs in Eigenwährung
- geschaeftsJahr: Geschäftsjahr in dem der Vorgang erfolgte
- inventarNr: nur bei Kreditoren. Inventarnummer des Anlagegutes falls eines beteiligt ist
- nnAZKonto: Bei Debitoren das Konto der noch nicht erhaltenen Anzahlungen (Abschlagsrechnung). Bei Kreditoren das Konto der noch nicht geleisteten Anzahlungen (Abschlagsrechnung).
- opRefID: RefID des OPs
- personenkonto: Personenkonto zu dem der OP gehört.
- relativePeriode: relative Periode in welche der OP gehört.
- skontierfaehig: Ob der OP skontierfähig ist (ungleich 0) oder nicht (0)
- skontoKonto: Konto, auf das ggf. Skonto zu buchen ist.
- steuerKonto: Konto für die Umsatzsteuer bzw. Vorsteuer.
- unterNr: nur bei Kreditoren. Unternummer des Anlagegutes falls eines beteiligt ist
- user: Benutzer der den Vorgang gebucht oder als letzter geändert hat.
- vorgangdatum: (Kalender)Datum an welchem der Vorgang erfolgte.
- vorgangsart: Art des OP Vorgangs. Hier wird das Vorgangskürzel zurückgegeben

OPStatistikDebTop100

Gibt die Top 100 Debitoren (nach Ausstand) zurück mit den wichtigsten OP-Daten.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von `<kontoSaldo>` mit den Feldern

- `konto`: Kontonummer
- `saldo`: Gesamter OP-Ausstand des Kontos

OPStatistikDebTop100Faellig

Gibt die Top 100 Debitoren nach Ausstand nur aus fälligen OPs zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von `<kontoSaldo>` mit den Feldern wie unter `OPStatistikDebTop100` beschrieben.

OPStatistikDebTop100FaelligMinWert

Gibt die Top 100 Debitoren nach Ausstand größer einem Mindestwert nur aus fälligen OPs zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<minWert>double</minWert>`: Mindestwert zur Abgrenzung

Rückgabe

Array von <kontoSaldo> mit den Feldern wie unter OPStatistikDebTop100 beschrieben.

OPStatistikDebTop100Gemahnt

Gibt die Top 100 Debitoren nach Ausstand aus den gemahnten OPs zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Rückgabe

Array von <kontoSaldoGemahnt> mit diesen Feldern:

- konto: Kontonummer
- saldo: Gesamter OP-Ausstand des Kontos / Debitoren
- saldoGemahnt: OP-Ausstand des Kontos aus gemahnten OPs
- mahnstufe: Maximale aktuelle Mahnstufe des Kontos

OPStatistikDebTop100GemahntStufe

Gibt die Top 100 Debitoren nach Ausstand nur aus gemahnten OPs einer Mahnstufe zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <mahnstufe>int</mahnstufe>: Gewünschte Mahnstufe

Rückgabe

Array von <kontoSaldoGemahnt> mit den Feldern wie unter OPStatistikDebTop100Gemahnt beschrieben.

OPStatistikDebTop100MinWert

Gibt die Top 100 Debitoren (nach Ausstand) größer einem Mindestwert zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<minWert>double</minWert>`: Mindestwert zur Abgrenzung

Rückgabe

Array von `<kontoSaldo>` mit den Feldern wie unter `OPStatistikDebTop100` beschrieben.

OPStatistikDebTop100Posten

Gibt die OPs eines Debitoren kompakt aus.

Diese Methode ist letztlich eine Performance-optimierte, weil reduzierte Variante von `OPGetDeb`.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<konto>int</konto>`: Kontonummer des gewünschten Debitoren

Rückgabe

Array von `<kontoPosten>` mit diesen Feldern:

- `konto`: Kontonummer
- `faellig`: Netto-Fälligkeitsdatum
- `offen`: offener Ausstand des OP
- `mahnstufe`: Aktuelle Mahnstufe des OP
- `rechnungsnummer`: OP-Rechnungsnummer des OP

OPStatistikSaldoFristen

Gibt die Salden der Debitoren-OPs gegliedert nach Fälligkeitsfristen zurück.

Es können bis zu 9 Fristen in Tagen in die Vergangenheit und Zukunft mitgegeben werden (z.B. - 30 / 0 / 30 / 60). Die Rückgabe enthält ein Element mehr durch die offenen Enden.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<art>Enum</art>`: Kontoart

- Eine Struktur <fristen> mit den Elementen:
- <frist1>int</frist1>: Frist 1 in Tagen
- <frist2> bis <frist9> wie <frist1>

Rückgabe

Array <fristenSaldo> mit den Elementen:

- frist: die übergebene Frist. Die letzte Frist kommt doppelt zurück, das 2. Element enthält den Saldo mit Fälligkeit größer der letzten Frist.
- fristVon: Beginndatum der Frist.
- fristBis: Endedatum der Frist.
- saldo: Saldo der OPs in dem Zeitraum.

OPStatistikSaldoGesamt

Gibt den Gesamtsalden der Debitoren-OPs einer Kontoart (Debitoren / Kreditoren) zurück.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <art>Enum</art>: Kontoart

Rückgabe

- saldo: Saldo aller OPs der Kontoart.

OPStatistikSaldoMahnstufen

Gibt die Salden der Debitoren-OPs gegliedert nach Mahnstufen zurück.

Die Daten werden ausgewerte gruppiert nach nicht fällig (Mahnstufe -1), fällig und gemahnt nach Mahnstufe. Eine Mindestmahnstufe kann zur Abgrenzung angegeben werden.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <minMahnstufe>int</minMahnstufe>: Mindestmahnstufe

Rückgabe

Array <stufenSaldo> mit den Elementen:

- stufe: Die Mahnstufe. -1 bedeutet noch nicht fällig.
- saldo: Saldo der OPs in dem Zeitraum.

OPStatistikSaldoWochen

Gibt die Salden der Debitoren-OPs gegliedert nach Kalenderwochen zurück.

Die Kalenderwochen werden relativ zur aktuellen Kalenderwochen verwendet. Negative Kalenderwochen erstrecken sich in der Vergangenheit.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <art>Enum</art>: Kontoart
- <wocheVon>int</wocheVon>: Von-Kalenderwoche relativ zur aktuellen Woche.
- <wocheBis>int</wocheBis>: Bis-Kalenderwoche relativ zur aktuellen Woche.

Rückgabe

Array <fristenSaldo> mit den Elementen:

- frist: die übergebene Kalenderwoche.
- fristVon: Beginndatum der Frist.
- fristBis: Endedatum der Frist.
- saldo: Saldo der OPs in der Woche.

RechnungGet

Holt die Daten einer Eingangsrechnung im Rechnungseingangsbuch per ID.

Siehe auch die parallele Methode RechnungGetBelegNr.

Das Zusammenspiel mit dem ERP-System ist so vorgesehen: Eingangsrechnungen werden in der Fibu erfasst. Die Rechnungsprüfung findet im ERP statt, erstellt den Buchungssatz und übergibt diesen an die Fibu und löscht dabei die Rechnung im Rechnungseingangsbuch (logische Löschung).

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

- <rechnungID>int</rechnungID>: Interne Nummer der Eingangsrechnung.

Rückgabe

- rechnungID: interne Nummer der Rechnung
- anmerkung: Anmerkungstext (Langtext)
- auftragID: Auftragsnummer
- bankRefID: Bank
- belegDatum: Belegdatum
- belegNr: Belegnummer
- belegRefID: Interne Buchungsnummer
- betrag: Betrag
- betragNetto: Nettobetrag
- bestellNr: Bestellnummer
- buchungsText: Buchungstext
- clearingFlag: Clearingflag, Zahlungsart (Enum)
- dmsDocumentID: DokumentenID
- eingangsDatum: Eingangsdatum
- einmalAdressRefid: (Konto-)Nummer der Einmaladresse
- esr: ESR-Nummer des Schweizer Zahlungsverkehrs
- geschaeftsjahr: Geschäftsjahr
- kategorie: Interne Nummer der Kategorie
- kennung: Kennung
- konto: Personenkonto
- opBelegNr: OP-Belegnummer/Rechnungsnummer
- periode: Periode
- stapelLfdBuchNr: Laufende Nummer im Buchungsstapel (falls übergeben)
- stapelNr: Nummer des Buchungsstapel (falls übergeben)
- skfBetrag: skontierfähiger Betrag
- status: Status
- valutaDatum: Valutadatum
- waehrung: Währung
- wiedervorlageDatum: Wiedervorlagedatum
- zahlungsSperre: Zahlungssperre
- zahlziel: Zahlungsziel-Kürzel
- buchungskreisID: Nummer des Buchungskreises
- kategorieName: Bezeichnung der Kategorie

RechnungGetAll

Liefert alle Eingangsrechnungen eines Mandanten (unabhängig vom Status).

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von `<RechnungData>` mit den Feldern wie unter `RechnungGet` beschrieben.

RechnungGetOpen

Liefert nur die offenen, nicht in die Fibu übernommenen Eingangsrechnungen eines Kreditors. Ab 3.0.7.0.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<kontoNr>int</kontoNr>`: Kontonummer des Kreditors

Rückgabe

Array von `<RechnungData>` mit den Feldern wie unter `RechnungGet` beschrieben.

RechnungGetBelegNr

Holt die Daten einer Eingangsrechnung per Fibu-Belegnummer. Diese Methode setzt implizit die Eindeutigkeit der Belegnummer voraus.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<belegNr>string</belegNr>`: Fibu-Belegnummer der Eingangsrechnung.

Rückgabe

Beschreibung der Felder siehe RechnungGet.

RechnungGetBelegNrDok

Holt das Sichtdokument und zugehörige Informationen einer Eingangsrechnung per Fibu-Belegnummer. Diese Methode setzt implizit die Eindeutigkeit der Belegnummer voraus.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<belegNr>string</belegNr>`: Fibu-Belegnummer der Eingangsrechnung.

Rückgabe

- `text`: Textliche Rechnungsdaten aus der Originaldatei. Das kann der XML-Text einer X-Rechnung sein, oder auch der reine Text einer PDF-Rechnung.
- `pfad`: Dateiname und Pfad der Rechnungsdatei
- `sicht`: Sichtdokument in Base64-Codierung. Die verwendende Anwendung muss dies ggf. rückumwandeln.
- `keinDokument`: true, falls zur Rechnung kein Dokument hinterlegt ist.

RechnungGetBelegKonto

Holt die Daten der Eingangsrechnungen eines Kreditors per Fibu-Belegnummer (unabhängig vom Status). Ab 3.0.7.0.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<belegNr>string</belegNr>`: Fibu-Belegnummer der Eingangsrechnung.
- `<kontoNr>int</kontoNr>`: Kontonummer des Kreditors

Rückgabe

Array von `<RechnungData>` mit den Feldern wie unter RechnungGet beschrieben.

RechnungSetBelegNrGeloesch

Setzt eine Eingangsrechnung auf gelöscht, identifiziert durch die Belegnummer.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<belegNr>string</belegNr>`: Fibu-Belegnummer der Eingangsrechnung.

Rückgabe

Enum der möglichen Returncodes.

RechnungSetGeloesch

Setzt eine Eingangsrechnung auf gelöscht, identifiziert durch die ID

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<rechnungID>int</rechnungID>`: Interne Nummer der Eingangsrechnung.

Rückgabe

Enum der möglichen Returncodes.

SpracheExistsNummer

Ist eine bestimmte Sprachnummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummer>int</nummer>`: Die gewünschte Nummer der Sprache.

Rückgabe

Ergebnis True oder False.

SpracheGet

Liefert die Daten einer Sprache.

Sprachen werden derzeit im Mahnwesen genutzt. Sie werden in der Schnittstelle mit ihrem Klartext verwendet. Diese Sprachbezeichnungen sollten jedoch mit den in der Fibu-Sprachenverwaltung angelegten Sprachen übereinstimmen. Die Sprachen können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<spracheID>int</spracheID>`: Interne Nummer der Sprache.

Rückgabe

- `bezeichnung`: Bezeichnung der Sprache
- `spracheID`: interne Nummer der Sprache

SpracheGetAll

Liefert alle Sprachen eines Mandanten.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer

Rückgabe

Array von `<SpracheData>` mit den Feldern wie unter `SpracheGet` beschrieben.

StapelBuchungGetAll

Liest Stapelbuchungen aus und gibt sie in einem Array zurück.

Damit können die aktuell in einen Stapel übergebenen, aber noch nicht verbuchten Buchungen ausgelesen werden.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<stapelID>int</stapelID>`: Nummer des Buchungsstapels.

Rückgabe

Array von `<StapelBelegData>` wie unter `StapelBuchungSave` beschrieben.

StapelBuchungSave

Fügt eine Fibu-Buchung in den Buchungsstapel ein.

Die Finanzbuchhaltung kennt die Dialogbuchung (jede Buchung wird sofort verbucht) und die Stapelbuchung (Buchungen werden erfasst, geprüft und anschließend gesammelt verbucht). Vorsysteme müssen Buchungen in einen Buchungsstapel einstellen.

Im Regelfall ist in der Voranwendung hinterlegt, in welchen Buchungsstapel sie (in einem Mandanten) übergibt. Falls eine Auswahl zu treffen ist, ermöglicht `StapelGetAll` das Auslesen der Buchungsstapel.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<stapelID>int</stapelID>`: Nummer des Buchungsstapels.

Eine Struktur `<stapelBuchung>` mit den Kopfdaten einer Stapelbuchung:

- `<auftragID>int</auftragID>`: Ein freies Feld in dem das VORSYSTEM eine eigene Auftragsnummer mitgeben kann. enventa ProFI hat keine auf der Auftragsnummer basierende Funktionalität.
- `<belegDatum>string</belegDatum>` (Pflicht): Das Belegdatum muss als einer der ersten Werte gesetzt werden zusammen mit der Belegwährung. Auf jeden Fall bevor irgendeine der Buchungszeilen-Methoden `AddHaben` oder `AddSoll` aufgerufen werden. Bestimmt in einem periodenfreien Mandant zugleich über die Buchungsperiode.
- `<belegNr>string</belegNr>` (Pflicht): Die externe Belegnummer. Intern bekommt jede Buchung außerdem eine Buchungsnummer, die erst beim Verbuchen vergeben wird.
- `<buchungsart>Enum</buchungsart>`: Art der Buchung. Für übliche Buchungen des täglichen Geschäftsverkehrs ist `baLfdBuchung` zu wählen. Erläuterungen zu den meisten Buchungsarten enthält die Dokumentation zu enventa ProFI. Zusätzliche, an der Oberfläche nicht vorhandene Möglichkeiten:
 - `baAZSchlussKomplettOhneAZ`: Anzahlungsschlussrechnung über den Gesamtbetrag (ohne Anzahlung). Die Standardmethode `baAZSchlussrechnung` dagegen verwendet nur die noch nicht berechneten Differenzbeträge zum Gesamtvolumen.

- baAZSchlussKomplettMitAZ: Anzahlungsschlussrechnung über den Gesamtbetrag mit Ausbuchungszeilen der Anzahlungen (negativ).
- baAZTeilschlussrechnung: Teilschlussrechnung im Anzahlungsverfahren, Sonder-Erweiterung.
- baAZLetzteTeilschlussrechnung: Letzte Teilschlussrechnung, die einen Anzahlungsvorfall mit Teilschlussrechnungen abschließt.
- baAZTSRStornoOffen: Storno ggf. noch offener Anzahlungsrechnungen im Zuge einer Teilschlussrechnung.
- baAZStorno.....: Über die Schnittstelle können alle Rechnungen im Anzahlungsverfahren storniert werden – sofern der Zustand das noch zulässt. Ggf. muss die schon gebuchte Zahlung darauf zuerst in der Oberfläche storniert werden. Die Belegnummer der stornierten Rechnung ist im Element <rechnungID> mitzugeben.
- <dmsDocumentID>string</dmsDocumentID>: ID des zugehörigen Dokumentes im Dokumentenmanagementsystem (DMS).
- <kalenderJahr>int</kalenderJahr>: Kalenderjahr der Buchung. Wird dieses Feld angegeben, so muss auch das Feld „Periode“ gesetzt werden.
- <kategorie>string</kategorie> (optional): Wenn mit Kategorien gebucht wird kann diese hier angegeben werden. Es wird geprüft, ob die Kategorie in den Stammdaten angelegt ist. Diese Prüfung kann deaktiviert werden, indem der Systemparameter OFCK-ValidateKategorie auf 0 gesetzt wird. Beim Ändern von Buchungen im Stapel kann die Kategorie jedoch ggf. verloren gehen.
- <kurs>double</kurs> (optional): Angabe eines Umrechnungskurses. Dies hat rein informativen Charakter. Es werden keine Umrechnungen über diesen Kurs vorgenommen. Ab OFCKS.NET 5.0.6.1: Mit der Sondereinstellung FI-FW-KursAusBeleg=1 wird (bei fehlendem Eigenwährungsbetrag) in Fremdwährungsbelegen mit dem Kurs des Belegs umgerechnet. Dann muss aber konsequent in allen Fremdwährungsbelegen ein Kurs mitgegeben werden!
Bei EWS Mandanten darf jedoch für EWS Währungen kein Kurs angegeben werden!
- <periode>Enum</periode>: bei Mandanteneinstellung „Periodenbestimmt buchen“ absolute (kalendarische) Periode in die gebucht werden soll. (Januar = 1, Februar = 2, ...). Falls nicht angegeben wird die Periode automatisch aus dem Belegdatum ermittelt. Wird dieses Feld angegeben, so muss auch KalenderJahr gesetzt werden.
- <stapelLfdBuchNr>int</stapelLfdBuchNr> (nur Lesen): Erst nach dem Insert verfügbar. Gibt die laufende Stapelbuchungsnummer zurück.
- <stapelNr>int</stapelNr>: Nummer des Stapels, in den gebucht werden soll.
- <stapelID>int</stapelID>: Eigenschaft zum Auslesen der Stapelnummer für die Buchungen. Die Stapelnummer ergibt sich daraus, als welchem Stapel-Objekt heraus man die Stapelbuchung erzeugt hat.
- <verdichten>boolean</verdichten>: Steuerung ob die Fibubuchung und Kostenkontierungen verdichtet werden soll. Kriterien der Verdichtung detailliert in der Dokumentation des OFCKS.NET unter "Hinweise zur Verdichtung".
- <verdichtenNachVZ>boolean</verdichtenNachVZ>: Steuerung ob die Buchungszeilen und Kostenkontierungen verdichtet werden sollen, wobei nur Buchungen gleichen Vorzeichens zusammengefasst werden.
- <waehrung>string</waehrung> (Pflicht): Währung der Buchung. Diese Eigenschaft muss gesetzt werden. Ggf. muss die Hauswährung mitgegeben werden.
- <zahlungsSperre>string</zahlungsSperre>: für OP-Anlage mit Zahlungssperre: Datum, bis zu dem der OP gesperrt sein soll.

- `<anzahlungID>int</anzahlungID>`: Nummer des Anzahlungsvorfalles, wird nur beachtet bei Rechnungsbuchungen im Anzahlungsverfahren (Abschlags-, Schlussrechnungen).
- `<quelle>string</quelle>`: In diesem Feld kann und sollte die verwendende Anwendung ihren Namen mitgeben. Die Buchungsherkunft kann dann in der Finanzbuchhaltung rückverfolgt werden. Es stehen zwar 50 Zeichen zur Verfügung, Sie sollten jedoch eine in bis zu 15 Zeichen lesbare Bezeichnung verwenden. Wird das Feld nicht gefüllt, trägt der OFCKS.NET Webdienst sich als Buchungsherkunft ein.
- `<freierWert1>string</freierWert1>`: Übergabe der bis zu 15 freien Werte für einen Beleg. In der Fibu werden diese als Belegzusätze bezeichnet. Deren Bezeichnungen können definiert werden.
- `<freierWert2>` bis `<freierWert15>`: Die weiteren freien Werte.
- `<koreAutoKorrekturBetrag>double</koreAutoKorrekturBetrag>` (optional): Erlaubt die Vorgabe einer Kleindifferenzgrenze, unter der Abweichungen zwischen Fibu und Kore automatisch ausgeglichen werden sollen. Aus Kompatibilitätsgründen gilt: 0 = beliebig große Abweichungskorrektur, andere Beträge (i.d.R. 0.01) stellen die Kleindifferenzgrenze dar.
- `<buchungskreisID>int</buchungskreisID>`: Buchungskreisnummer. Falls nicht angegeben wird die Standardbuchungskreisnummer 1 gesetzt.

Die Buchungszeilen werden mit einem Array von `<StapelBelegZeileData>` übergeben, das als `<zeileData>` in die Stapelbuchungsstruktur eingehängt wird. Die Felder der Buchungszeile sind:

- `<sollHabenKennzeichen>Enum</sollHabenKennzeichen>` Soll- oder Habenbuchungszeile.
- `<konto>int</konto>`: bebuchtes Konto
- `<einmalAdressRefid>int</einmalAdressRefid>`: "Kontonummer" der Einmal-Adresse (des Einmal-Kreditors oder –Debitors) für diese Buchung. Einmalkonten können auch ohne Einmaladresse bebucht werden. Dies sollte jedoch aus GOB-Grundsätzen nur in Ausnahmefällen genutzt werden.
- `<bruttoBetrag>double</bruttoBetrag>`: Bruttobetrag der auf dieses Konto gebucht werden soll in Buchungswährung.
- `<eigenBetrag>double</eigenBetrag>`: (muss/optional) Bruttobetrag in Hauswährung des Mandanten (immer in Hauswährung zum Zeitpunkt der Buchung, NICHT frühere Hauswährung vor Euro-Umstellung). Muss nur angegeben werden bei Fremdwährungsbuchungen (nicht nötig bei Buchungen in Hauswährung oder alter Hauswährung). Kann mit Hilfe der Funktion "EigenBetrag" des Mandant-Objekts errechnet werden.
Wird 0 übergeben werden die Eigenbeträge intern über den hinterlegten Kurs berechnet und falls nicht NULL übergeben wurde zurückgegeben. Ein evtl. mitgegebener Kurs wird nicht zur Umrechnung benutzt (Ausnahme siehe oben zum Kurs). Empfehlung: Betrag in Belegwährung und Eigenwährung mitgeben, dann besteht definitiv Betragsgleichheit zum Vorsystem.
- `<buchungsText>string</buchungsText>`: (optional) Der Buchungstext der 1. Zeile wird zum Belegkopf-Buchungstext.
- `<steuerArtID>int</steuerArtID>`: Es kann eine vom Konto abweichende Steuerart angegeben werden. Soll der Default des Kontos verwendet werden so muss hier –1 übergeben werden.
Es wird empfohlen diesen Parameter zu übergeben, insbesondere wenn ein Konto mit wechselnden Vorsteuersätzen bebucht wird.
Achtung: vom Bebuchen von Erlöskonten mit wechselnden Umsatzsteuersätzen wird dringend abgeraten, da dies zu Abweichungen bei der Umsatzsteuervoranmeldung führen kann!

- `<steuerBetrag>double</steuerBetrag>`: Steuerbetrag in Buchungswährung, falls auf dieses Konto mit Steuer gebucht wird.
- `<eigenSteuerBetrag>double</eigenSteuerBetrag>`: (muss/optional) Steuerbetrag in Hauswährung. Es gelten die Regeln für den EigenBetrag.
- `<kostenSchablone>string</kostenSchablone>`: (optional) Angabe einer Schablone zur Kostenaufteilung, nur mit installierter Kore erlaubt.
- `<kostenArt>int</kostenArt>`: (optional) Falls abweichend kann hier bei installierter Kore die Kostenart angegeben werden.
- `<valutaDatum>string</valutaDatum>`: (optional) falls nicht angegeben wird das Buchungsdatum als Valutadatum verwendet.
- `<opBelegNr>string</opBelegNr>`: (optional) Rechnungsnummer des OPs. Wenn nicht gefüllt, wird er gleich der Fibu-Belegnummer gesetzt.
- `<zahlziel>string</zahlziel>`: (optional) Bezeichnung (nicht ZahlzielRefID) des Zahlzieles, auf der Personenkonto-Seite der Buchung zu setzen.
Hierbei kann das Kürzel eines in enventa ProFI definierten Zahlziels oder ein individueller Zahlzielstring nach folgendem Muster verwendet werden:
NnSsPp,pTtQq,qMmZz
n=Nettotage, s=Skontotage, p=Skontoprozent, t=Skontotage 2, q=Skontoprozent 2,
m=Zielmonat (0=laufender, 1=nächster), z=Zieltag im Monat.
Beispiel:
N30S10P2,5 = netto 30 Tage, 10 Tage Skonto mit 2,5%
N60S10P2,5T20Q1,0 = netto 60 Tage, 10 Tage Skonto mit 2,5%, 20 Tage Skonto mit 1,0% (zweistufiges Skonto)
- `<esr>string</esr>`: (optional) Angabe einer ESR-Nummer für Zahlungsverkehr in der Schweiz.
- `<inventarNr>string</inventarNr>`: Inventarnummer des zu bebuchenden Anlagegutes für einen Teilzugang. Bei einem Neuzugang ohne geplantem Anlagegut muss hier ein Leerstring übergeben werden.
- `<unterNr>int</unterNr>`: Unternummer des zu bebuchenden Anlagegutes für einen Teilzugang. Bei einem Neuzugang ohne geplantem Anlagegut muss hier eine 0 übergeben werden.
- `<anbuKostenStelle>string</anbuKostenStelle>`: Kostenstelle für das Anlagegut (auf die später die kalk. AfA der Anlage gebucht wird). Angabe ist optional.
- `<clearingFlag>Enum</clearingFlag>`: Zahlungsart. Falls nicht wie im Personenkonto hinterlegt bezahlt werden soll kann hier eine abweichende Clearingmethode angegeben werden. Bei Übergabe von cfUseDefault wird die Zahlungsart aus dem Konto verwendet.
- `<bankKontoNr>string</bankKontoNr>`: Kontonummer der Bank
- `<bankBLZ>string</bankBLZ>`: Bankleitzahl der Bank
- `<bankName>string</bankName>`: Name der Bank
- `<sprache>string</sprache>`: Sprache, in der der OP ggf. gemahnt werden soll.
- `<bankLand>string</bankLand>`: Land der Bank für Zahlungswege, bei denen die Adresse der Bank notwendig ist.
- `<bankIban>string</bankIban>`: IBAN der Bankverbindung.
- `<bankSwift>string</bankSwift>`: Swiftcode oder BIC der Bankverbindung.
- `<bankText1>string</bankText1>` bis `<bankText5>`: Weitere Angaben zur Adresse für Zahlungswege, bei denen die Adresse der Bank notwendig ist. Die Felder sind belegt Name1 / Name2 / Ort / PLZ / Strasse.

- `<skontierFaehig>Enum</skontierFaehig>`: Die Skontierfähigkeit ist an den Sachbuchungszeilen anzugeben und kann gemischt angegeben werden. Mit `skfEntsprechendStamm` wird die Skontierfähigkeit aus dem Sachkonto verwendet. Standard ist `skfNicht`, deshalb sollte bei Rechnungsbuchungen eine Skontierfähigkeit gesetzt werden.
- `<rechnungID>string</rechnungID>`: Übergabe der RechnungID für den Sachkonten-OP pro Stapelzeile.
- `<ustIDNr>string</ustIDNr>`: Individuelle Umsatzsteuer-Identnummer zur Buchung
- `<sepaMandat>string</sepaMandat>`: SEPA-Einzelformat zum OP oder das zu verwendende Dauermandat.
- `<abzugsfaehig>double</abzugsfaehig>`: Prozentsatz einer nur teilweisen Vorsteuer-Abzugsfähigkeit (sonst NULL oder nicht vorhanden). Eine Abzugsfähigkeit von 0 % kann über die Schnittstelle nicht gesetzt werden, wohl aber Prozentsätze > 0, auch 100 %.
- `<sepaTermin>dateTime</sepaTermin>`: Gewünschter Ausführungstermin einer SEPA-Lastschrift.
- `<abgrenzung>string</abgrenzung>`: (optional, ab Version 5.1.1.2) In Abhängigkeit vom übergebenen Wert und der Abgrenzungskonfiguration des bebuchten Sachkontos wird die Abgrenzung durchgeführt. Mögliche Werte:
 - "0" = nicht abgrenzen
 - "1" = abgrenzen
 - Importtoken (ab Version 5.1.4.0)
 Der **Wert "1"** wird nur in Datenbanken ab enventa -ProFI-Version 2022 (13.x) und von StapelBuchungSave ab Version 5.1.1.2 unterstützt. Ab Version 5.1.3.0 wird er auch von StapelBuchungSaveAndBook unterstützt.

Importtoken werden nur in Datenbanken ab enventa -ProFI-Version 2024 (14.x) und von der Methode Insert ab Version 5.1.4.0 unterstützt. Über ein Importtoken können detaillierte Informationen zur Art und Weise der Abgrenzung übergeben werden, mit folgenden Elementen:

- **Abgrenzen**: 1 | 0
- **Jahr**: Geschäftsjahr der ersten Auflösung
- **Periode**: abs. Periode der ersten Auflösung
- **Zeitintervall**: Intervall der Auflösungen in Monaten
- **Anzahl**: Anzahl Auflösungsbuchungen
- **Jahresabgrenzung**: 1 | 0

Aufbau des Importtokens:

a) für monatsbasierte Abgrenzungen (i.d.R. mit mehreren Auflösungen):

`Import;<Abgrenzen>;Per;<Jahr>;<Periode>;Int;<Zeitintervall>;Anz;<Anzahl>;`

Bsp. 1: Abgr. Monatlich 12x ab September im GJ 2023:

`Import;1;Per;2023;9;Int;1;Anz;12;`

Bsp. 2: Abgr. Quartalsweise 4x ab Februar im GJ 2022:

`Import;1;Per;2022;2;Int;3;Anz;4;`

b) für Jahresabgrenzungen (mit 1 Auflösung zu Beginn des nachfolgenden Geschäftsjahres):

`Import;<Abgrenzen>;Jabgr;<Jahresabgrenzung>;`

Bsp: Import;1;Jabgr;1;

Fehlende Angaben werden durch die Standardwerte der betroffenen Sachkonten ersetzt.

Für Buchungszeilen auf ein Kostenkonto kann entweder die Kostenschablone in der Buchungszeile angegeben werden oder es muss ein Array von `<StapelKoreData>` im Element `<koreData>` mitgegeben werden.

- `<nettoEigenBetrag>double</nettoEigenBetrag>`: Der angegebene Kore-Betrag in Eigenwährung muss dem Fibu-Nettobetrag in Eigenwährung(!) entsprechen. Dies gilt pro Buchungszeile für die Summe der Kostenzeilen-Beträge. Es gilt auch für das Vorzeichen, OFCKS.NET gibt Kostenbeträgen zu Habenbuchungen dann das umgekehrte Vorzeichen.

- `<nettoBetrag>double</nettoBetrag>`: Alternativ zum Kore-Betrag in Eigenwährung kann auch der Betrag in Belegwährung angegeben werden.
- `<kostenStelle>string</kostenStelle>`: Kostenstelle falls erforderlich. Ansonsten ein Leerstring, falls in Kore nicht als Pflicht eingestellt.
- `<kostenTraeger1>string</kostenTraeger1>`: (optional). Angabe des Merkmals, das in der Kostenrechnung als Kostenträger 1 definiert wurde. Es müssen die Einstellungen der Kore berücksichtigt werden, z.B. wenn Kostenträger als Pflichtfeld deklariert wurden.
- `<kostenTraeger2>` bis `<kostenTraeger5>`: siehe `<kostenTraeger1>` für die bis zu 5 Kostenträger-Dimensionen, die definiert werden können.
- `<fixVariabel>Enum</fixVariabel>`: Fixkosten (= fvFix, default) oder Variabel (= fvVariabel). Nur notwendig, wenn in dem Mandant die Unterscheidung getroffen wird.
- `<menge>double</menge>`: optionale Angabe der Menge, falls mit Mengen gebucht wird.
- `<buchungsText>string</buchungsText>`: Buchungstext

Rückgabe

Enum der möglichen Returncodes.

StapelBuchungSaveAndBook

Fügt eine Fibu-Buchung in den Buchungsstapel ein, prüft sie und verbucht sie.

Die Methode ist eine Erweiterung der Methode `StapelBuchungSave` um die nachgeschalteten Funktionen des Prüfens und Verbuchens.

Hinweise: In den Schritten Einfügen, vor allem aber Prüfen und Verbuchen kann es zu vielfältigen Fehlern kommen. Für die Schritte Prüfen und Verbuchen werden die entsprechenden Routinen von `enventa ProFI` genutzt. Die sozusagen üblichen vorhersehbaren Fehler werden abgefangen und rückgemeldet. Es kann jedoch auch zu Fehlerbedingungen kommen, die dazu führen, dass das Programm versucht, eine Fehlermeldung an den Bildschirm zu bringen und auf die Bestätigung durch den Anwender zu warten.

Was dann passiert, hängt von der Einbindung durch die aufrufende Anwendung ab. Ist es wie in diesem Fall ein Webdienst, kommt es zu einem Fehler `InvalidOperationException`.

Grundsätzlich ist die Art der verarbeiteten Buchungen nicht beschränkt. Getestet wurde jedoch nur ein Kranz üblicher Schnittstellenbuchungen wie einfache Sachbuchungen oder Rechnungen. Werden andere Buchungsarten übergeben und mit dieser Methode geprüft und verbucht, müssen durch die einbindende Anwendung auch Fehlerfälle und deren Verhalten getestet werden.

Aufrufparameter

Dieselben wie die Methode `StapelBuchungSave`.

Rückgabe

- `<ergebnis>Enum</ergebnis>`: Enum des Erfolgs oder der möglichen Fehlerschritte.

- `<fehlermeldung>string</fehlermeldung>`: Die Fehlermeldung im Fehlerfall.
- `<protokoll>string</protokoll>`: Das Protokoll im Erfolgsfall.
- `<buchungsnummer>int</buchungsnummer>`: Gibt im Erfolgsfall die Buchungsnummer (i_beleg_refid) der verbuchten Buchung zurück, sonst 0.

StapelDelete

Löscht einen Buchungsstapel.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<stapelNr>int</stapelNr>`: Nummer des Buchungsstapels.

Rückgabe

Enum der möglichen Returncodes.

StapelExistsBelegNr

Prüft, ob eine Buchung mit der angegebenen Belegnummer im angegebenen Stapel vorhanden ist. Voranwendungen können damit Doppelübergaben vermeiden, insbesondere nach Fehlern, die mit der Methode `StapelbuchungSaveAndBook` beim Prüfen oder Verbuchen auftraten. Denn dann steht die Buchung schon im Stapel.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<stapelNr>int</stapelNr>`: Der zu prüfende Stapel.
- `<belegNr>string</belegNr>`: Die zu prüfende (Fibu-)Belegnummer.

Rückgabe

Ergebnis True oder False.

StapelExistsNummer

Ist eine Stapelnummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<nummer>int</nummer>`: Die gewünschte Stapelnummer.

Rückgabe

Ergebnis True oder False.

StapelGet

Gibt die Daten eines Stapel zurück.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<stapelNr>int</stapelNr>`: Nummer des Buchungstapels.

Rückgabe

- `<faelligkeitsDatum>dateTime</faelligkeitsDatum>`: Ende-Datum für diesen Stapel
- `<isWiederkehrend>boolean</isWiederkehrend>`: Ist es ein wiederkehrender Stapel?
- `<stapelArt>Enum</stapelArt>`StapelArt: Art des Stapels. Mögliche Werte:
 - `staBrutto`: Bruttostapel
 - `staNettoInclSteuer`: Netto Stapel
 - `staKoreStapel`: Kostenrechnungstapel. Nur für Kostenrechnung.
- `<bezeichnung>string</bezeichnung>`: Bezeichnung
- `<stapelID>int</stapelID>`: Eindeutige Nummer des Stapels
- `<wiederkehrArt>Enum</wiederkehrArt>`: Art der Wiederkehr (Enum)
- `<wiederkehrendIn>int</wiederkehrendIn>`: Anzahl Tage oder Monate, in denen der Stapel wieder verbucht werden soll.

StapelGetAll

Gibt alle Buchungstapel eines Mandanten zurück, ggf. gefiltert

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer
- <filterStapelArt>Enum</filterStapelArt>: Auswahl, welche Stapel in der Rückgabe enthalten sein sollen – oder alle.

Rückgabe

Array von <StapelData> mit den unter StapelGet beschriebenen Feldern.

StapelSave

Speichert einen Buchungsstapel - Anlegen oder Ändern.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantenummer
- <isNew>boolean</isNew>: Neuanlage oder Ändern?

Struktur <data> mit diesen Feldern:

- <faelligkeitsDatum>dateTime</faelligkeitsDatum>: Ende-Datum für diesen Stapel
- <isWiederkehrend>boolean</isWiederkehrend>: Ist es ein wiederkehrender Stapel?
- <stapelArt>Enum</stapelArt>StapelArt: Art des Stapels. Mögliche Werte:
 - staBrutto: Bruttostapel
 - staNettoInclSteuer: Netto Stapel
 - staKoreStapel: Kostenrechnungsstapel. Nur für Kostenrechnung.
- <bezeichnung>string</bezeichnung>: Bezeichnung
- <stapelID>int</stapelID>: Eindeutige Nummer des Stapels
- <wiederkehrArt>Enum</wiederkehrArt>: Art der Wiederkehr (Enum)
- <wiederkehrendIn>int</wiederkehrendIn>: Anzahl Tage oder Monate, in denen der Stapel wieder verbucht werden soll.

Rückgabe

Enum der möglichen Returncodes.

StapelCheckAndBook

Prüft und verbucht die Buchungen in einem Buchungsstapel. Die Buchungen sind auf anderem Wege in den Buchungsstapel gekommen. Die Methode ist auf Fibu-Buchungsstapel und Kore-Buchungsstapel anwendbar.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<stapelID>int</stapelID>`: Eindeutige Nummer des Stapels

Rückgabe

- `ergebnis`: Enum der möglichen Fehler- und Ergebniszustände.
- `fehlermeldung`: Wenn die Funktion mit einem Fehler abgebrochen wurde, die Fehlermeldung oder das Fehlerprotokoll.
- `protokoll`: Das Protokoll des Prüf- oder Verbuchungsvorgangs.

SteuerArtExistsNummer

Ist eine bestimmte Steuerartnummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummer>int</nummer>`: Die gewünschte Steuerartnummer.

Rückgabe

Ergebnis True oder False.

SteuerArtGetAll

Liefert die Steuerarten (Steuerschlüssel). Siehe auch Methode `SteuerArtGetAllSatz` (ab 5.0.8.0). Steuerarten können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Rückgabe

Array von <SteuerArtData> mit diesen Feldern:

- beschreibung: Erweiterte Beschreibung der Steuerart
- bezeichnung: Bezeichnung, Steuerschlüssel
- steuerartID: Nummer der Steuerart

SteuerArtGetAllSatz

Liefert die Steuerarten (Steuerschlüssel) mit erweiterten Informationen. Dazu gehört auch der Steuersatz zum gewünschten Datum. Siehe auch Methode SteuerArtGetAll.

Steuerarten können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer
- <satzDatum>dateTime</satzDatum>: Datum, zu dem der Steuersatz ermittelt werden soll.

Rückgabe

Array von <SteuerArtSatzData> mit diesen Feldern:

- steuerartID: Nummer der Steuerart
- bezeichnung: Bezeichnung, Steuerschlüssel
- beschreibung: Erweiterte Beschreibung der Steuerart
- steuerKonto: Konto auf das standardmäßig die Steuer gebucht wird
- skontoKonto: Konto auf das standardmäßig der Skonto gebucht wird
- isVstUst: Kennzeichen, ob es sich um eine Steuerart mit VSt/USt-Automatik handelt wie EG-Erwerb oder inländische Reverse Charge Fälle.
- satz: Der Steuersatz zum gewünschten Datum. Ist zum Datum kein Steuersatz definiert, ist dieses Feld mit 0 gefüllt, auch die Steuerkonten sind dann leer.

SteuerSatzExistsNummer

Ist eine bestimmte Steuersatz-ID im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<nummer>int</nummer>`: Die gewünschte Steuersatz-ID.

Rückgabe

Ergebnis True oder False.

Version

Gibt die Versionsnummer als Konstante zurück.

Aufrufparameter

Keine

Rückgabe

Versionsnummer des Webdienstes als Zahl.

ZahlzielDelete

Löscht ein Zahlungsziel.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<zahlzielID>int</zahlzielID>`: Interne Nummer des Zahlungsziels.

Rückgabe

Enum der möglichen Returncodes.

ZahlzielExistsKuerzel

Ist ein bestimmtes Zahlzielkürzel im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<kuerzel>string</kuerzel>`: Das gewünschte Zahlzielkürzel.

Rückgabe

Ergebnis True oder False.

ZahlzielExistsNummer

Ist eine bestimmte Zahlzielnummer im Mandant schon vergeben?

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<nummer>int</nummer>`: Die gewünschte Zahlzielnummer.

Rückgabe

Ergebnis True oder False.

ZahlzielGet

Stellt die Daten eines Zahlungsziels zur Verfügung.

Für die üblichen Zahlungsziele in Abstandstagen gilt die Regel, dass das Kürzel (Bezeichnung) eindeutig sein muss. Es gibt jedoch außerdem die Zahlungsziele mit Gültigkeit für Bereiche eines Monats. Hier gilt, dass es unter einem Kürzel mehrere Einträge gibt, die mit ihrem Gültig-Ab-Datum den ganzen Monat abdecken müssen.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantenummer
- `<zahlzielID>int</zahlzielID>`: Interne Nummer des Zahlungsziels.

Rückgabe

- `beschreibung`: Erweiterte Beschreibung des Zahlungsziels

- bezeichnung: eindeutige Bezeichnung des Zahlziels, in der Fibu "Kürzel". Diese Eigenschaft kann nur bei neuen Zahlzielen eingegeben werden. Es ist nicht möglich die Bezeichnung eines bestehenden Zahlzieles zu ändern.
- zahlzielID: Interne Nummer des Zahlungsziels.
- monatePlus (Zahlziele mit Gültigkeitsbereich): Monatsversatz für die Nettofälligkeit. 0, wenn der Nettofälligkeits-Zieltag im Monat der Entstehung des OPs liegt.
- nettotage (Zahlziele in Abstandstagen): Anzahl an Tagen bis zu dem der Betrag in Netto fällig ist.
- zieltag (Zahlziele mit Gültigkeitsbereich): Zieltag im Monat für die Nettofälligkeit.
- skontoProzent1 : Skonto-Prozentsatz für die erste Skontofrist.
- skontoProzent2 : Skonto-Prozentsatz für die zweite Skontofrist.
- skontotage1 (Zahlziele in Abstandstagen): Anzahl Tage für die erste Skontofrist.
- skontotage2 (Zahlziele in Abstandstagen): Anzahl Tage für die zweite Skontofrist.
- monatSkontoTage1 (Zahlziele mit Gültigkeitsbereich): Zieltag und Monatsversatz für die erste und zweite Skontofälligkeit. Achtung: Monate mit 100 multipliziert addieren!
- monatSkontoTage2 (Zahlziele mit Gültigkeitsbereich): Zieltag und Monatsversatz für die erste und zweite Skontofälligkeit. Achtung: Monate mit 100 multipliziert addieren!
- nurNettoSkontierf: Skontierfähiger Betrag aus Netto.
- zahltag (Zahlziele mit Gültigkeitsbereich): Beginn des Gültigkeitsbereichs. In der Fibu "Gültig ab (Tag im Monat)".

ZahlzielGetAll

Stellt die Daten aller Zahlungsziele eines Mandanten zur Verfügung.

Aufrufparameter

- <mandantID>int</mandantID>: Mandantennummer

Rückgabe

Array von <ZahlzielData> mit den unter ZahlzielGet beschriebenen Feldern.

ZahlzielSave

Speichert ein Zahlungsziel - Anlegen oder Ändern.

Aufrufparameter

- `<mandantID>int</mandantID>`: Mandantennummer
- `<isNew>boolean</isNew>`: Neuanlage oder Ändern?

Struktur `<data>` mit diesen Feldern:

- `<beschreibung>string</beschreibung>`: Erweiterte Beschreibung des Zahlungsziels
- `<bezeichnung>string</bezeichnung>`: eindeutige Bezeichnung des Zahlziels, in der Fibu "Kürzel". Diese Eigenschaft kann nur bei neuen Zahlzielen eingegeben werden. Es ist nicht möglich die Bezeichnung eines bestehenden Zahlzieles zu ändern.
- `<zahlzielID>int</zahlzielID>`: Interne Nummer des Zahlungsziels.
- `<monatePlus>int</monatePlus>` (Zahlziele mit Gültigkeitsbereich): Monatsversatz für die Nettofälligkeit. 0, wenn der Nettofälligkeit-Zieltag im Monat der Entstehung des OPs liegt.
- `<nettotage>int</nettotage>` (Zahlziele in Abstandstagen): Anzahl an Tagen bis zu dem der Betrag in Netto fällig ist.
- `<zieltag>int</zieltag>` (Zahlziele mit Gültigkeitsbereich): Zieltag im Monat für die Nettofälligkeit.
- `<skontoProzent1>double</skontoProzent1>` : Skonto-Prozentsatz für die erste Skontofrist.
- `<skontoProzent2>double</skontoProzent2>` : Skonto-Prozentsatz für die zweite Skontofrist.
- `<skontoTage1>int</skontoTage1>` (Zahlziele in Abstandstagen): Anzahl Tage für die erste Skontofrist.
- `<skontoTage2>int</skontoTage2>` (Zahlziele in Abstandstagen): Anzahl Tage für die zweite Skontofrist.
- `<monatSkontotage1>int</monatSkontotage1>` (Zahlziele mit Gültigkeitsbereich): Zieltag und Monatsversatz für die erste und zweite Skontofälligkeit. Achtung: Monate mit 100 multipliziert addieren!
- `<monatSkontotage2>int</monatSkontotage2>` (Zahlziele mit Gültigkeitsbereich): Zieltag und Monatsversatz für die erste und zweite Skontofälligkeit. Achtung: Monate mit 100 multipliziert addieren!
- `<nurNettoSkontierf>boolean</nurNettoSkontierf>`: Skontierfähiger Betrag aus Netto.
- `<zahltag>int</zahltag>` (Zahlziele mit Gültigkeitsbereich): Beginn des Gültigkeitsbereichs. In der Fibu "Gültig ab (Tag im Monat)".

Rückgabe

Enum der möglichen Returncodes.

Index

Keine Indexeinträge gefunden.