



## SUPA - Subsembly Payments

### *Dateiformat und JavaScript-Schnittstelle zum Austausch von Zahlungsverkehrsdaten*

### Version 2.0



Creative Commons  
Namensnennung-Keine Bearbeitung 3.0  
Deutschland

Dieses Werk ist unter einem Creative Commons Namensnennung-Keine Bearbeitung 3.0 Deutschland Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by-nd/3.0/de/> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Es wurden alle Anstrengungen unternommen um die Richtigkeit des Dokuments sicher zu stellen. Subsembly GmbH übernimmt jedoch keine Garantie hinsichtlich der Richtigkeit oder Vollständigkeit. Die Tauglichkeit oder Eignung für einen bestimmten Zweck wird nicht gewährleistet. Die enthaltenen Informationen können ohne besondere Ankündigung geändert werden. Ein Rechtsanspruch ist hieraus nicht ableitbar.

Copyright © 2010-2014 Subsembly GmbH.

## Inhaltsverzeichnis

1 Vorwort .....	3
2 Generelles Dateiformat .....	4
3 SUPA-Dateitypen .....	5
3.1 Kontoinformationen .....	5
3.2 Zahlungsaufträge .....	6
3.3 Kontoumsätze .....	10
3.4 Kontoverbindungen .....	13
4 Datentypen .....	15
5 JavaScript-Schnittstelle .....	16
5.1 JSON .....	16
5.2 Funktion Login .....	17
5.3 Funktion Logout .....	18
5.4 Funktion DownloadBalance .....	19
5.5 Funktion DownloadStatement .....	21

# 1 Vorwort

SUPA ist der Name für eine Familie von Dateiformaten zum Austausch von Zahlungsverkehrsdaten, wie zum Beispiel Überweisungs- und Lastschriftaufträge, Kontoinformationen und Kontoumsätze, sowie Kontoverbindungen.

Bei der Definition der SUPA Dateiformate wurde darauf geachtet, dass die Dateien möglichst einfach erzeugt und verarbeitet werden können. Inhaltlich lehnen sich die SUPA Dateiformate an den SEPA XML Datenformaten an.

Ab Version 2.0 definiert die SUPA Spezifikation zusätzlich eine JavaScript-Schnittstelle zum automatisierten Abruf von Kontoständen (Balance) und Kontoumsätzen (Statement). Dienstleister oder Banken können diese Schnittstelle zur Verfügung stellen und so den Zugriff auf diese Daten für Finanzsoftware ermöglichen. Die SUPA JavaScript-Schnittstelle wird in den Subsembly Banking Apps für die Integration externer Screen-Scraper unterstützt.

## 2 Generelles Dateiformat

SUPA-Dateien sind einfache Textdateien mit Tabulator separierten Inhalten (**Tab-Separated-Values**), wie sie zum Beispiel auch von Kalkulationsprogrammen wie Microsoft Excel geladen und gespeichert werden können. Diese Dateiart ist offiziell als MIME-Typ registriert und spezifiziert. Die Spezifikation kann unter <http://www.iana.org/assignments/media-types/text/tab-separated-values> abgerufen werden.

Das Tabulatorzeichen hat den Code 0x09. Eine Zeile wird durch eine CR-LF-Sequenz (Codes 0x0D 0x0A) oder einem einfachen LF (Code 0x0A) abgeschlossen.

Im Gegensatz zu den im SEPA Zahlungsverkehr eingeschränkten Basiszeichensatz können in SUPA-Dateien alle Zeichen des westeuropäischen Latin-1 Zeichensatzes (Siehe ISO 8859-1 [http://de.wikipedia.org/wiki/ISO\\_8859-1](http://de.wikipedia.org/wiki/ISO_8859-1)) verwendet werden.

Die nicht-druckbaren Zeichen haben eine spezielle Bedeutung und dürfen innerhalb von Datenfeldern nicht vorkommen.

Die erste Zeile einer SUPA-Datei ist eine Kopfzeile mit Spaltennamen. Diese Spezifikation definiert die für die verschiedenen SUPA-Dateien verwendeten Spaltennamen und die Semantik und Syntax für die zugehörigen Inhalte.

Die Reihenfolge der Spalten ist generell nicht vorgeschrieben. Ebenso sind viele Spalten optional und können in einer Datei auch weggelassen werden. In gleicher Weise können beliebige zusätzliche Spalten von einer Anwendung hinzugefügt werden. Beim Import einer SUPA-Datei müssen die Inhalte der Spalten ausschließlich über die zugehörigen Spaltenüberschriften und nicht über die Spaltenposition zugeordnet werden. Unbekannte Spalten dürfen beim Import ignoriert werden.

Um SUPA-Dateien leichter zu identifizieren soll der Suffix **.supa** (nur Kleinbuchstaben) im Dateinamen verwendet werden. Der gleiche Suffix wird für alle SUPA-Dateien, unabhängig vom Inhalt verwendet.

Das nächste Kapitel spezifiziert die zu verwendenden Standardspalten für die verschiedenen SUPA-Dateien. Im Kapitel 4 wird die Syntax der verwendeten Datentypen definiert.

## 3 SUPA-Dateitypen

In diesem Kapitel werden die zu verwendenden Standardspalten für verschiedene SUPA-Dateitypen definiert. Generell wird auf eine einheitliche Spaltenverwendung geachtet, so dass zum Beispiel eine Datei mit Kontoumsätzen auch zum Import von Kontoverbindungen verwendet werden kann.

In den folgenden Abschnitten werden nur die Spaltennamen aufgezählt. Die genaue Spezifikation der Datentypen befindet sich im nächsten Kapitel. In den folgenden Tabellen wird in der dritten Spalte jeweils durch ein **M** eine verpflichtende Spalte gekennzeichnet, durch ein **O** eine optionale Spalte, die auch weggelassen werden kann. Ein **C** steht für ein teilweise verpflichtendes Feld, gemäß Felddescription.

### 3.1 Kontoinformationen

Eine Datei mit Kontoinformationen enthält eine Liste mit Konten und optional dem letzten bekannten Kontostand.

Spaltenname	Typ		Beschreibung
Id	Max35Text	O	Interne ID welche ein Konto innerhalb der exportierenden Anwendung eindeutig identifiziert.
AcctCtry	CountryCode	O	Land des Kontos.
AcctIBAN	IBANIdentifier	O	IBAN des Kontos.
AcctNo	Max35Text	O	Bank spezifische, nationale Kontonummer.
AcctBIC	BICIdentifier	O	SWIFT BIC des Kontos.
AcctBankCode	Max35Text	O	Nationaler Bankcode (Bankleitzahl) des Kontos.
AcctCcy	ISOCurrency	O	Kontowährung.  Darf nur bei Wertpapierdepots weggelassen werden. Bei allen anderen Kontotypen ist die Angabe verpflichtend.
AcctNm	Max35Text	O	Kontobezeichnung.
AcctTpCd	Code	O	Kontotyp. Folgende Codes sind zugelassen:  <b>GIRO</b> - Kontokorrent- oder Girokonto <b>CASH</b> - Bargeld, Kasse (Cash) <b>PRTF</b> - Wertpapierdepot (Securities Portfolio) <b>DPST</b> - Festgeld (Fixed-Term Deposit) <b>CRDC</b> - Kreditkarte (Credit Card)

OwnrNm	Max70Text	O	Name des Kontoinhabers.
BalAmt	Amount	O	Betrag des gebuchten Kontostandes.
BalCdtDbtInd	Code	O	Vorzeichen des gebuchten Kontostandes. Folgende Belegung ist zugelassen:  <b>CRDT</b> - Haben (Credit) <b>DBIT</b> - Soll (Debit)
BalDt	ISODate	O	Datum des gebuchten Kontostandes.

## 3.2 Zahlungsaufträge

Das Format für SUPA-Zahlungsaufträge kann für SEPA-Überweisungen, SEPA-Lastschriften oder für Auslandsüberweisungen im DTAZV-Format verwendet werden.

Spaltenname	Typ		Beschreibung
Id	Max35Text	O	Optionale interne ID welche eine Zahlung innerhalb der exportierenden Anwendung eindeutig identifiziert. Wird beim Import ignoriert.
SvcLvl	Code	M	Zahlungsdienst. Folgende Belegung ist zugelassen:  <b>SEPA</b> - SEPA Zahlung <b>AZV</b> - DTAZV Zahlung
PmtMtd	Code	M	Art der Zahlung. Folgende Belegung ist zugelassen:  <b>TRF</b> - Überweisung <b>DD</b> - Lastschrift  Bei DTAZV-Zahlungen ist hier nur TRF zugelassen.
ReqdExctnDt	ISODate	C	Gewünschtes Ausführungsdatum. Ist dieses nicht belegt, so wird eine sofortige Ausführung gewünscht.  <b>SEPA:</b> Muss bei SEPA-Lastschriften mit dem gewünschten Einzugsdatum belegt werden.
Amt	Amount	M	Betrag.
AmtCcy	ISOCurrency	M	Währung.  <b>SEPA:</b> Bei SEPA-Zahlungen ist die einzige zugelassene Währung EUR.
EndToEndId	Max35Text	O	<b>SEPA:</b> Ende-zu-Ende-Referenz (EREF) bei SEPA-

			Zahlungen. Ist diese bei SEPA-Zahlungen nicht belegt, so wird "NOTPROVIDED" verwendet.
PmtInflId	Max35Text	O	<p><b>SEPA:</b> Möglichst eindeutige Identifizierung für einen SEPA-Sammelauftrag. Alle Einzelaufträge eines Sammlers haben die gleiche PmtInflId.</p> <p>Wichtig: Beim importieren einer SUPA-Datei werden alle Aufträge mit gleicher PmtInflId automatisch zu einem Sammler zusammengefasst. Ist die PmtInflId nicht belegt, so werden die Zahlungen als Einzelzahlungen importiert.</p>
MndtId	Max35Text	C	<p><b>SEPA:</b> Mandats-ID bei SEPA-Lastschriften.</p> <p>Muss bei SEPA-Lastschriften angegeben werden.</p>
MndtDtOfSgntr	ISODate	C	<p><b>SEPA:</b> Das Unterzeichnungsdatum des SEPA-Lastschriftmandates bei SEPA-Lastschriften.</p> <p>Muss bei SEPA-Lastschriften angegeben werden.</p>
MndtLclInstrm	Code	C	<p><b>SEPA:</b> Angabe zur Art der Lastschrift (Basis- oder Firmenlastschrift) bei SEPA-Lastschriften. Folgende Belegung ist zugelassen:</p> <p><b>CORE</b> - SEPA-Basislastschrift  <b>COR1</b> - SEPA-Basislastschrift D-1  <b>B2B</b> - SEPA-Firmenlastschrift</p> <p>Muss bei SEPA-Lastschriften angegeben werden.</p>
CdtrId	Max35Text	C	<p><b>SEPA:</b> Gläubiger-ID (CRED) oder „Creditor Identifier“ CI bei SEPA-Lastschriften.</p> <p>Muss bei SEPA-Lastschriften angegeben werden.</p>
SeqTp	Code	C	<p><b>SEPA:</b> Information zur Lastschriftfolge bei SEPA-Lastschriften. Folgende Belegung ist zugelassen:</p> <p><b>FRST</b> - Erstmalige Lastschrift  <b>RCUR</b> - Wiederholte Lastschrift  <b>FNAL</b> - Letztmalige Lastschrift  <b>OOFF</b> - Einmalige Lastschrift</p> <p>Muss bei SEPA-Lastschriften angegeben werden.</p>
RmtInf	Max140Text	O	Verwendungszweck. Darf keine Zeilenumbrüche enthalten. Zeichen die im Zeichensatz des Zahlungsauftrags nicht erlaubt sind, werden

			automatisch ersetzt oder entfernt.
PurpCd	Code	O	<p><b>SEPA:</b> Vier Zeichen Purpose Code bei SEPA-Zahlungen (jedoch nicht bei SEPA Lastschriften).</p> <p>Beispielhafte Werte sind:</p> <p><b>BONU, PENS, SALA</b> - Lohn-, Gehalts-, Rentengutschrift (53)</p> <p><b>CBFF</b> - Vermögenswirksame Leistungen (54)</p> <p><b>GOVT, SSBE, BENE</b> - Überweisung öffentlicher Kassen (56)</p> <p><b>CHAR</b> - Spende (69)</p>
CostBearer	Code	M	<p><b>AZV:</b> Entgeltregelung bei AZV-Zahlungen, die aussagt, wer (Auftraggeber oder Zahlungsempfänger / Zahlungspflichtiger oder beide) die Kosten für den Zahlungsauftrag übernimmt.</p> <p>Folgende Belegung ist zugelassen:</p> <p><b>OUR</b> - Auftraggeber</p> <p><b>SHA</b> - beide</p> <p><b>BEN</b> - Zahlungsempfänger / Zahlungspflichtiger</p> <p>Muss für eine sofortige Versendung des Auftrags belegt werden.</p>
OwnrNm	Max70Text	O	Name des Auftraggebers.
OwnrAcctCtry	CountryCode	O	Land des Auftraggebers.
OwnrAcctIBAN	IBANIdentifier	O	IBAN des Auftraggeberkontos.
OwnrAcctNo	Max35Text	O	Bank spezifische, nationale Kontonummer des Auftraggeberkontos.
OwnrAcctBIC	BICIdentifier	O	BIC der Bank des Auftraggeberkontos.
OwnrAcctBankCode	Max35Text	O	Nationaler Bank-Code (Bankleitzahl) des Auftraggeberkontos.
RmtdNm	Max70Text	M	Name des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAdrLine1	Max70Text	O	<b>AZV:</b> Bei AZV-Zahlungen die erste Adresszeile des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAdrLine2	Max70Text	O	<b>AZV:</b> Bei AZV-Zahlungen die zweite Adresszeile des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctCtry	CountryCode	M	Land des Zahlungsempfängers bzw.



			Zahlungspflichtigen.
RmtdAcctIBAN	IBANIdentifier	C	<p>IBAN des Zahlungsempfängers bzw. Zahlungspflichtigen.</p> <p><b>SEPA:</b> Verpflichtend für SEPA-Zahlungen.</p> <p><b>AZV:</b> Bei AZV-Zahlungen kann dieses Feld leer gelassen werden, wenn das Feld RmtdAcctNo mit einer Kontonummer befüllt ist.</p>
RmtdAcctNo	Max35Text	C	<p><b>AZV:</b> Bank spezifische, nationale Kontonummer des Zahlungsempfängers bzw. Zahlungspflichtigen. Dieses Feld kann leer gelassen werden, wenn das Feld RmtdAcctIBAN mit einer IBAN belegt ist.</p>
RmtdAcctBIC	BICIdentifier	C	<p>BIC der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.</p> <p><b>SEPA:</b> Sollte bei SEPA-Zahlungen bis zum 01.02.2016 unbedingt angegeben werden.</p> <p><b>AZV:</b> Bei AZV-Zahlungen kann dieses Feld leer gelassen werden, wenn das Feld RmtdAcctBankCode mit einem nationalen Bank-Code befüllt ist.</p>
RmtdAcctBankCode	Max35Text	C	<p><b>AZV:</b> Bei AZV-Zahlungen der nationale Bank-Code des Zahlungsempfängers bzw. Zahlungspflichtigen. Kann entfallen, wenn das Feld RmtdAcctBIC mit einer BIC belegt ist.</p>
RmtdAcctBankName	Max70Text	C	<p><b>AZV:</b> Name der Bank des Zahlungsempfängerkontos. Es muss entweder RmtdAcctBIC oder dieses Feld zusammen mit der Bankadresse in den Feldern RmtdAcctBankAdrLine1 und RmtdAcctBankAdrLine2 belegt werden.</p>
RmtdAcctBankAdrLine1	Max70Text	C	<p><b>AZV:</b> Bei AZV-Zahlungen die erste Adresszeile der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.</p>
RmtdAcctBankAdrLine2	Max70Text	C	<p><b>AZV:</b> Bei AZV-Zahlungen die zweite Adresszeile der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.</p>

### 3.3 Kontoumsätze

Eine Datei mit Kontoumsätzen enthält eine Liste mit Buchungsposten für ein oder mehrere Konten. Werden in einer Datei Buchungen für mehrere Konten exportiert, so muss über die Felder OwnrAcctXxx eine Kontozuordnung möglich sein. Werden ausschließlich Buchungen eines einzelnen, bekannten Kontos exportiert, so können diese Felder komplett entfallen.

Die Buchungen müssen je Konto entsprechend dem Buchungsdatum chronologisch sortiert sein. Werden Buchungen von mehreren Konten exportiert, so können die Buchungen der Konten abwechselnd vorkommen. Die Buchungen eines Kontos für sich betrachtet müssen aber dennoch chronologisch sortiert sein.

Ein Anfangssaldo, Endsaldo oder laufender Saldo wird nicht exportiert.

Spaltenname	Typ		Beschreibung
Id	Max35Text	O	Interne ID welche einen Buchungsposten innerhalb der exportierenden Anwendung eindeutig identifiziert.
OwnrAcctCcy	ISOCurrency	O	Kontowährung des Kontos zu dem diese Buchung gehört, wenn zur eindeutigen Identifizierung erforderlich.
OwnrAcctIBAN	IBANIdentifier	O	IBAN des Kontos zu dem diese Buchung gehört. Für eine Identifizierung des Kontos muss entweder das Feld OwnrAcctIBAN oder das Feld OwnrAcctNo belegt werden.
OwnrAcctNo	Max35Text	O	Bank spezifische, nationale Kontonummer des Kontos zu dem diese Buchung gehört. Für eine Identifizierung des Kontos muss entweder das Feld OwnrAcctIBAN oder das Feld OwnrAcctNo belegt werden.
OwnrAcctBIC	BICIdentifier	O	BIC der Bank des Kontos zu dem diese Buchung gehört. Für eine Identifizierung des Kontos muss entweder das Feld OwnrAcctBIC oder das Feld OwnrAcctBankCode belegt werden.
OwnrAcctBankCode	Max35Text	O	Nationaler Bank-Code (Bankleitzahl) des Kontos zu dem diese Buchung gehört. Für eine Identifizierung des Kontos muss entweder das Feld OwnrAcctBIC oder das Feld OwnrAcctBankCode belegt werden.
BookgDt	ISODate	M	Buchungsdatum.
ValDt	ISODate	M	Wertstellungsdatum.

Amt	Amount	M	Betrag.
AmtCcy	ISOCurrency	M	Währung.
CdtDbtInd	Code	M	Soll/Haben Kennzeichen. Folgende Belegung ist zugelassen:  <b>CRDT</b> - Haben (Credit) <b>DBIT</b> - Soll (Debit)
EndToEndId	Max35Text	O	Ende-zu-Ende-Referenz (EREF) bei SEPA-Zahlungen, Kundennummer bei DTA-Zahlungen.  Die Kundennummer aus dem DTA Feld C6b wird aus dem SWIFT MT-940 Feld 61 „Bankreferenz“ entnommen.
PmtInflId	Max35Text	O	Kundenreferenz (KREF) bei SEPA-Zahlungen, Referenznummer des Einreichers bei DTA-Zahlungen.  Die Kundenreferenz wird bei SEPA-Zahlungen aus dem Feld PmtInflId entnommen, welches einen Sammler innerhalb einer SEPA-Datei eindeutig identifiziert.  Die Referenznummer des Einreichers aus dem DTA Feld A10 wird aus dem SWIFT MT-940 Feld 61 „Referenz“ entnommen.  Die Kundenreferenz bei SEPA-Zahlungen wird aus dem SWIFT MT-940 Feld 86 Subfeld KREF bei strukturierter Belegung, entnommen.
MndtId	Max35Text	O	Mandats-ID (MREF) bei SEPA-Lastschriften.
CdtrId	Max35Text	O	Gläubiger-ID (CRED) oder „Creditor Identifier“ CI bei SEPA-Lastschriften.
RmtInf	Max140Text	O	Verwendungszweck. Darf keine Zeilenumbrüche enthalten. Ein ursprünglich mehrzeiliger Verwendungszweck wird durch Verkettung zu einer einzigen Zeile zusammengeführt.  Die maximale Länge gilt nur für SEPA-Zahlungen. Eine Anwendung darf hier bis zu 378 Zeichen exportieren. Beim Import wird das Datenfeld ggf. Abgeschnitten.
BookgTxt	Max35Text	O	Buchungstext.  Aus dem SWIFT MT-940 Feld 86, Subfeld 00.
PrimaNotaNo	Max35Text	O	Primanota.

			Aus dem SWIFT MT-940 Feld 86, Subfeld 10.
BankRef	Max35Text	O	Bankreferenz.  Aus dem SWIFT MT-940 Feld 61 subfeld 8 (Account Servicing Institution's Reference). Einige wenige Banken stellen im SWIFT Feld 61 eine eindeutige Buchungs-ID ein welche ideal zur Duplikatsprüfung verwendet werden kann. Leider machen das nur sehr wenige Banken, meist bleibt dieses Feld einfach leer.  Beim Import/Export von PayPal Transaktionen soll in diesem Feld die PayPal Transaktions-ID eingestellt werden.
BkTxCd	Code	O	Geschäftsvorfallscode  <b>NMSC</b> - Sonstige (Miscellaneous) <b>NTRF</b> - Überweisung (Transfer) <b>NDDT</b> - Lastschrift (Direct Debit) <b>NCLR</b> - Aus- /Einzahlung (Cash Letter) <b>NCHK</b> - Scheck (Check) <b>NSTO</b> - Dauerauftrag (Standing Order) <b>NRTI</b> - Storno (Returned Item)
RmtdNm	Max70Text	O	Name des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctCtry	CountryCode	O	Land des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctIBAN	IBANIdentifier	O	IBAN des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctNo	Max35Text	O	Bank spezifische, nationale Kontonummer des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBIC	BICIdentifier	O	BIC der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBankCode	Max35Text	O	Nationaler Bank-Code (Bankleitzahl) des Zahlungsempfängers bzw. Zahlungspflichtigen.
Category	Category	O	Anwendungsspezifische Kategorie welche dieser Buchung zugewiesen wurde. Ein Doppelpunkt wird als Trennzeichen verwendet um eine Kategorienhierarchie zu schaffen.
Notes	Max140Text	O	Vom Benutzer frei vergebene Notizen zu dieser

			Buchung.
--	--	--	----------

### 3.4 Kontoverbindungen

Eine Datei mit Kontoverbindungen enthält eine Liste mit Zahlungsempfängern beziehungsweise Zahlern.

Spaltenname	Typ		Beschreibung
Id	Max35Text	O	Interne ID welche eine Kontoverbindung innerhalb der exportierenden Anwendung eindeutig identifiziert.
RmtdNm	Max70Text	M	Name des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdNmMemo	Max70Text	O	Zusätzlicher Memotext zum Namen des Zahlungsempfängers bzw. Zahlungspflichtigen. Dieser wird in Aufträgen nicht verwendet, sondern dient lediglich zur Unterscheidung von Kontoverbindungen mit dem gleichen RmtdNm.
RmtdAdrLine1	Max70Text	O	Die erste Adresszeile des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAdrLine2	Max70Text	O	Die zweite Adresszeile des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctCtry	CountryCode	O	Land des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctIBAN	IBANIdentifier	O	IBAN des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctNo	Max35Text	O	Bank spezifische, nationale Kontonummer des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBIC	BICIdentifier	O	BIC der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBankCode	Max35Text	O	Nationaler Bank-Code (Bankleitzahl) des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBankName	Max70Text	O	Name der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.

RmtdAcctBankAdrLine1	Max70Text	O	Die erste Adresszeile der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtdAcctBankAdrLine2	Max70Text	O	Die zweite Adresszeile der Bank des Zahlungsempfängers bzw. Zahlungspflichtigen.
RmtInf	Max35Text	O	Kundennummer oder Kundenreferenz welche bei Zahlungen normalerweise übermittelt werden soll.
Category	Category	O	Anwendungsspezifische Kategorie welche allen neuen Buchungen zu dieser Kontoverbindung zugewiesen werden soll. Ein Doppelpunkt wird als Trennzeichen verwendet um eine Kategorienhierarchie zu schaffen.
MndtId	Max35Text	O	Die Mandatsreferenz des SEPA-Lastschriftmandats dieses Zahlungspflichtigen.
MndtDtOfSgntr	ISODate	O	Das Unterzeichnungsdatum des SEPA-Lastschriftmandats

## 4 Datentypen

Die Inhalte werden generell entsprechend den SEPA XML Formaten formatiert. Das heißt, Datumsangaben müssen im ISO-Format, zum Beispiel “2009-02-19” oder “2008-05-11T09:30:47.000Z” angegeben sein; Beträge mit einfachem Punkt als Dezimaltrenner, zum Beispiel “6543.14”.

Folgende Tabelle definiert die verwendeten Datentypen.

Name	Beschreibung
Max35Text Max70Text Max140Text	Text mit der im Namen enthaltenen maximalen Länge.
Amount	Betragsangabe ohne Vorzeichen mit einem optionalen, einfachen Punkt als Dezimaltrenner. Beispiele:  0.00 123456.78 0.1 123
ISODate	ISO Datumsangabe im Format JJJJ-MM-TT. Es sind nur reelle Datumsangaben erlaubt, d.h. Eine Datumsangabe wie der 30. Februar ist nicht zugelassen. Beispiele:  2009-02-20 9999-12-31 0000-01-01
CountryCode	ISO-Länderkennzeichen bestehend aus zwei Großbuchstaben. Zum Beispiel „DE“ für Deutschland.
IBANIdentifier	Internationale Kontonummer IBAN.
BICIdentifier	SWIFT BIC.
ISOCurrency	ISO-Währungskennzeichen bestehend aus drei Großbuchstaben. Zum Beispiel „EUR“ für Euro.
Code	Ein Codewort aus zwei bis vier Großbuchstaben. Die erlaubten Codes sind für die jeweilige Spalte angegeben.
Category	Kategorie. Ein Doppelpunkt wird als Trennzeichen verwendet um eine Kategorienhierarchie zu schaffen.

## 5 JavaScript-Schnittstelle

Die SUPA JavaScript-Schnittstelle ermöglicht den Zugriff auf Kontodaten und Kontoumsätze über eine JavaScript-Umgebung, wie etwa einen Web Browser oder Node.js. Die JavaScript-Schnittstelle ist gedacht für das Screen-Scraping von Online-Banking-Webseiten bei Banken, die keine Standard-Schnittstelle wie etwa HBCI oder FinTS unterstützen.

Eine einzelne JavaScript-Datei implementiert die Funktionen zur Anmeldung, dem Datenabruf und der Abmeldung vom Online-Banking einer bestimmten Bank. Diese JavaScript-Datei kann in Client-Anwendungen direkt eingebettet, oder über eine SSL gesicherte HTTPS-Verbindung dynamisch nachgeladen werden. Der eigentliche Abruf der Finanzdaten erfolgt dann durch den Aufruf von in dieser Spezifikation festgelegten JavaScript-Funktionen aus dem geladenen JavaScript.

Die Implementierung der JavaScript-Funktionen wird gezwungenermaßen oft asynchrone Aktionen ausführen müssen, wie zum Beispiel das Laden einer HTML-Seite über das XMLHttpRequest-Objekt. Aus diesem Grund sind alle hier spezifizierten JavaScript-Funktionen asynchron angelegt und liefern das eigentliche Ergebnis erst über eine Callback-Funktion welche beim Funktionsaufruf mitgegeben wird.

Die JavaScript-Schnittstelle ist sitzungsorientiert, d.h. eine Sitzung wird durch eine Anmeldung begonnen, gefolgt von mehreren Abfragen und abgeschlossen durch eine Abmeldung. Während der gesamten Sitzung bleibt die HTML-Seite mit dem vollständigen JavaScript-Kontext erhalten. In einer Sitzung können verschiedene Konten eines Nutzers abgerufen werden.

### 5.1 JSON

Die Finanzdaten werden generell als Strings im JSON-Format, also keine JavaScript-Objekte, übergeben. Innerhalb der übergebenen JSON-Objekte werden die einzelnen Datenfelder mit den SUPA Spaltennamen identifiziert. Alle Daten sind vom Typ "string". Der Inhalt des Strings muss jeweils den hier spezifizierten SUPA Datenformaten entsprechen.

Leider stellen nicht alle JavaScript-Umgebungen eine Implementierung der Funktionen `JSON.stringify` und `JSON.parse` zur Verfügung, so dass diese Funktionen in den Scripts **nicht** verwendet werden dürfen.



## 5.2 Funktion Login

Die Funktion Login muss zu Beginn zur Anmeldung des Kunden am System aufgerufen werden. Nach Abschluss der asynchron durchgeführten Anmeldung muss das JavaScript die übergebene Callback-Funktion OnLoginComplete aufrufen.

```
function Login(sUserID, sPIN, fnOnLoginComplete)
function OnLoginComplete(nResult, sErrorText)
```

Die Parameter sind wie folgt definiert:

Parameter	Beschreibung
sUserID	Die vom Benutzer angegebene Anmeldekennung. Je nach Bank kann dies eine Kontonummer, ein Benutzername oder eine Kreditkartennummer sein.
sPIN	Die vom Benutzer eingegebene PIN.
fnOnLoginComplete	Callback-Funktion welche vom Script nach Abschluss der asynchronen Anmeldung aufgerufen werden muss. Diese Funktion muss immer aufgerufen werden, auch wenn die Anmeldung fehlgeschlagen ist.
Parameter	Beschreibung
nResult	Fehlercode welcher anzeigt ob die Anmeldung erfolgreich war oder nicht. Folgende Werte sind definiert:  0: Anmeldung wurde erfolgreich abgeschlossen.  1: Bei der Anmeldung trat ein Fehler auf. Es wurde keine Sitzung eröffnet. Im Parameter sErrorText ist optional ein Fehlertext zu finden.  3: Die Anmeldung wurde vom Benutzer abgebrochen. Es wurde keine Sitzung eröffnet.
sErrorText	Bei einem Fehler kann hier optional ein Fehlertext für die Kundenanzeige mitgegeben werden.

## 5.3 Funktion Logout

Muss am Ende der Sitzung aufgerufen um den Kunden sauber vom System abzumelden.

```
function Logout(fnOnLogoutComplete)
function OnLogoutComplete()
```

Die Parameter sind wie folgt definiert:

Parameter	Beschreibung
fnOnLogoutComplete	Callback-Funktion welche vom Script nach Abschluss der asynchronen Abmeldung aufgerufen werden muss. Diese Funktion muss immer aufgerufen werden.

## 5.4 Funktion DownloadBalance

Abruf des aktuellen Kontostands.

```
function DownloadBalance(sJsonAcct, fnOnDownloadBalanceComplete)
function OnDownloadBalanceComplete(nResult, sJsonBalance)
```

Die Parameter sind wie folgt definiert:

Parameter	Beschreibung
sJsonAcct	<p>Ein JSON-String mit einem einzelnen, unbenannten JSON-Objekt mit Informationen zum Konto dessen Kontostand abgerufen werden soll. Folgende SUPA-Datenfelder können in diesem Objekt vorhanden sein:</p> <p><b>AcctNo</b>  <b>AcctBankCode</b>  <b>AcctCcy</b>  <b>AcctIBAN</b>  <b>AcctBIC</b></p> <p>Bei Kreditkarten wird die Kreditkartennummer im Feld AcctNo übergeben, alle anderen Felder werden nicht benötigt.</p>
fnOnDownloadBalanceComplete	<p>Callback-Funktion welche vom Script nach Abschluss des asynchronen Saldenbilanz aufgerufen werden muss. Diese Funktion muss immer aufgerufen werden, auch wenn der Abruf fehlgeschlagen ist.</p>
Parameter	Beschreibung
nResult	<p>Fehlercode welcher anzeigt ob der Abruf erfolgreich war oder nicht. Folgende Werte sind definiert:</p> <p>0: Der Saldo wurde erfolgreich abgerufen und liegt in sJsonBalance vor.</p> <p>1: Beim Abruf trat ein Fehler auf. Die mit Login eröffnete Sitzung ist weiterhin aktiv und es können weitere Abrufe ohne erneute Anmeldung durchgeführt werden.</p> <p>2: Beim Abruf trat ein schwerer Fehler auf, die mit Login eröffnete Sitzung kann nicht mehr fortgeführt werden, dennoch muss die Funktion Logout aufgerufen werden um die Sitzung endgültig abzuschließen.</p>

	3: Der Abruf wurde vom Benutzer abgebrochen und es wurden keine Daten geliefert. Im Anschluss sollte die Sitzung sofort durch den Aufruf von Logout beendet werden.
sJsonBalance	Falls der Abruf erfolgreich war, ein JSON-String mit einem einzelnen, unbenannten JSON Objekt mit dem aktuellen Kontostand. Folgende SUPA-Datenfelder müssen vom Script in diesem Objekt gesetzt werden:  <b>BalAmt</b> <b>BalCdtDbtInd</b> <b>BalDt</b>

## 5.5 Funktion DownloadStatement

Abruf der Kontoumsätze für einen bestimmten Zeitbereich.

```
function DownloadStatement(sJsonAcct, sFromDate, fnOnDownloadStatementComplete)
function OnDownloadStatementComplete(nResult, sJsonStatement)
```

Die Parameter sind wie folgt definiert:

Parameter	Beschreibung
sJsonAcct	<p>Ein JSON-String mit einem einzelnen, unbenannten JSON-Objekt mit Informationen zum Konto dessen Kontoumsätze abgerufen werden sollen. Folgende SUPA-Datenfelder können in diesem Objekt vorhanden sein:</p> <p><b>AcctNo</b>  <b>AcctBankCode</b>  <b>AcctCcy</b>  <b>AcctIBAN</b>  <b>AcctBIC</b></p> <p>Bei Kreditkarten wird die Kreditkartennummer im Feld AcctNo übergeben, alle anderen Felder werden nicht benötigt.</p>
sFromDate	<p>Optionales Startdatum ab dem die Umsätze geliefert werden sollen. Dieser Parameter wird als String im Format ISODate übergeben. Ist dieser Parameter null oder undefined, dann sollen alle vorhandenen Umsatzdaten für das Konto geliefert werden.</p>
Parameter	Beschreibung
nResult	<p>Fehlercode welcher anzeigt ob der Abruf erfolgreich war oder nicht. Folgende Werte sind definiert:</p> <p>0: Die Kontoumsätze wurden erfolgreich abgerufen und liegen in sJsonStatement vor. Liegen für das Konto im abgerufenen Zeitraum keine Umsätze vor und kann auch kein Buchungssaldo ermittelt werden, so wird in sJsonStatement ein leeres Objekt übergeben.</p> <p>1: Beim Abruf trat ein Fehler auf. Die mit Login eröffnete Sitzung ist weiterhin aktiv und es können weitere Abrufe ohne erneute Anmeldung durchgeführt werden.</p>

	<p>2: Beim Abruf trat ein schwerer Fehler auf, die mit Login eröffnete Sitzung kann nicht mehr fortgeführt werden, dennoch muss die Funktion Logout aufgerufen werden um die Sitzung endgültig abzuschließen.</p> <p>3: Der Abruf wurde vom Benutzer abgebrochen und es wurden keine Daten geliefert. Im Anschluss sollte die Sitzung sofort durch den Aufruf von Logout beendet werden.</p>
sJsonStatement	<p>Als Ergebnis wird ein JSON-String mit einem unbenannten Objekt geliefert. Dieses enthält wiederum optional ein <b>Bal</b> Objekt mit dem gebuchten Kontostand und , ebenfalls optional, ein <b>Stmt</b> Array mit den einzelnen Buchungen geliefert.</p> <p>Sofern ein gebuchter, aktueller Saldo ermittelt werden kann muss dieser als Objekt mit Namen <b>Bal</b> mit folgenden Feldern eingestellt werden:</p> <p style="margin-left: 40px;"><b>BalAmt</b> <b>BalCdtDbtInd</b> <b>BalDt</b></p> <p>Sofern Buchungen grundsätzlich abgerufen werden können, muss immer ein Array mit Namen <b>Stmt</b> im Ergebnis eingestellt werden. Sollten für den abgefragten Zeitraum keine Buchungen vorliegen, so ist dieses als leeres Array einzustellen.</p> <p>Die einzelnen Buchungen müssen im Array immer chronologischer Reihenfolge, beginnend bei der ältesten Buchung, eingestellt sein. Für jede Buchung muss im Array ein unbenanntes Objekt mit den SUPA Kontoumsatzdaten der Buchung eingestellt werden. Dabei müssen mindestens die oben im Kapitel "Kontoumsätze" als verpflichtend markierten Felder vorhanden sein.</p>

Das JSON-Objekt eines Umsatzabrufs kann beispielhaft wie folgt aussehen:

```
{
  "Bal" :
  {
    "BalAmt": "123.45",
    "BalCdtDbtInd": "CRDT",
    "BalDt": "2014-04-04"
  },
  "Stmt":
  [
```

```
{
  "BookgDt": "2014-04-04",
  "ValDt": "2014-04-04",
  "Amt": "1.45",
  "AmtCcy": "EUR",
  "CdtDbtInd": "CRDT",
  "RmtInf": "Test with Scraper Template"
},
{
  "BookgDt": "2014-04-05",
  "ValDt": "2014-04-04",
  "Amt": "2.78",
  "AmtCcy": "EUR",
  "CdtDbtInd": "DBIT",
  "EndToEndId": "Meine Kundenreferenz",
  "RmtInf": "Test with Scraper Template 2"
}
]
```