

kesh

Schnittstellenbeschreibung für Fremd-Apps V1.0.5

Stand:

20.10.2015

Inhaltsverzeichnis

1 Historie.....	4
2 Allgemeine Informationen zur Schnittstelle.....	4
2.1 Erläuterungen zur Beschreibung.....	4
2.2 Grundlegende Informationen.....	7
2.3 Kommunikation.....	7
2.4 Error-Codes.....	7
2.5 Datentypen.....	7
3 Wiederkehrende Datenstrukturen.....	8
3.1 Address.....	8
3.2 Amount.....	8
3.3 BankAccount.....	8
3.4 Contact.....	9
3.5 ContactInfo.....	10
3.6 Creditor.....	10
3.7 Group.....	10
3.8 GroupMember.....	11
3.9 ImageData.....	11
3.10 JobInfo.....	11
3.11 LimitUsage.....	11
3.12 PersonalData.....	12
3.13 Transaction.....	12
3.14 UserData.....	14
4 Requests & Responses.....	15
4.1 AuthorizationAnswer.....	15
4.2 Benutzerbild abrufen.....	17
4.3 Benutzerbild speichern.....	19
4.4 Cent Code verifizieren.....	21
4.5 Geld einfordern.....	22
4.6 Geld senden.....	25
4.7 Geld senden an nicht-kesh Nutzer.....	28
4.8 Geldeinforderung bestätigen.....	31
4.9 Gruppe anlegen.....	33
4.10 Gruppe ändern.....	35
4.11 Gruppe entfernen.....	37
4.12 Gruppenliste abfragen.....	38

4.13 Keep Alive.....	39
4.14 Kontakt entfernen.....	40
4.15 Kontakt hinzufügen.....	41
4.16 Kontaktliste abrufen.....	43
4.17 Konto aufladen.....	45
4.18 Konto entladen.....	47
4.19 Login.....	49
4.20 Logout.....	52
4.21 Nutzerdaten abrufen.....	53
4.22 Passwort ändern.....	55
4.23 Promocode senden.....	56
4.24 Saldoabfrage.....	57
4.25 SEPA Mandat abrufen.....	59
4.26 SEPA Mandatsvorschau abrufen.....	61
4.27 SEPA Mandat erteilen.....	63
4.28 Telefonnummer (SMS-Code) bestätigen.....	65
4.29 Telefonnummer ändern.....	67
4.30 Token senden.....	68
4.31 Transaktionsliste abfragen.....	70
5 Notifications.....	72
5.1 Allgemeine Information.....	72
5.2 Autorisierungsanfrage.....	73
5.3 Kontostand.....	75
5.4 Upgrade auf „kesh“-Kudentyp.....	76
5.5 Zahlungsinformation.....	77
6 Verwendung der Bibliothek.....	79
6.1 iOS.....	79
6.2 Android.....	83
6.3 Verbindungsdaten.....	86

1 Historie

Datum	Version	Mitarbeiter	Beschreibung
20.02.2014	1.0	Thomas Kläß	Initiale Version
11.08.2014	1.0	Alexander Nöske	Ergänzt um Verwaltungsfunktionen
17.10.2014	1.0	Alexander Nöske	Ergänzt um Installationsanleitung für iOS & Android
09.12.2014	1.0	Alexander Nöske	<ul style="list-style-type: none"> - Ergänzt um Abruf der Nutzerdaten, Transaktionsliste, Mandat & Mandatsvorschau, Erteilung des Mandats, Geld einfordern, Geldeinforderung bestätigen, Telefonnummer bestätigen und Konto entladen - Notifications: Kontostand, Zahlungsinformation - Transaction-Objekt ergänzt bei Responses zu Konto aufladen, Geld senden
12.01.2015	1.0.2	Alexander Nöske	<ul style="list-style-type: none"> - Bereich iOS-Implementierung ergänzt um Erklärung zu "connectionClosed"-Callback - Bereich Android-Implementierung ergänzt um Logging & Verbindungsstatus, Überarbeitung der Konfigurationsbeispiele
03.02.2015	1.0.3	Alexander Nöske	- Bereich iOS-Implementierung ergänzt um Erklärung zum autoReconnect
19.03.2015	1.0.4	Alexander Nöske	Methoden zur Bearbeitung von Webshop-Zahlungen ergänzt (Token senden, AuthorizationAnswer & Notification Autorisierungsanfrage)
20.10.15	1.0.5	Alexander Nöske	Methoden & Datentypen zur Bearbeitung von Gruppen & Kontakten ergänzt

2 Allgemeine Informationen zur Schnittstelle

2.1 Erläuterungen zur Beschreibung

2.1.1 Felddescriptionstabellen

Jedes Nachrichtenobjekt wird mit einer Tabelle beschrieben:

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
Feldname, wie er im JSON-Objekt angegeben werden muss.	Beispiel zur besseren Visualisierung, wie ein Wert aussehen könnte	Datentyp: String, Number, Array oder JSON	Falls nur bestimmte Werte zugelassen sind, werden diese hier aufgelistet	M steht für mandatory, also verpflichtend, O für optional, M/O steht für ein konditionales Pflichtfeld. Beispiel: body, oder errors müssen gefüllt sein	Beschreibung zum Feld

2.1.2 Parameter für Pflichtfelder

- (m)andatory
- (o)ptional
- (m/o)konditionales Pflichtfeld

2.1.3 Parameter für Status-Meldungen

- (s)uccess
- (w)arning
- (e)rror
- (f)ailure

HINWEIS: In neueren Version der Schnittstellenbeschreibung werden Änderungen wie folgt gekennzeichnet:

- Anpassungen sind gelb markiert
- Löschungen sind rot markiert
- Zu klärende Punkte sind blau markiert

2.1.4 Nachrichtenaufbau

Nachrichten bestehen aus folgendem Standardrahmen. Hier werden Metadaten zur Anfrage platziert. Anhand des Felds "request_type", wird unterschieden um welche Anliegen es sich handelt. Die tatsächlichen Inhalte finden sich im Feld "body".

```
"kesh_request"|"kesh_response"|"kesh_notification":{
  "<system_parameter_1>":"<wert_1>",
  ..
  "<system_parameter_n>":"<wert_n>"
  "body": {
    "<function_parameter_1>":"<wert_1>",
    ..
    "<function_parameter_n>":"<wert_n>"
  }
  "return_codes": {[
    "return_code": {
      "code":"00",
      "message":"Text"
    }
  ]}
}
```

Parameter	Verwendung	Beschreibung
request_art	--	Art des Requests <ul style="list-style-type: none"> • kesh_request: Anfrage an den Server, auf welchen der Server immer mit einem kesh_response antwortet • kesh_response: Antwort auf eine Anfrage (ist immer vom gleichen request_typ wie die Anfrage) • kesh_notification: requestunabhängigen Information des Servers an den Client bezüglich Ereignissen
interface_version	request response notification	Version der benutzten Funktion (wird bei Schnittstellenänderungen geändert)
request_type	request response notification	Bezeichnung der aufgerufenen Funktion
session_token	request response notification	eindeutiger sessionbezogener Identifier für den angemeldeten User; Wird beim Login generiert
request_id	response notification	<ul style="list-style-type: none"> • dient der Zuordnung des Response zum Request • ist durch das Client-System zu erzeugen und damit frei wählbar • erlaubt sind 0..9, a..z, A..Z mit max 20 Zeichen Länge • der Server wird die request_id immer im zugehörigen response wieder mitschicken • damit kann der Client die Zuordnung der Antworten zu den Requests vornehmen, wenn parallele Anfragen an den Server gestellt werden
status	response notification	<ul style="list-style-type: none"> • globaler Status bezüglich des Erfolgs der Funktion • Wertebereich "success warning error failure" • wenn Error, wird kein body-Bereich geliefert bzw. ist dessen Inhalt nicht benutzbar
body	request response notification	enthält die funktionsbezogenen Daten
return_codes	response	Array mit definierten Fehlermeldungen

2.2 Grundlegende Informationen

- die Funktionen der Schnittstelle werden versioniert, dies soll eine größere Flexibilität der Schnittstelle bei Änderungen ermöglichen
- Soweit möglich werden ältere Varianten einer Funktion mindestens über eine Schnittstellenversion hinweg unterstützt
- optionale Felder die "null"-Werte beinhalten sollen nicht gesendet werden

2.3 Kommunikation

- Die Kommunikation erfolgt im aktuellen Stand der Schnittstelle nur über socketIO.

2.4 Error-Codes

Bei allen in dieser Schnittstelle aufgeführten Requests können zusätzlich zu den im Request selbst spezifizierten Error-Codes auch die folgenden auftreten:

Error-Code	Beschreibung
99	System Error
501	JSON Validierungsfehler
502	Unsupported Request Type
901	Unsupported Interface Version

2.5 Datentypen

Typ	Darstellung	Beispiel	Anmerkungen
Datetime - Zeitstempel	Datumsstring mit folgender Formatierung: yyyy-MM-dd'T'HH:mm:ssz	"2014-07-16T19:47:18Z"	Standardmäßig wird die Zeitzone UTC (Zulu) verwendet.
Date - Einfach	Datumsstring mit folgender Formatierung: yyyy-MM-dd	"2014-07-16"	
Boolean	als Primitive	true	
Number	als Primitive	100.05	Javascript (damit auch JSON) Numbers sind immer 64bit Fließkommazahlen nach IEEE 754 Standard
String	als String	"6002317700"	Die Länge der String-Felder ist, falls nicht anders angegeben, auf 255 Zeichen begrenzt.

3 Wiederkehrende Datenstrukturen

In einigen Requests und Responses werden bestimmte Strukturen wiederverwendet. Diese sind an dieser Stelle aufgelistet.

3.1 Address

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
street	"Musterstr."	String		m	Straße
house_number	"12a"	String		m	Hausnummer
address_addon	"Zweiter Stock, links"	String		o	Zusatzinfo zur Adresse
post_code	"47877"	String		m	PLZ
city	"Willich"	String		m	Ort
country_code	"DE"	String		m	Land (ISO-Code)

3.2 Amount

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
value	99.34	Number		m	Wert
currency	"EUR"	String		m	Die Währung des Betrags, aktuell fix "EUR" (ISO 4217)

3.3 BankAccount

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
account_holder	"John Doe"	String		m	Kontoinhaber
bank_name	"biw"	String		m	Name der Bank
iban	"DE1234567892346386"	String		m	IBAN
bic	"BIWBDE33XXX"	String		o	BIC
mandate_reference	"2343j43g324"	String		o	Referenz des SEPA Mandates
mandate_valid_from	"2014-01-31"	String	yyyy-MM-dd	o	Gültigkeitsdatum des SEPA Mandates

3.4 Contact

Das Objekt "Contact" kann sowohl einen Händler, als auch einen Kunden repräsentieren. Das

kesh: Schnittstellenbeschreibung für Fremd-Apps V1.0.5

Feld `is_merchant` legt fest, ob es sich um einen Händler (true) oder einen Kunden (false) handelt.

Beim Händler ist `company_name` ein Pflichtfeld, beim Kunden `first_name` und `last_name`.

Je nach Einsatzzweck können aus Datenschutzgründen `first_name`, `last_name` und `phone_number` als Leerstring geliefert werden.

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
<code>first_name</code>	"John"	String		m/o	Vorname des Kontaktes
<code>last_name</code>	"Doe"	String		m/o	Nachname des Kontaktes
<code>user_id</code>	2	Number		m	User-Id des Kontaktes
<code>image_id</code>	35	Number		o	Bild-Id des Kontaktes
<code>kesh_type</code>	"P"	String	S, K, P	m	Kesh-Typ, zur Definition
<code>is_merchant</code>	false	Boolean		m	Gibt an, ob es sich um einen Händler handelt
<code>phone_number</code>	"015274298"	String		o	Telefonnummer des Kontaktes
<code>egdt_nr</code>	"60000024"	String		m	egdt_nr des Kontaktes, wird nach außen als "accountNumber" gegeben.
<code>is_on_contact_list</code>	true	Boolean		o	Gibt an, ob der Kontakt bereits in der Kontaktliste des Nutzers ist. Kann nicht "false" sein, wenn "is_favorite" gleichzeitig auf "true" steht
<code>is_favorite</code>	false	Boolean		o	Gibt an, ob der Kontakt bereits in der Kontaktliste des Nutzers ist und als Favorit gekennzeichnet wurde
<code>company_name</code>	"John Doe's company"	String		m/o	Ist der Kontakt ein Händler, steht hier der Firmenname. Muss gefüllt sein, wenn es ein Händler ist

3.5 ContactInfo

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
<code>email</code>	"john.doe@xcom.de"	String		m	eMail
<code>phone_number</code>	"015734375"	String		m	Telefonnummer

3.6 Creditor

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
creditor_id	"DE66KSH00000011391"	String		m	Gläubiger-Identifikationsnummer
creditor_name	"biw Bank für Investments und Wertpapiere AG"	String		m	Name des Gläubigers
address	{"street": "Musterstr.", "house_number": "12", "post_code": "47877", "city": "Willich", "country_code": "DE"}	JSON (Address)		m	Adresse des Gläubigers

3.7 Group

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
group_id	12	Number		m	ID der Gruppe
name	"Persönliche Kontakte"	String		m	Name der Gruppe, wird vom Kunden angegeben
image_id	35	Number		o	Bild-Id der Gruppe
group_members	[{"egdt_nr": "60000024", "name": "John Doe", "phone_number": "017577"}, ...]	Array (Group-Member)		m	Daten der Gruppenmitglieder mit kesh-Account

3.8 GroupMember

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
egdt_nr	"60000024"	String		m/o	ID des Users; nicht gesetzt, wenn kein kesh-User Braucht nicht gefüllt werden, wenn "phone_number" angegeben ist
name	"Walter Weiß"	String		o	Name des Users
phone_number	"0175777777"	String		m/o	Telefonnummer des Users Braucht nicht gefüllt werden, wenn "user_id" angegeben ist

3.9 ImageData

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
image	"28324gjk34"	String		m	Bild als BASE64-String codiert
mimetype	"image/jpg"	String	"image/jpg", "image/png", "image/gif", ...	m	Mimetype des Bildes

3.10 JobInfo

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
group	"biw"	String		m	Arbeitgeber
position	"Student"	String		m	Position
industry_sector	"Finanzen"	String		m	Sektor

3.11 LimitUsage

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflicht-feld	Kommentar
limit_type	"LASTSCHRIFT_TAG"	String	BAREINZAHLUNG_TAG, LASTSCHRIFT_MONAT, LASTSCHRIFT_TAG, PROMO_BAREINZAHLUNG_TAG, UMSATZ_MONAT_IN, UMSATZ_MONAT_OUT	m	Limit-Typ
current_amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Aktueller Betrag
max_amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Maximaler Betrag

3.12 PersonalData

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
academic_title	"Dr."	String		o	Akademischer Titel
first_name	"John"	String		m	Vorname des Kontaktes
first_names	"John Emil"	String		o	Vornamen des Kontaktes
last_name	"Doe"	String		m	Nachname des Kontaktes
gender	"m"	String		o	Geschlecht
birthdate	"1970-01-01"	String	yyyy-MM-dd	o	Geburtsdatum
birthplace	"Willich"	String		o	Geburtsort
birth_name	"Mustermann"	String		o	Geburtsname
marital_status	"ledig"	String		o	Familienstand
nationality	"DE"	String		o	Nationalität

3.13 Transaction

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	"6000035464"	String		m	E-Geld-Datenträgnr. / kesh-Konto-nummer des Transaktionspartners
internal_transaction_id	12	Number		m	Id der kesh-Transaktion
is_reversable	true	Boolean		m	Boolscher Wert, der angibt, ob die Transaktion noch nicht einem Tagesabschluss zugeordnet wurde und damit stornierbar ist.
transaction_date	"2014-09-28T19:03:12Z"	Datetime		m	Buchungszeitpunkt oder Erstellungszeitpunkt (created_at) bei Einforderungen, die noch nicht gebucht wurden.
amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Betrag der Transaktion
fee_amount	{"value":9.34, "currency":"EUR"}	JSON (Amount)		m	Betrag der Transaktion, den der angemeldete Nutzer zahlen muss.
status	"accepted"	String	accepted, declined, pending, reversed	m	Status der Transaktion.
type	"T"	String	T, IT, IF, OT, D, P, GI, GO, R, Q, W,	m	Typ der Transaktion.

			B, BP		
display_text	"Max Mustermann"	String		m	Name, Telefonnummer oder kesh-Kontonummer des Transaktionspartners, je nachdem zu welcher Anzeige der Nutzer berechtigt ist.
description	"Pizza für alle"	String		m	Verwendungszweck bzw. Beschreibung (z.B. bei Promo-Aktionen) der Transaktion. Bei iOS als „memo“ nach außen gegeben.
image_id	123	Number		m	Id des Kundenbildes/Logos des Transaktionspartners oder ein Systemicon (z.B. für eine Baraufladung). Die Systemicons werden über negative Zahlen vom Server identifiziert. Die Zahl -1 ist ausgenommen, da diese für andere Belange reserviert ist.
paypoint_id	12	Number		o	Id der beteiligten Zahlstelle
paypoint_name	„Kasse 1“	String		o	Name der beteiligten Zahlstelle

- Eingehende Transaktionen haben immer einen positiven Betrag, ausgehende Transaktionen einen negativen
- Die Bedeutung der "type"-Kürzel sind in nachfolgender Tabelle ersichtlich

Transaktionstypen

Feldname	Beispielwert
T	Transaktionen unter Kunden (peer2peer)
IT	Eingehende Zahlung vom Referenzkonto
IF	Eingehende Zahlung von externem Konto
OT	Ausgehende Zahlung zu externem Konto
D	Bareinzahlung über die Schnelllegitimierung (Quickregistration)
P	Kauf/Zahlung beim Händler
GI	Einzahlung am Geldautomat
GO	Auszahlung am Geldautomat
R	Storno
Q	Transaktionen im Zuge von Promotion-Aktionen
W	Webshop-Transaktionen
B	Baraufladung beim Händler/kesh-Point
BP	Geburtstagsgrüße

3.14 UserData

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	"6000024"	String		o	Egdt-Nr des Kontaktes, wird nach außen als "accountNumber" gegeben.
address	{ "street": "Musterstr.", "house_number": "12", "post_code": "47877", "city": "Willich", "country_code": "DE" }	JSON (Address)		o	Adressdaten des Kontaktes
bank_account	{ "account_holder": "John Doe", "bank_name": "biw", "iban": "DE1234567892346386" }	JSON (BankAccount)		o	Bankdaten des Kontaktes
contact_info	{ "email": "john.doe@xcom.de", "phone_number": "015734375" }	JSON (ContactInfo)		o	Kontaktdaten des Kontaktes
personal_data	{ "first_name": "John", "last_name": "Doe" }	JSON (PersonalData)		o	Persönliche Daten des Kontaktes
job_info	{ "group": "biw", "position": "Student", "industry_sector": "Finanzen" }	JSON (JobInfo)		o	Job Informationen zum Kontakt

4 Requests & Responses

4.1 AuthorizationAnswer

Eine zuvor erhaltene Autorisierungsanfrage (s. u.) wird mit diesem Request beantwortet. In der Antwort wird noch einmal das Token und der Anfragentyp mitgeschickt sowie der Antwortstatus.

- status *"accepted"*:
 - Bedeutung: Der Nutzer autorisiert kesh, die Aktion, die mit dem kesh-Token verbunden ist, durchzuführen.
 - Reaktion: Diese Autorisierung wird am Token vermerkt. Die darauf folgende Reaktion ist abhängig vom Typ der Aktion:
 - **webshop_payment**: Die Zahlung kann vom Webshop aus ausgeführt werden. Der Webshop wird entsprechend benachrichtigt.
 - **webshop_address_transfer**: Der kesh-Server überträgt die Adressdaten an den Webshop.
 - **webshop_transfer_and_pay**: Die Zahlung kann vom Webshop aus ausgeführt werden. Der kesh-Server überträgt die Adressdaten an den Webshop.
- status *"declined"*:
 - Bedeutung: Der Nutzer bricht/lehnt die Aktion ab.
 - Reaktion: Falls der Token bereits personalisiert ist, wird er verworfen. Der Webshop wird anschließend entsprechend informiert.

4.1.1 Request: authorization_answer

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
token	"08153434"	String		m	Aus z. B. dem QR-Code ausgelesener kesh-Token
status	"accepted"	String		m	siehe Methodenbeschreibung: "accepted declined"
type	"webshop_payment"	String		m	Definition siehe oben. Wird nochmal zum Abgleich im Server benutzt, ob der Token auch wirklich zum bestätigten Type gehört.

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "authorization_answer",
  "request_id": "1",
  "session_token": "login_response_session_token",
  "body": {
    "token": "08153434",
    "status": "accepted",
    "type": "webshop_payment"
  }
}
```

4.1.2 Response

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "authorization_answer",
  "request_id": "1",
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Aktion erfolgreich autorisiert"
      }
    }
  ]
}
```

4.1.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Aktion erfolgreich autorisiert / abgebrochen.
27	e	Übergebener Type stimmt nicht mit Type des Tokens überein
28	e	Token nicht (mehr) gültig

4.2 Benutzerbild abrufen

Abfrage des Benutzerbildes

4.2.1 Request: fetch_avatar

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
size	"full"	String		m	Bildgröße ("full", "medium", "thumb")
style	"square"	String		o	Bildstil ("square", "rounded"). Wird nichts angegeben, wird default das rechteckige Bild geliefert
image_id	5	Number		o	ID des Bildes. Wenn nicht angegeben, wird das Bild des aktuell angemeldeten Nutzers gesendet

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"fetch_avatar",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1",
  "body": {
    "image_id":5,
    "size":"full",
    "style":"square"
  }
}
```

4.2.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
image_data	{"image":"28324gjk34j3486238", "mimetype":"image/jpeg"}	JSON (ImageData)		m	Bilddaten
image_id	5	Number		m	ID unter der das Bild im System gespeichert ist

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "fetch_avatar",
  "request_id": "1",
  "body": {
    "image_data": { "image": "28324gjk34j3486238", "mimetype": "image/jpg" },
    "image_id": 45
  },
  "return_codes": [
    { "return_code": {
      "code": "00",
      "message": "Bild erfolgreich abgerufen"
    }
  }
}]
}
```

4.2.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Bild erfolgreich geladen
15	s	Bild konnte nicht geladen werden: Es sind sowohl egdt_nr als auch image_id definiert. oder Bild konnte nicht geladen werden.
16	s	Bild steht nicht öffentlich zur Verfügung (Kein Kontakt und kein Händler).

4.3 Benutzerbild speichern

Benutzerbild speichern

4.3.1 Request: save_avatar

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
image_data	{"image":"28324gjk34j3486238", "mimetype":"image/jpg"}	JSON (ImageData)		m	Bilddaten

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"save_avatar",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "image_data":{"image":"28324gjk34j3486238","mimetype":"image/jpg"}
  }
}
```

4.3.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
image_id	5	Number		m	ID unter der das Bild im System gespeichert ist

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"save_avatar",
  "request_id":"1",
  "body": {
    "image_id": 45
  },
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Bild erfolgreich gespeichert"
      }
    }
  ]
}
```

4.3.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Bild erfolgreich gespeichert
12	s	error saving avatar
13	s	Unbekannter MIME-Type

4.4 Cent Code verifizieren

Ein User bekommt eine 1-ct Überweisung auf das Referenzkonto. Im Verwendungszweck steht einen Aktivierungscode drin. Dieser Aktivierungscode muss in den Apps eingegeben werden und mit dem confirm_cent_transaction_request abgesetzt werden.

4.4.1 Request: confirm_cent_transaction

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
activation_code	"123456"	String		m	Der zu verifizierende Aktivierungscode

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"confirm_cent_transaction",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "activation_code":"123456"
  }
}
```

4.4.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"confirm_cent_transaction",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Konto erfolgreich aktiviert"
      }
    }
  ]
}
```

4.4.3 Status

Return-Code	Status	Message/Bedeutung
00	s	<keine Message definiert>
01	s	Es gibt kein Referenzkonto

4.5 Geld einfordern

Geld von einem anderen kesh-Nutzer einfordern. Die App-Library bietet hier 2 Methoden an; eine zum Einfordern von einer Kontonummer (egdt-nr), die andere zum Einfordern von einer Telefonnummer. Generell nicht für kesh-Starter möglich. Es muss entweder die kesh-Kontonummer oder die Handynummer des Ziels angegeben werden.

4.5.1 Request: request_payment

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	"12345678"	String		m/o	Das kesh-Konto, von welchem Geld eingefordert werden soll. Braucht nicht angegeben werden, wenn "phone_number" gesetzt ist. Wird nach außen als "accountNumber" gegeben.
amount	{"value":9.1, "currency":"EUR"}	JSON (Amount)		m	Der Betrag, der eingefordert werden soll.
phone_number	"01774743"	String		m/o	Die Telefonnummer, von der das Geld eingefordert werden soll. Braucht nicht gefüllt werden, wenn "egdt_nr" angegeben ist
description	"Kantine"	String		o	Überweisungstext

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"request_payment",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "device_hash":"hashed_identifizier",
  "body": {
    "egdt_nr":"6000023425",
    "amount":{"value":99.34,"currency":"EUR"},
    "description":"Überweisungstext Beispiel"
  }
}
```

4.5.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{ "egdt_nr": "6000046574", "internal_transaction_id": 22, "type": "T", "is_reversible": false, "transaction_date": "2014-09-28T19:03:12Z", "amount": {"value": 99.34, "currency": "EUR"}, "fee_amount": {"value": 0.09, "currency": "EUR"}, "status": "pending", "display_text": "Max Mustermann", "description": "1242", "image_id": 15, "paypoint_id": 12, "paypoint_name": "Kasse 2"} }	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "request_payment",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "T",
      "is_reversible": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": {
        "value": 99.34, "currency": "EUR"
      },
      "fee_amount": {
        "value": 0.09, "currency": "EUR"
      },
      "status": "pending",
      "display_text": "Max Mustermann",
      "description": "Pizza für alle",
      "image_id": 15,
      "paypoint_id": 12,
      "paypoint_name": "Kasse 2"
    }
  },
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Einforderung erfolgreich"
      }
    }
  ]
}
```

4.5.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Zahlung erfolgreich
10	e	Zahlung nicht möglich
11	e	Dieser Empfänger ist unbekannt.
12	e	Sender und Empfänger dürfen nicht identisch sein.
13	e	Minimaler Überweisungsbetrag unterschritten. (bei Händlern)
14	e	Um diese Aktion durchführen zu können, ist ein Upgrade nötig.
15	e	Sie oder Ihr Transaktionspartner haben den SMS Code noch nicht eingegeben.
16	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch sein Konto verifizieren.
17	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch ein Upgrade durchführen.
18	e	Die Funktion steht erst nach einer erfolgreichen Einzahlung von Ihrem Referenzkonto zur Verfügung.
19	e	Diese Aktion ist nicht erlaubt (Wenn keine Regel zutrifft)
20	e	Guthaben reicht nicht aus.
21	e	Monatliches Umsatzlimit ist erschöpft
30	e	Buchungsfehler (Grund wird als Text zurückgegeben)

4.6 Geld senden

Auslösen einer Zahlung.

Die App-Library bietet hier 2 Methoden an; eine zum Senden an eine Account-Number, die andere zum Senden an eine Phone-Number.

4.6.1 Request: send_money

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	"12345678"	String		m/o	Das kesh-Konto, auf welches das Geld überwiesen werden soll. Braucht nicht gefüllt werden, wenn "phone_number" angegeben ist. Wird nach außen als "accountNumber" gegeben.
amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Der Betrag, der gesendet werden sollte.
phone_number	"017773423439"	String		m/o	Die Telefonnummer, an die das Geld überwiesen werden soll. Braucht nicht gefüllt werden, wenn "egdt_nr" angegeben ist
description	"Mittagessen"	String		o	Überweisungstext
external_transaction_id	"ABC123"	String		o	Externe Transaktions-ID
external_picture_url	"http://biw.de/pic.jpg"	String		o	Link zu einem Bild, welches als Zahlungsempfänger für die Zahlung hinterlegt werden soll
paypoint_id	12	Number		o	Zahlstellen-ID

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"send_money",
  "session_token":"dsfgfdshzukturdyafdst4355zdhdhf",
  "request_id":"1"
  "body": {
    "egdt_nr":"6000023425",
    "amount":{"value":99.34, "currency":"EUR"}
    "description":"Mittagessen",
    "external_transaction_id":"ABC123",
    "external_picture_url":"http://test.de/picture.jpg",
    "paypoint_id":12
  }
}
```

4.6.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{ "egdt_nr": "6000046574", "internal_transaction_id": 22, "type": "T", "is_reversible": false, "transaction_date": "2014-09-28T19:03:12Z", "amount": {"value": 99.34, "currency": "EUR"}, "fee_amount": {"value": 0.09, "currency": "EUR"}, "status": "pending", "display_text": "Max Mustermann", "description": "1242", "image_id": 15, "paypoint_id": 12, "paypoint_name": "Kasse 2" }	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "send_money",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "T",
      "is_reversible": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": {"value": 99.34, "currency": "EUR"},
      "status": "pending",
      "display_text": "Max Mustermann",
      "description": "Pizza für alle",
      "image_id": 15
    }
  },
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Zahlung erfolgreich"
      }
    }
  ]
}
```

4.6.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Zahlung erfolgreich
10	e	Zahlung nicht möglich
11	e	Dieser Empfänger ist unbekannt.
12	e	Sender und Empfänger dürfen nicht identisch sein.
13	e	Minimaler Überweisungsbetrag unterschritten. (bei Händlern)
14	e	Um diese Aktion durchführen zu können, ist ein Upgrade nötig.
15	e	Sie oder Ihr Transaktionspartner haben den SMS Code noch nicht eingegeben.
16	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch sein Konto verifizieren.
17	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch ein Upgrade durchführen.
18	e	Die Funktion steht erst nach einer erfolgreichen Einzahlung von Ihrem Referenzkonto zur Verfügung.
19	e	Diese Aktion ist nicht erlaubt (Wenn keine Regel zutrifft)
20	e	Guthaben reicht nicht aus.
21	e	Monatliches Umsatzlimit ist erschöpft
22	e	Ungültige Zahlstelle
30	e	Buchungsfehler (Grund wird als Text zurückgegeben)

4.7 Geld senden an nicht-kesh Nutzer

Auslösen einer Zahlung an eine dem System unbekannte Telefonnummer.

4.7.1 Request: send_money_unknown_number

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Der Betrag, der gesendet werden sollte.
phone_number	"017773423439"	String		m/o	Die Telefonnummer, an die das Geld überwiesen werden soll.
description	"Mittagessen"	String		o	Überweisungstext
External_transaction_id	"ABC123"	String		o	Externe Transaktions-ID
external_picture_url	"http://biw.de/pic.jpg"	String		o	Link zu einem Bild, welches als Zahlungsempfänger für die Zahlung hinterlegt werden soll

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"send_money_unkown_number",
  "session_token":"dsfgfdshzuktrdgyafdstr4355zdhdhf",
  "request_id":"1"
  "body": {
    "phone_number":"01777344353",
    "amount":{"value":99.34, "currency":"EUR"}
    "description":"Mittagessen",
    "external_transaction_id":"ABC123",
    "external_picture_url":"http://test.de/picture.jpg"
  }
}
```

4.7.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{ "egdt_nr": "6000046574", "internal_transaction_id":22, "type":"T", "is_reversable":false, "transaction_date": "2014-09-28T19:03:12Z", "amount": {"value":99.34, "currency":"EUR"}, "fee_amount":{"value":0.09, "currency": "EUR"}, "status":"pending", "display_text":"Max Mustermann", "description":"1242", "image_id": 15}	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "send_money_unkown_number",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "T",
      "is_reversable": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": { "value": 99.34, "currency": "EUR" },
      "status": "pending",
      "display_text": "Max Mustermann",
      "description": "Pizza für alle",
      "image_id": 15
    }
  },
  "return_codes": [
    { "return_code": {
      "code": "00",
      "message": "Zahlung erfolgreich"
    }
  }
}]
}
```

4.7.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Zahlung erfolgreich
10	e	Zahlung nicht möglich
11	e	Dieser Empfänger ist unbekannt.
12	e	Sender und Empfänger dürfen nicht identisch sein.
13	e	Minimaler Überweisungsbetrag unterschritten. (bei Händlern)
14	e	Um diese Aktion durchführen zu können, ist ein Upgrade nötig.
15	e	Sie oder Ihr Transaktionspartner haben den SMS Code noch nicht eingegeben.
16	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch sein Konto verifizieren.
17	e	Eine Zahlung an den angegebenen Empfänger ist nicht möglich. Möglicherweise muss dieser noch ein Upgrade durchführen.
18	e	Die Funktion steht erst nach einer erfolgreichen Einzahlung von Ihrem Referenzkonto zur Verfügung.
19	e	Diese Aktion ist nicht erlaubt (Wenn keine Regel zutrifft)

kesh: Schnittstellenbeschreibung für Fremd-Apps V1.0.5

20	e	Guthaben reicht nicht aus.
21	e	Monatliches Umsatzlimit ist erschöpft
22	e	Ungültige Zahlstelle
30	e	Buchungsfehler (Grund wird als Text zurückgegeben)

4.8 Geldeinforderung bestätigen

Einforderung eines anderen kesh-Nutzers bestätigen oder ablehnen.

4.8.1 Request: confirm_payment

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
internal_transaction_id	123	Number		m	kesh-Transaktionsnummer für die Zahlung
status	"accepted"	String		m	"accepted" für Zahlung angenommen, "declined" für Zahlung abgelehnt

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "confirm_payment",
  "session_token": "dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id": "1",
  "device_hash": "hashed_identifizier",
  "body": {
    "internal_transaction_id": 123,
    "status": "accepted"
  }
}
```

4.8.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{ "egdt_nr": "6000046574", "internal_transaction_id": 22, "type": "T", "is_reversible": false, "transaction_date": "2014-09-28T19:03:12Z", "amount": { "value": 99.34, "currency": "EUR" }, "fee_amount": { "value": 0.09, "currency": "EUR" }, "status": "declined", "display_text": "Max Muster", "description": "1242", "image_id": 15 }	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "confirm_payment",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "T",
      "is_reversible": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": {
        "value": 99.34, "currency": "EUR"
      }
    },
    "status": "pending",
    "display_text": "Max Mustermann",
    "description": "Pizza für alle",
    "image_id": 15
  }
},
"return_codes": [
  {
    "return_code": {
      "code": "00",
      "message": "Einforderung erfolgreich bestätigt"
    }
  }
]
}
```

4.8.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Erfolgreich bestätigt/abgelehnt

4.9 Gruppe anlegen

Legt eine Gruppe an

4.9.1 Request: save_group

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
name	„Klassenfahrt“	String		m	Name der zu speichernden Gruppe
group_members	[{"egdt_nr": "600024", "name": "John Doe", "phone_number": "017577867896"}, ...]	Array (Group Member)		o	Teilnehmer der Gruppe
image_data	{"image": "28324gjk346238", "mimetype": "image/jpg"}	JSON (ImageData)		o	Bild der Gruppe

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "save_group",
  "session_token": "dsfgfdshzukturdyafdst4355zdhdh",
  "request_id": "1",
  "device_hash": "hashed_identifizier",
  "body": {
    "name": "Klassenfahrt",
    "group_members": [{"egdt_nr": "60000024", "name": "John Do", "phone_number": "01"}, ...],
    "image_data": {"image": "28324gjk34j3486238", "mimetype": "image/jpg"}
  }
}
```

4.9.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
group	{"group_id": 12, "name": "Klassenfahrt", "image_id": 1, "group_members": [{"egdt_nr": "60000024", "name": "John Do", "phone_number": "0175777777"}, ...]}	JSON (Group)		m	Gespeicherte Gruppe

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"save_group",
  "request_id":"1",
  "body": {
    "group":{
      "group_id": 12,
      "name": "Klassenfahrt",
      "image_id": 1,
      "group_members":[{"egdt_nr":"60000024","name":"John Do","phone_number": "0"},...]
    }
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Gruppe erfolgreich gespeichert."
    }}
  ]
}
```

4.9.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Gruppe erfolgreich gespeichert

4.10 Gruppe ändern

Ändert eine Gruppe. Hierfür muss zwingend die Id der zu ändernden Gruppe angegeben werden. Zusätzlich kann der neue Name, die neue Teilnehmerliste (vollständig!) und/oder ein neues Bild mitgegeben werden. Wird kein neuer Wert mitgegeben, wird der alte Wert beibehalten.

4.10.1 Request: change_group

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
group_id	12	Number		m	Id der zu ändernden Gruppe
name	„Klassenfahrt“	String		o	(neuer) Name der Gruppe
group_members	[{"egdt_nr": "600024", "name": "John Doe", "phone_number": "0175778678"}, ...]	Array (Group Member)		o	Teilnehmer der Gruppe
image_data	{"image": "28324gjk346238", "mimetype": "image/jpg"}	JSON (ImageData)		o	Bild der Gruppe

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "change_group",
  "session_token": "dsfgfdshzukturdyafdst4355zdhdh",
  "request_id": "1",
  "device_hash": "hashed_identifizier",
  "body": {
    "group_id": 12,
    "name": "Klassenfahrt Berlin"
  }
}
```

4.10.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
group	{"group_id": 12, "name": "Klassenfahrt Berlin", "image_id": 1, "group_members": [{"egdt_nr": "60000024", "name": "John Do", "phone_number": "0175777777"}, ...]}	JSON (Group)		m	Geänderte Gruppe

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "change_group",
  "request_id": "1",
  "body": {
    "group": {
      "group_id": 12,
      "name": "Klassenfahrt Berlin",
      "image_id": 1,
      "group_members": [{"egdt_nr": "60000024", "name": "John Do", "phone_number": "0"}, ..]
    }
  },
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Gruppe erfolgreich geändert."
      }
    }
  ]
}
```

4.10.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Gruppe erfolgreich geändert
01	e	Gruppe gehört einem anderen User

4.11 Gruppe entfernen

Entfernt eine Gruppe.

4.11.1 Request: remove_group

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
group_id	12	Number		m	Id der zu entfernenden Gruppe

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"remove_group",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1",
  "device_hash":"hashed_identifizier",
  "body": {
    "group_id": 12
  }
}
```

4.11.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"remove_group",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Gruppe erfolgreich entfernt."
      }
    }
  ]
}
```

4.11.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Gruppe erfolgreich entfernt
01	e	Gruppe unbekannt

4.12 Gruppenliste abfragen

Liefert eine Liste aller Gruppen des Benutzers zurück.

4.12.1 Request: list_groups

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"list_groups",
  "session_token":"dsfgfdshzukturdgyafdst4355zdhdh",
  "request_id":"1",
  "device_hash":"hashed_identifizier"
}
```

4.12.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
groups	[{"group_id": 12, "name": "Klassenfahrt Berlin", "image_id": 1, "group_members": [{"egdt_nr": "60000024", "name": "John Do", "phone_number": "0175777777"},...]}	Array (Group)		m	Liste der Gruppen des angemeldeten Users

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"list_groups",
  "request_id":"1",
  "body": {
    "groups":[{"group_id": 12,
      "name": "Klassenfahrt Berlin",
      "image_id": 1,
      "group_members":[{"egdt_nr":"60000024", "name": "John Do", "phone_number": "0"},...]}], ...]
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Gruppen erfolgreich ermittelt."
    }}
  ]
}
```

4.12.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Gruppen erfolgreich ermittelt

4.13 Keep Alive

Erneuert die aktuelle Session des Nutzers. In der Response wird das neue Session-Token übermittelt, dass für die folgenden Requests verwendet werden muss.

4.13.1 Request: keep_alive

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"keep_alive",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "device_hash":"hashed_identifizier"
}
```

4.13.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
session_token	"5gh239"	String		m	Identifiziert die Session und muss bei allen anderen Requests mitgegeben werden

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"keep_alive",
  "request_id":"1",
  "body": {
    "session_token":"5gh239"
  },
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Gruppe erfolgreich entfernt."
      }
    }
  ]
}
```

4.13.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Neue Session erstellt

4.14 Kontakt entfernen

Löscht einen Kontakt anhand der Egdt-Nr. aus der Kontaktliste des Benutzers.

4.14.1 Request: remove_contact

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	„6000023“	String		m	Egdt-Nr des Users welcher als Kontakt gelöscht werden soll (Account-Nr. des User-Datensatzes), wird nach außen als "accountNumber" gegeben.

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "remove_contact",
  "session_token": "dsfgfdshzuktrdgyafdstr4355zdhdh",
  "request_id": "1",
  "device_hash": "hashed_identifizier",
  "body": {
    "egdt_nr": "6000023"
  }
}
```

4.14.2 Response

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "remove_contact",
  "request_id": "1",
  "return_codes": [
    {
      "return_code": {
        "code": "00",
        "message": "Kontakt erfolgreich entfernt."
      }
    }
  ]
}
```

4.14.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Kontakt erfolgreich entfernt
01	e	Kontakt nicht in der Kontaktliste enthalten
02	e	Kontakt nicht gefunden
99	e	Es kam zu einem unbekannten Fehler beim Löschen des Kontaktes

4.15 Kontakt hinzufügen

Fügt einen Kontakt anhand der Egdt-Nr zur Kontaktliste des Benutzers hinzu. Dazu muss der Kontakt dem Hinzufügen zugestimmt haben. In der Response werden nun im Body die Kontaktinformationen direkt zurückgeliefert so dass sie in der App angezeigt werden können Contact-Objekt.

4.15.1 Request: add_contact

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	„6000023“	String		m	Egdt-Nr des Users welcher als Kontakt hinzugefügt werden soll (Account-Nr. des User-Datensatzes), wird nach außen als "accountNumber" gegeben.

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "add_contact",
  "session_token": "dsfgfdshzuktrdgyafdst4355zdhdhf",
  "request_id": "1",
  "device_hash": "hashed_identifizier",
  "body": {
    "egdt_nr": "6000023"
  }
}
```

4.15.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
contact	{"first_name": "John", "last_name": "Doe", "user_id": 2, "is_merchant": false, "egdt_nr": "60000024"}	String		m	Kontakt, der hinzugefügt wurde

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"add_contact",
  "request_id":"1",
  "body":{
    "contact":{"first_name": "John","last_name":"Doe", "user_id":2, "is_merchant":
      false,"egdt_nr":"60000024"}
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Kontakt erfolgreich hinzugefügt."
    }}
  ]
}
```

4.15.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Kontakt erfolgreich hinzugefügt
01	e	Kontakt nicht hinzugefügt, da dieser das Hinzufügen zu den Kontakten verboten hat.
02	e	Kontakt nicht gefunden
99	e	Es kam zu einem unbekannten Fehler beim Hinzufügen des Kontaktes

4.16 Kontaktliste abrufen

Liefert die Liste aller dem Nutzer zugeordneten Kontakte.

4.16.1 Request: add_contact

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
egdt_nr	„6000023“	String		m	Egdt-Nr des Users welcher als Kontakt hinzugefügt werden soll (Account-Nr. des User-Datensatzes), wird nach außen als "accountNumber" gegeben.

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"list_contacts",
  "session_token":"dsfgfdshzuktrdgyafdstr4355zdhdh",
  "request_id":"1",
  "device_hash":"hashed_identifizier"
}
```

4.16.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
contact	{"first_name":"John", "last_name":"Doe", "user_id":2, "is_merchant":false, "egdt_nr":"60000024"}	String		m	Kontakt, der hinzugefügt wurde

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"list_contacts",
  "request_id":"1",
  "body":{
    "contacts":[{"first_name": "John","last_name":"Doe", "user_id":2, "is_merchant":false,"egdt_nr":"60000024"}, ..]
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Kontakte erfolgreich ermittelt."
    }}
  ]
}
```

4.16.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Kontakt erfolgreich hinzugefügt

4.17 Konto aufladen

kesh-Konto um bestimmten Betrag vom Referenzkonto aufladen (nur kesh Basic & Premium)

4.17.1 Request: charge_account

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Betrag der Transaktion

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"charge_account",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "amount":{"value":99.34, "currency":"EUR"}
  }
}
```

4.17.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{"egdt_nr": "6000574", "internal_transaction_id": 22, "type":"IT", "is_reversable":false, "transaction_date":"2014-09-28T19:03:12Z", "amount":{"value":99.34, "currency":"EUR"}, "fee_amount":{"value":9.3, "currency":"EUR"}, "status": "pending", "display_text":"Max Must", "description":"1242", "image_id": 15}	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "charge_account",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "IT",
      "is_reversable": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": {
        "value": 99.34, "currency": "EUR"
      },
      "status": "pending",
      "display_text": "Max Muster",
      "description": "Pizza für alle",
      "image_id": 15
    }
  },
  "return_codes": [
    { "return_code": {
      "code": "00",
      "message": "Konto erfolgreich aufgeladen"
    }
  }
}]
}
```

4.17.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Konto erfolgreich aufgeladen
21	s	Transaktion nicht möglich.

4.18 Konto entladen

Bestimmten Betrag vom kesh-Konto aufs Referenzkonto entladen (nur kesh Basic & Premium)

4.18.1 Request: discharge_account

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
amount	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	Betrag der Transaktion

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"discharge_account",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "amount":{"value":99.34,"currency":"EUR"}
  }
}
```

4.18.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
transaction	{"egdt_nr": "6000574", "internal_transaction_id": 22, "type":"OT", "is_reversable":false, "transaction_date":"2014-09-28T19:03:12Z", "amount":{"value":99.34, "currency":"EUR"}, "fee_amount":{"value":9.3,"currency":"EUR"}, "status": "pending", "display_text":"Max Must", "description":"1242", "image_id": 15}	JSON (Transaction)		m	Daten der Transaktion

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "discharge_account",
  "request_id": "1",
  "body": {
    "transaction": {
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "OT",
      "is_reversible": false,
      "transaction_date": "2014-09-28T19:03:12Z",
      "amount": { "value": 99.34, "currency": "EUR" },
      "status": "pending",
      "display_text": "Max Mustermann",
      "description": "Pizza für alle",
      "image_id": 15
    }
  },
  "return_codes": [
    { "return_code": {
      "code": "00", "message": "Konto erfolgreich entladen"
    }
  }
}]
}
```

4.18.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Konto erfolgreich entladen
21	s	Transaktion nicht möglich.

4.19 Login

Anmeldung eines berechtigten Device (Handy oder Tablet) an der Schnittstelle

4.19.1 Request: login_user

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
phone_number	"01774567890"	String	[0-9]	m	Handynummer
password	"123456"	String	[0-9]	m	Passwort
device_hash	"hashed identifier"	String	[A-Za-z0-9]	m	Zum Beispiel: Hashed IMEI
app_version	"0.7.2c 15187"	String	[A-Za-z0-9._]	m	Version der kesh-App
app_type	"merchant"	String	"merchant" "customer" ...	m	Info für die APNS bzw. GCM Zuordnung
device_model	"Samsung Galaxy"	String	[A-Za-z0-9]"	m	Smartphone-Model
platform	"android"	String	"android" "ios"	m	Info für die APNS bzw. GCM Zuordnung
os_version	"1.0"	String		o	Betriebssystemversion
language	"de"	String	"de"-Deutsch, "en"-Englisch	m	Die Meldungen vom Server kommen in dieser Sprache

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"login_user",
  "request_id":"1",
  "body": {
    "phone_number":"01774567890",
    "password":"123456",
    "device_hash":"hashed identifier",
    "app_version":"0.7.2c 15187",
    "app_type":"merchant",
    "device_model":"Samsung Galaxy S",
    "os_version":"3.5",
    "platform":"android",
    "language":"de"
  }
}
```

4.19.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
session_token	"5gh239"	String		m	Identifiziert die Session und muss bei allen anderen Requests mitgegeben werden
user_id	1	String		m	User-Id
egdt_nr	"6000123"	String		m	kesh-Kontonummer, wird nach außen als "accountNumber" gegeben.
kesh_type	"p"	String		m	Kesh-Typ, zur Definition
account_blocked	false	Boolean		m	Angabe, ob das hinterlegende Konto gesperrt ist
is_merchant	false	Boolean		m	Angabe, ob es sich um ein Händler-Konto handelt
initial_pin	false	Boolean		m	Angabe, ob die es sich um eine Initial-Pin handelt, die nach dem Login geändert werden muss
acc_proofed	true	Boolean		m	Angabe, ob Referenzkonto bereits verifiziert ist
sms_confirmed	true	Boolean		m	Angabe, ob die Telefonnummer bereits verifiziert ist
agb_accepted	true	Boolean		m	Angabe, ob den AGB zugestimmt wurde
telephonebook_sync_accepted	false	Boolean		m	Angabe, ob den Bedingungen der Telefonbuchsynchronisation zugestimmt wurde
has_sepa_mandate	true	Boolean		m	Angabe, ob SEPA-Mandat erteilt wurde
one_ct_transferred	true	Boolean		m	Angabe, ob 1ct überwiesen wurde
must_confirm_new_phone_number	true	Boolean		m	Die Verifizierung einer neuen Telefonnummer steht noch aus.

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"login_user",
  "request_id":"1",
  "body":{
    "session_token":"login_response_session_token",
    "user_id": 3,
    "egdt_nr":"6000124344",
    "kesh_type": "S"|"K"|"P",
    "account_blocked":true|false,
    "is_merchant": true|false,
    "initial_pin": true|false,
    "acc_proofed":true|false,
    "must_confirm_new_phone_number":true|false,
    "agb_accepted":true|false,
    "telephonebook_sync_accepted":true|false,
    "has_sepa_mandate" : true|false,
    "one_ct_transferred" : true|false
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Anmeldung erfolgreich"
    }}
  ]
}
```

4.19.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Anmeldung erfolgreich
10	e	Handynummer/Pin falsch
11	e	Konto gesperrt
12	e	Du hast noch XX Fehlversuche bis dein kesh-Konto vorläufig gesperrt wird.
13	e	Anmeldungen von fremden Kunden sind nicht erlaubt.
14	e	Der Partner ist unerkannt
15	e	Deine App-Version wird nicht mehr unterstützt. Bitte lade dir die neue Version bei Google Play herunter.

4.20 Logout

Device (Handy oder Tablet) Abmeldung vom System

4.20.1 Request: logout_user

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"logout_user",
  "session_token":"login_response_session_token",
  "request_id":"1"
}
```

4.20.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"logout_user",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Abmelden erfolgreich"
      }
    }
  ]
}
```

4.20.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Abmelden erfolgreich

4.21 Nutzerdaten abrufen

Mit dem fetch_user_data-Request können die Stammdaten des angemeldeten Nutzers abgerufen werden.

4.21.1 Request: fetch_user_data

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"fetch_user_data",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1"
}
```

4.21.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
user_data	{ "egdt_nr":"6000024", "address":{ "street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE" }, "bank_account":{ "account_holder":"John Doe", "bank_name":"biw", "iban":"DE1234567892346386" }, "contact_info":{ "email":"john.doe@xcom.de", "phone_number":"015734375" }, "personal_data":{ "first_name":"John", "last_name":"Do" }, "job_info":{ "group":"bi", "position":"Student", "industry_sector":"Finanzen" } }	JSON (UserData)		m	Daten des angemeldeten Nutzers

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "fetch_user_data",
  "request_id": "1",
  "body": {
    "user_data": {
      "egdt_nr": "6000024",
      "address": { "street": "Musterstr.", "house_number": "12", "post_code": "47877", "city": "Willich", "country_code": "DE" },
      "bank_account": { "account_holder": "John Doe", "bank_name": "biw", "iban": "DE12636" },
      "contact_info": { "email": "john.doe@xcom.de", "phone_number": "015734375" },
      "personal_data": { "first_name": "John", "last_name": "Doe" },
      "job_info": { "group": "biw", "position": "Student", "industry_sector": "Finanzen" }
    },
    "return_codes": [
      { "return_code": {
        "code": "00",
        "message": ""
      }
    }
  ]
}
```

4.21.3 Status

Return-Code	Status	Message/Bedeutung
00	s	<keine Message definiert>

4.22 Passwort ändern

Benutzerpasswort ändern

4.22.1 Request: change_password

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
password_old	"123456"	String		m	Altes Passwort des Nutzers
password_new	"654321"	String		m	Neues Passwort des Nutzers

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"change_password",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "password_old":"123456",
    "password_new":"654321"
  }
}
```

4.22.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"change_password",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Passwort erfolgreich geändert"
      }
    }
  ]
}
```

4.22.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Passwort erfolgreich geändert
01	s	Passwort konnte nicht geändert werden
06	s	Altes Passwort ist nicht korrekt

4.23 Promocode senden

Ein Gutscheincode kann an den Server mittels dieser Methode übermittelt werden.
(Anwendungsfall: Werbeaktion).

4.23.1 Request: send_promo_code

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
promo_code	"123456"	String		m	Der Code, den der Kunde eingibt oder einscannt

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"send_promo_code",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdh",
  "request_id":"1",
  "body": {
    "promo_code":"123456"
  }
}
```

4.23.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"send_promo_code",
  "request_id":"1",
  "body":{
    "message":"Ihr Konto wurde erfolgreich mit 10,00 Euro aufgeladen"
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Werbecode erfolgreich eingelöst"
    }}
  ]
}
```

4.23.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Ihr Konto wurde erfolgreich mit <Betrag> aufgeladen.
01	s	Der eingegebene Werbecode ist ungültig oder abgelaufen.

4.24 Saldoabfrage

Abfrage des aktuellen Saldos

4.24.1 Request: account_balance

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"account_balance",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1"
}
```

4.24.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
balance	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuelles Guthaben
balance_locked	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuell gesperrtes Guthaben (= kann nur innerhalb des Systems ausgegeben werden)
Balance_ dechargable	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuell auf Referenzkonto entladbares Guthaben
limit_usages	{"limit_type": "BAREINZAHLUNG_TAG" , "current_amount": {"value":99.34, "currency":"EUR"}, "max_amount": {"value":99.34, "currency":"EUR"}}}	Array (LimitUsage)		m	Limits des Nutzers

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"account_balance",
  "request_id":"1",
  "body": {
    "balance":{"value":99.34,"currency":"EUR"},
    "balance_locked":{"value":99.34,"currency":"EUR"},
    "balance_dechargable":{"value":99.34,"currency":"EUR"},
    "limit_usages":[
      {"limit_type":"BAREINZAHLUNG_TAG",
        "current_amount":{"value":99.34, "currency":"EUR"},
        "max_amount":{"value":99.34, "currency":"EUR"}},
      ...
    ]
  },
  "return_codes": [
    {"return_code":{
      "code":"00",
      "message":"Limits erfolgreich abgerufen"
    }}
  ]
}
```

4.24.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Kontostand erfolgreich abgerufen

4.25 SEPA Mandat abrufen

Die Daten zu dem vom User erteilten SEPA-Mandat werden abgefragt.

4.25.1 Request: fetch_mandate

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"fetch_mandate",
  "session_token":"login_response_session_token",
  "request_id":"1",
  "device_hash":"jdjsfkljdkdsjfkld",
}
```

4.25.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
mandate_status	"SDD"	String		m	Status des Mandats.
mandate_type	"Hiermit ermächtige ich die biw AG, Zahlungen von..."	String		m	Der Typ des Mandats
user_data	{ "egdt_nr":"6000024", "address":{ "street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE" }, "bank_account":{ "account_holder":"John Do", "bank_name":"biw", "iban":"DE12386" }, "personal_data":{ "first_name":"John", "last_name":"Do" } }	JSON (UserData)		m	Benutzerdaten des Zahlungspflichtigen.
creditor	{ "creditor_id":"DE66KSH00000011391", "creditor_name":"biw Bank für Investments und Wertpapiere AG", "address":{ "street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE" } }	JSON (Creditor)		m	Informationen zum Zahlungsempfänger

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "fetch_mandate",
  "request_id": "1",
  "body": {
    "mandate_status": "0000104",
    "mandate_type": "SDD",
    "creditor": {
      "creditor_id": "DE66KSH00000011391",
      "creditor_name": "biw Bank für Investments und Wertpapiere AG",
      "address": {
        "street": "Hausbroicher Straße",
        "house_number": "222",
        "post_code": "47877",
        "city": "Willich",
        "country_code": "DE"
      },
    },
  },
  "user_data": {
    "egdt_nr": "6000024",
    "address": { "street": "Musterstr.", "house_number": "12", "post_code": "47877", "city": "Willich", "country_code": "DE" },
    "bank_account": { "account_holder": "John Doe", "bank_name": "biw", "iban": "DE123086", "mandate_reference": "2343j4", "mandate_valid_from": "01.01.2014" },
    "personal_data": { "first_name": "John", "last_name": "Doe" },
  },
},
"return_codes": [
  { "return_code": {
    "code": "00",
    "message": "Mandatsdaten erfolgreich abgerufen."
  }
}
]
```

4.25.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Mandatsdaten erfolgreich abgerufen.
01	s	Daten konnten nicht abgerufen werden

4.26 SEPA Mandatsvorschau abrufen

Daten für eine Mandatsvorschau abrufen

4.26.1 Request: fetch_mandate_preview

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"fetch_mandate_preview",
  "session_token":"login_response_session_token",
  "request_id":"1",
  "device_hash":"jdjsfkljdkdsjfkld"
}
```

4.26.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
mandate_type	"SDD"	String		m	Der Typ des Mandats
direct_debit_info	"Hiermit ermächtige ich die biw AG, Zahlungen von..."	String		m	Die Einzugsermächtigung die der Nutzer erteilt.
refund_time_period_info	"Ich kann innerhalb von 8 Wochen..."	String		m	Hinweistext unter der Vorschau. Informiert den Nutzer darüber, eine Lastschrift rückgängig machen zu können
terms_and_conditions_url	"http"	String		m	Link auf die AGBs
user_data	{"egdt_nr":"6024", "address":{"street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE"}, "bank_account":{"account_holder":"John Do", "bank_name":"biw", "iban":"DE192346386"}, "personal_data":{"first_name":"John", "last_name":"Do"}}	JSON (UserData)		m	Die Daten des Nutzers
creditor	{"creditor_id":"DE66KSH00000011391", "creditor_name":"biw Bank", "address":{"street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE"}}	JSON (Creditor)		m	Informationen zum Zahlungsempfänger

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"fetch_mandate",
  "request_id":"1",
  "body": {
    "direct_debit_info":"Hiermit ermächtige ich die biw AG, Zahlungen von...",
    "refund_time_period_info":"Ich kann innerhalb von 8 Wochen...",
    "mandate_type":"SDD",
    "creditor":{
      "creditor_id":"DE66KSH00000011391",
      "creditor_name":"biw Bank für Investments und Wertpapiere AG",
      "address":{
        "street":"Hausbroicher Straße",
        "house_number":"222",
        "post_code":"47877",
        "city":"Willich",
        "country_code":"DE"
      },
    },
    "user_data":{
      "egdt_nr":"6000024",
      "address":{
        "street":"Musterstr.", "house_number":"12", "post_code":"47877", "city":"Willich", "country_code":"DE"},
      "bank_account":{"account_holder":"John Doe", "bank_name":"biw", "iban":"DE1234567892346386"},
      "personal_data":{"first_name":"John", "last_name":"Doe"},
    },
    "return_codes": [
      {"return_code":{
        "code":"00",
        "message":"Mandatsvorschau erfolgreich abgerufen."
      }}
    ]
  }
}
```

4.26.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Mandatsvorschau erfolgreich abgerufen.
01	s	Daten konnten nicht abgerufen werden

4.27 SEPA Mandat erteilen

Request der App, wodurch ein neues Sepa-Mandat für den Kunden beantragt wird.

Dazu wird zunächst eine Preview auf den Gerät des Kunden angezeigt. Die entsprechenden Texte sind dazu in der App hinterlegt. / Die Texte für die Preview werden mit einem `create_mandate_preview` Request abrufen.

Nachdem der Kunde die Preview gesehen und dem Mandats-Antrag zu gestimmt hat, wird durch die App dieser Request gesendet.

4.27.1 Request: create_mandate

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"create_mandate",
  "session_token":"login_response_session_token",
  "request_id":"1",
  "device_hash":"jdjsfkljdkdsjfkld"
}
```

4.27.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
mandate_reference	"Referenz des aktuellen Mandats"	String		m	Referenz des Mandats.

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"create_mandate",
  "request_id":"1",
  "body": {
    "mandate_reference":"Referenz des aktuellen Mandats"
  },
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Mandat erfolgreich erteilt."
      }
    }
  ]
}
```

4.27.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Mandat erfolgreich erteilt.
01	e	Bankkonto nicht bestätigt.
02	e	Es existiert bereits ein Mandat.
03	e	Es existiert kein Referenzkonto.
04	e	Fehler bei der Erteilung des SEPA-Mandats.

4.28 Telefonnummer (SMS-Code) bestätigen

Bei der Registrierung wird bereits die Telefonnummer bestätigt. Sollte der Nutzer jedoch seine Handynummer einmal ändern wollen, so wird ihm in diesem Rahmen eine SMS mit einem Code zugeschickt. Diesen Code muss er dann in der App eingeben können.

4.28.1 Request: confirm_phone_number

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
phone_confirmation_code	"123456"	String		m	Der dem Nutzer zugesendete SMS-Code

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"confirm_phone_number",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1",
  "device_hash":"fdosifdoiivj",
  "body": {
    "phone_confirmation_code":"123456"
  }
}
```

4.28.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"confirm_phone_number",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Telefonnummer erfolgreich bestätigt."
      }
    }
  ]
}
```

4.28.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Telefonnummer erfolgreich bestätigt.
01	e	Code ist nicht korrekt. Noch <x> Versuche übrig.
02	e	Änderung der Telefonnummer nicht für Händler möglich
03	e	Der Code wurde zu oft falsch eingegeben. Bitte fordere eine neue SMS an.
04	e	Für den Kunden ist keine Telefonnummer bekannt, die verifiziert werden kann.
83	e	Es ist bereits ein Kunde unter der angegebenen Telefonnummer registriert.

4.29 Telefonnummer ändern

Benachrichtigt den Server über den Wunsch des Kunden, seine Telefonnummer zu ändern.

4.29.1 Request: change_phone_number

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
new_phone_number	"0178754873"	String		m	Neue Telefonnummer

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"change_phone_number",
  "session_token":"dsfgfdshzuktrdgyafdst4355zdhdhfh",
  "request_id":"1",
  "body": {
    "new_phone_number":"0178754873"
  }
}
```

4.29.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"change_phone_number",
  "request_id":"1",
  "return_codes": [
    {
      "return_code":{
        "code":"00",
        "message":"Telefonnummer als nicht verifiziert gespeichert. SMS versandt"
      }
    }
  ]
}
```

4.29.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Telefonnummer als nicht verifiziert gespeichert. SMS erfolgreich versandt
01	e	SMS nicht versandt. SMS Provider ist nicht erreichbar.
02	e	Änderung der Telefonnummer nicht für Händler möglich
81	e	Telefonnummer nicht gültig
82	e	Telefonnummer keine Deutschhandynummer
83	e	Es ist bereits ein Kunde unter der angegebenen Telefonnummer registriert

4.30 Token senden

Die App sendet das kundenneutrale Token (z. B. aus einem QR-Code ausgelesen) mittels dieser Methode an den kesh-Server. Die App trifft keinerlei Entscheidungen zum weiteren Ablauf. Sie geht z. B. wieder zurück auf den Home-Screen und wird über eine Notification über den weiteren Verlauf der Zahlung / gestarteten Aktion informiert. Damit werden kleinere Änderungen im Ablauf und in der Darstellung möglich, ohne ein Zwangsupdate der Apps durchführen zu müssen.

4.30.1 Request: send_token

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
token	"08153434"	String		m	Aus z. B. dem QR-Code ausgelesener kesh-Token

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"send_token",
  "request_id":"1"
  "session_token":"login_response_session_token"
  "body": {
    "token":"08153434"
  }
}
```

4.30.2 Response

Beispiel

```
{
  "interface_version":"2.0",
  "status":"success",
  "request_type":"send_token",
  "request_id":"1",
  "return_codes": [
    { "return_code":{
      "code":"00",
      "message":"Token erfolgreich gesendet"
    }
  }
}]
}
```

4.30.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Token erfolgreich gesendet
01	e	Webshop unbekannt
02	e	Autorisierungsanfrage konnte nicht verschickt werden.
21	e	Transaktion nicht möglich. Unzureichendes Guthaben.
26	e	Transaktion nicht möglich. Limit des kesh-Kontos wird überschritten.
28	e	Token nicht (mehr) gültig
30	e	Kundentyp darf Transaktion nicht durchführen

4.31 Transaktionsliste abfragen

Mit diesem Request lassen sich die Transaktionen des angemeldeten Nutzers abrufen.

- alle neuen/geänderten Transaktionen ab einem bestimmten Zeitpunkt mittels "last_update"
- oder für eine bestimmte Seite ("page"-Parameter). Die aktuelle Anzahl an Transaktionen pro Seite ist 30.

Zu jedem stornierten Umsatz gibt es eine dazugehörige Stornierung (txn_type R). Es werden beim Aufruf dieser Methode beide Transaktionen zurückgeliefert, d.h. stornierter Umsatz und Stornierung werden nicht zu einer Transaktion zusammengefasst.

4.31.1 Request: list_transactions

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
last_update	"2014-12-12T12:37:09.881Z"	DateTime		m/o	Zeitpunkt ab welchem die Transaktionen geladen werden sollen
page	2	Number		m/o	Standardmäßig werden die 30 letzten Transaktionen geliefert. Mit dem Parameter lassen sich satzweise ältere Transaktionen laden.

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"list_transactions",
  "session_token":"dsfgfdshzukturdyafdst4355zdhdh",
  "request_id":"1",
  "device_hash":"hashed_identifizier",
  "body":{
    "last_update": "2014-12-12T12:37:09.881Z"
  }
}
```

4.31.2 Response

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
last_update	"2014-10-28T14:21:28Z"	DateTime		o	Zeitpunkt ab welchem die Transaktionen geladen werden sollen
per_page	30	Number		m	Im Server konfigurierte Datensatzanzahl pro Seite
total_entries	310	Number		m	Gesamtanzahl der Transaktionen
current_page	2	Number		m	Aktuelle Seite
transactions		Array (Transaction)		m	Liste der Transaktionen

Beispiel

```
{
  "interface_version": "2.0",
  "status": "success",
  "request_type": "list_transactions",
  "request_id": "1",
  "body": {
    "current_page": 1,
    "per_page": 30,
    "last_update": "2014-10-21T14:25:22Z",
    "total_entries": 204,
    "transactions": [{
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type": "T",
      "is_reversable": false,
      "transaction_date": "2014-09-28T19: 03: 12Z",
      "amount": {
        "value": 99.34,
        "currency": "EUR"
      },
      "fee_amount": {
        "value": 9.34,
        "currency": "EUR"
      },
      "status": "pending",
      "display_text": "Max Mustermann",
      "description": "Pizza für alle",
      "type": "T",
      "image_id": 15
    }]
  },
  "return_codes": [{
    "return_code": {
      "code": "00",
      "message": "Transaktionsliste erfolgreich abgerufen"
    }
  }]
}
```

4.31.3 Status

Return-Code	Status	Message/Bedeutung
00	s	Transaktionsliste erfolgreich abgerufen

5 Notifications

5.1 Allgemeine Information

Kann an die App gesendet werden, wenn eine allgemeine Information angezeigt werden soll.

5.1.1 Notification: info_message

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
visualization	<code>{"title": "Autorisierung", "display_data": "<h2>....", "button_text": "Autorisieren"}</code>	JSON (Visualization)		m	Informationen für die App zur Darstellung der übermittelten Nutzdaten. „title“ enthält den anzuzeigenden Titel, „display_data“ ist ein html-String, welcher bereits fertig formatiert ist, „button_text“ legt den text für den anzuzeigenden Button fest.

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "info_message",
  "body": {
    "visualization": {
      "title": "Server-Downtime",
      "display_data": "Am <b>24.11.</b> wird von <b>10:00 bis 12:00 Uhr</b> ...",
      "button_text": "OK"
    }
  }
}
```


5.2 Autorisierungsanfrage

Diese Notification wird an den Kunde geschickt, wenn eine kesh-Aktion von ihm bestätigt werden muss.

Die darzustellende Oberfläche kann von der kesh-App anhand der in visualization übertragenen Informationen und der restlichen Nutzdaten aufgebaut werden. Sollte die Darstellung für bekannte Typen auf nativem Weg an Hand der übrigen Parameter erfolgen, so dienen die visualization-Daten als Fall-Back-Lösung. Diese View wird dann mit einem Titel, einem HTML-Feld und einem Bestätigen- sowie Abbrechen-Button aufgebaut. Ändert sich für bereits bekannte Typen die Darstellung, so wird die interface_version erhöht. In diesem Fall muss die App ebenfalls die Fallback-Lösung anzeigen. Neben den angegebenen Informationen wird stets ein "abbrechen"-Button eingeblendet.

Die Bestätigung/Ablehnung der Aktion erfolgt durch die authorization_answer (s. o.).

Beinhaltet die Notification eine Payment-Information, muss geprüft werden, ob Limite und Guthaben ausreichen, um die Aktion durchzuführen. Ansonsten sind entsprechende Meldungen anzuzeigen.

5.2.1 Notification: authorization_required_notification

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
type	"webshop_payment"	String		m	Typ der Forderung (z. B. "webshop_payment", "webshop_address_transfer", "webshop_transfer_and_pay")
token	"123456"	String		m	Token, das mit der Aktion verbunden ist
initiator	{ "image_id": 5, "initiator_name": "Fortuna Webshop", "egdt_nr": "6000123", "webshop_url": "www.f95.de", "shop_id": "aA23", "shop_customer_session": "werdd" }	JSON (Initiator)		m	kesh-Teilnehmer, der die Aktion initiiert hat (also quasi den Token erstellt hat)
payment	{ "value": 99.34, "currency": "EUR" }	JSON (Amount)		o	Zahlungsinformationen. Werden nur übertragen, wenn type eine Zahlung definiert
user_data	{ "egdt_nr": "624", "address": { "street": "Einstr.", "house_number": "12", "post_code": "47877", "city": "Willich", "country_code": "DE" }, "contact_info": { "email": "john.do@biw.de", "phone_number": "015" }, "personal_	JSON (User Data)		o	Daten des angemeldeten Nutzers, wird übermittelt, wenn type eine Übertragung der Nutzerdaten an den Initiator definiert. Wird nur partiell gefüllt, je nachdem, welche Informationen von der App

	data:{"first_name":"John", "last_name": "Doe"}				autorisiert werden sollen.
visualization	{"title":"Autorisierung", "display_data": "<h2>....", "button_text": "Autorisieren"}	JSON (Visual- ization)		m	Informationen für die App zur Darstellung der übermittelten Nutzdaten. Kann als Fallback-Lösung für unbekannte Typen angesehen werden. In diesem Fall erfolgt die Darstellung des display_text in einer Webview.

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "authorization_required_notification",
  "body": {
    "type": "webshop_transfer_and_pay",
    "token": "3434refs...",
    "initiator": {
      "image_id": 34,
      "initiator_name": "Fortuna Webshop",
      "egdt_nr": "...",
      "webshop_url": "super-shop.de",
      "shop_id": 23,
      "shop_customer_session": "wertfdd"
    },
    "payment": {"value": 99.34, "currency": "EUR"},
    "user_data": {
      "egdt_nr": "6000024",
      "address": {"street": "Musterstr.", "house_number": "12", "post_code": "47877",
        "city": "Willich", "country_code": "DE"},
      "bank_account": {"account_holder": "John Doe", "bank_name": "biw", "iban": "DE12386"},
      "contact_info": {"email": "john.doe@xcom.de", "phone_number": "015734375"},
      "personal_data": {"first_name": "John", "last_name": "Doe"},
      "job_info": {"group": "biw", "position": "Student", "industry_sector": "Finanzen"}},
      "visualization": {
        "title": "Zahlung",
        "display_data": "<h2>Bitte prüfe und bestätige die Angaben</h2><br/><b>...",
        "button_text": "Weiter im Shop"
      }
    }
  }
}
```

5.3 Kontostand

Nach jeder Zahlung, Zahlungsempfang, etc. wird der Kontostand gepusht.

5.3.1 Notification: account_balance

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
balance	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuelles Guthaben
balance_locked	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuell gesperrtes Guthaben (= kann nur innerhalb des Systems ausgegeben werden)
balance_dechargable	{"value":99.34, "currency":"EUR"}	JSON (Amount)		m	aktuell auf Referenzkonto entladbares Guthaben
limit_usages	{"limit_type": "LASTSCHRIFT_TAG", "current_amount": {"value":99.34, "currency":"EUR"}, "max_amount": {"value":99.34, "currency":"EUR"}}	Array (Limit Usage)		m	Limits des Nutzers

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"account_balance",
  "body": {
    "balance":{"value":99.34,"currency":"EUR"},
    "balance_locked":{"value":99.34,"currency":"EUR"},
    "balance_dechargable":{"value":99.34,"currency":"EUR"},
    "limit_usages":[
      {"limit_type":"BAREINZAHLUNG_TAG",
        "current_amount":{"value":99.34, "currency":"EUR"},
        "max_amount":{"value":99.34, "currency":"EUR"}},
      ...
    ]
  }
}
```

5.4 Upgrade auf „kesh“-Kundentyp

Upgrade des Kunden vom Kundentyp „starter“ auf Kundentyp „kesh“ erfolgreich.

5.4.1 Notification: user_upgraded

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
phone_number	"01584634"	String		o	Telefonnummer des Nutzers
egdt_nr	"6000024"	String		m	egdt_nr des Nutzers, wird nach außen als "accountNumber" gegeben.
kesh_type	"K"	String		m	Neuer Kesh-Typ des Nutzers

Beispiel

```
{
  "interface_version": "2.0",
  "request_type": "user_upgraded",
  "body": {
    "phone_number": "015274298",
    "egdt_nr": "60000024",
    "kesh_type": "K"
  }
}
```

5.5 Zahlungsinformation

- **Type: confirm_payment**

Wird von einem kesh-Kunden Geld eingefordert, wird diese Nachricht (Notification: confirm_payment) an dessen Gerät gepusht. Die App reagiert darauf mit der Anzeige eines Entscheidungsdialogs, wo die Forderung bestätigt, oder abgelehnt werden kann. Die Entscheidung wird mit dem "confirm_payment"-Request zum Server übermittelt.

- **Type: payment_changed**

Bei der Status-Änderung einer Transaktion wird diese an das Gerät gepusht.

- "pending" (offene Forderung) zu "accepted" (akzeptiert)
- "pending" zu "declined" (abgelehnt)

- **Type: payment_received**

Eine Notification mit diesem Typ wird geschickt bei

- eingehender Zahlung
- Promocode-Gutschrift
- Geldautomat Aufladung/Entladung
- externe Überweisung
- Überweisung vom Referenzkonto (per Überweisung)
- Geburtstagsgruß
- Storno-Umsatz

5.5.1 Notification: payment_info

Feldname	Beispielwert	Datentyp	Erlaubte Werte	Pflichtfeld	Kommentar
type	"payment_changed"	String	payment_changed, payment_received, confirm_payment	m	Hiermit wird festgelegt, ob es sich um eine beantwortete Einforderung oder Storno, eine eingegangene Zahlung, oder eine Zahlungseinforderung handelt.
transaction	{"egdt_nr": "6000046574", "internal_transaction_id": 22, "type": "T", "is_reversible": false, "transaction_date": "2014-09-28T19:03:12Z", "amount": {"value": 99.34, "currency": "EUR"}, "fee_ -	JSON (Transaction)		m	Zahlung

	<code>amount":{"value":9.34, "currency":"EUR"}, "status": "pending", "display_text": "Max Muster", "description": "Pizza für alle", "image_id": -15}, "paypoint_id":12, "paypoint_name":"Kasse 2"}</code>				
--	---	--	--	--	--

Beispiel

```
{
  "interface_version":"2.0",
  "request_type":"payment_info",
  "body":{
    "type":"confirm_payment",
    "transaction":{
      "egdt_nr": "6000046574",
      "internal_transaction_id": 22,
      "type":"T",
      "is_reversible": false,
      "transaction_date":"2014-09-28T19:03:12Z",
      "amount":{"value":99.34, "currency":"EUR"},
      "fee_amount":{"value":9.34, "currency":"EUR"},
      "status": "pending",
      "display_text":"Max Mustermann",
      "description":"Pizza für alle",
      "image_id": 15,
      "paypoint_id":12,
      "paypoint_name":"Kasse 2"
    }
  }
}
```

6 Verwendung der Bibliothek

6.1 iOS

Für iOS wurde ein eigenes Framework erstellt. In den folgenden Abschnitten wird erläutert, wie das Framework in Ihr Projekt einzubinden ist und wie die grundlegenden Funktionen der Library verwendet werden können.

6.1.1 Installation

1. Name des Frameworks: KSKeshLibraryExt.framework (im Demo-Projekt enthalten)
2. Legen Sie das Framework per Drag & Drop in der "Frameworks"-Gruppe ihres Projektes in Xcode ab. Achten Sie beim Import darauf, dass der Haken unter "Copy items into destination group's folder" gesetzt ist.
3. Fügen Sie der Build-Phase "Link Binary With Libraries folgende Frameworks hinzu:
 - SystemConfiguration.framework
 - UIKit.framework
 - CFNetwork.framework
 - Security.framework
 - Foundation.framework
4. Um die Klassen des Frameworks nutzen zu können, können Sie
 - Den Convenience-Header mittels `#import <KSKeshLibraryExt/KSKeshLibrary>` importieren.

- Den Header der jeweiligen Klasse einzeln importieren:

```
#import <KSKeshLibraryExt/KSAccountBalanceNotificationData.h>
#import <KSKeshLibraryExt/KSAccountBalanceResponseData.h>
#import <KSKeshLibraryExt/KSAuthorizationRequiredNotificationData.h>
#import <KSKeshLibraryExt/KSAddress.h>
#import <KSKeshLibraryExt/KSAmount.h>
#import <KSKeshLibraryExt/KSBankAccount.h>
#import <KSKeshLibraryExt/KSChargeAccountResponseData.h>
#import <KSKeshLibraryExt/KSCommunicatorDelegate.h>
#import <KSKeshLibraryExt/KSContactInfo.h>
#import <KSKeshLibraryExt/KSCreateMandateResponseData.h>
#import <KSKeshLibraryExt/KSCreditor.h>
#import <KSKeshLibraryExt/KSDataFormatter.h>
#import <KSKeshLibraryExt/KSDeviceIdentifier.h>
#import <KSKeshLibraryExt/KSDischargeAccountResponseData.h>
#import <KSKeshLibraryExt/KSFetchAvatarResponseData.h>
#import <KSKeshLibraryExt/KSFetchMandateResponseData.h>
#import <KSKeshLibraryExt/KSFetchMandatePreviewResponseData.h>
#import <KSKeshLibraryExt/KSFetchUserDataResponseData.h>
#import <KSKeshLibraryExt/KSHostConfiguration.h>
#import <KSKeshLibraryExt/KSImageData.h>
#import <KSKeshLibraryExt/KSJobInfo.h>
#import <KSKeshLibraryExt/KSLimitUsage.h>
#import <KSKeshLibraryExt/KSListTransactionsResponseData.h>
#import <KSKeshLibraryExt/KSLoginUserResponseData.h>
#import <KSKeshLibraryExt/KSManager.h>
#import <KSKeshLibraryExt/KSManagerDelegate.h>
#import <KSKeshLibraryExt/KSPaymentInfoNotificationData.h>
#import <KSKeshLibraryExt/KSPersonalData.h>
#import <KSKeshLibraryExt/KSRequestPaymentResponseData.h>
#import <KSKeshLibraryExt/KSResponseData.h>
#import <KSKeshLibraryExt/KSRetryTimeKeeper.h>
#import <KSKeshLibraryExt/KSSaveAvatarResponseData.h>
#import <KSKeshLibraryExt/KSSendMoneyResponseData.h>
#import <KSKeshLibraryExt/KSTransaction.h>
#import <KSKeshLibraryExt/KSUserData.h>
#import <KSKeshLibraryExt/KSUserDefaults.h>
```

5. Fügen Sie "-ObjC" in den Build Settings unter "Other Linker Flags" ein.
6. Fügen Sie ihr Zertifikat (.p12-Datei) dem per Drag & Drop Projekt hinzu. Achten Sie auch darauf, dass die Target Membership korrekt gesetzt wurde.
7. Fügen Sie ebenso das Server-Zertifikat (.der-Datei) hinzu.

6.1.2 Verbindungsaufbau & Abschicken von Requests

Um die Funktionen der kesh-Schnittstelle nutzen und Anfragen erfolgreich an den Server schicken zu können, muss zunächst die Verbindung zum kesh-Server hergestellt werden. Hierzu wird zunächst eine KSHostConfiguration für den entsprechenden Server (Produktion/Demo) erstellt. Beispiel:

```
KSHostConfiguration *hostConfiguration = [KSHostConfiguration
                                         hostConfigurationWithAddress:hostAddress
                                         port:port
                                         securedWithSSL:securedWithSSL
                                         serverCertPath:serverCertPathDER
                                         clientCertChainPath:certChainPathP12
                                         chainPassphrase:chainPassphrase];
```

Die Zertifikate werden über den NSBundle-Pfad des Zertifikats mitgegeben. Um im Beispiel-Projekt eine Verbindung aufbauen zu können, müssen daher zwingend die Zertifikats-Dateien dem Projekt hinzugefügt werden. Für die Zertifikatskette muss zudem das entsprechende Passwort ebenfalls übergeben werden.

Diese Konfiguration wird im Anschluss für den Verbindungsaufbau genutzt:

```
KSManager *mgr = [KSManager managerWithAppType:@"KeshDemo"
                                         appVersion:@"1.0.5"
                                         autoReconnect:NO];
[mgr setDelegate:self];
[mgr connectWithHostConfiguration:hostConfiguration];
```

Über die Delegate-Methoden wird das Delegate über den erfolgreichen Verbindungsaufbau (`connectionEstablished`), aber auch über Verbindungsabbrüche (`connectionLost` bei unbeabsichtigtem Verbindungsverlust, `connectionClosed` nach vorherigem Aufruf von `disconnect`) informiert.

Die Bibliothek besitzt einen konfigurierbaren Reconnect-Mechanismus. Sollte die Verbindung zum Server gestört oder unterbrochen werden, so wird im eingeschalteten Modus versucht, diese Verbindung wieder aufzubauen.

Der nächste Schritt ist der Login des Nutzers. Durch den Login erhält die kesh-Bibliothek ein Session-Token, welches dann automatisch für die weiteren Requests genutzt wird (andernfalls wird das Delegate über die Methode "needsAuthentication" über den nötigen Login informiert):

```
[mgr loginUserWithPhoneNumber:phoneNumber password:password
 onSuccessful:^(KSLoginUserData *data) {
     self didLogin:data.userDefaults;
 } onError:^(NSError *error) {
     [self loginFailedWithError:error];
 }
];
```

kesh: Schnittstellenbeschreibung für Fremd-Apps V1.0.5

Die Response-Daten der jeweiligen Requests können im obigen Dokument eingesehen werden. Bis auf die Methoden zum Aufbau und Beenden der Verbindung sind alle Methoden mit Success- und Error-Blöcken ausgestattet.

Eine Implementierung aller Requests ist im beigefügten Demo-Projekt ersichtlich (KSKeshDemo).

6.1.3 Notifications empfangen

Die o. g. Notifications werden durch die Library empfangen, verarbeitet und über das NSNotificationCenter verteilt. Die Anmeldung für den Empfang einer solchen Notification erfolgt über die addObserver-Methode, z. B.:

```
[[NSNotificationCenter defaultCenter] addObserver:self
                                     selector:@selector(accountBalanceNotificationReceived:)
                                     name:KSAccountBalanceNotification
                                     object:nil];
```

Die Abmeldung erfolgt dementsprechend über die removeObserver-Methode:

```
[[NSNotificationCenter defaultCenter] removeObserver:self
                                     name:KSAccountBalanceNotification
                                     object:nil];
```

Die an die jeweiligen Selektoren übergebenen Parameter vom Typ "NSNotification" enthalten innerhalb des userInfo-Dictionarys das jeweilige Datenobjekt zur Notification. Dieses kann über den entsprechenden Key ausgelesen werden, z. B.:

```
- (void)accountBalanceNotificationReceived:(NSNotification *)notification {
    KSAccountBalanceNotificationData *balanceNotification =
        [notification.userInfo objectForKey:KSAccountBalanceNotificationDataKey];
}
```

6.1.3.1 Aktuell verfügbare Notification-Namen & -Keys

Notification	name	object key
Allgemeine Information	KSInfoMessageNotification	KSInfoMessageNotificationDataKey
Autorisierungsanfrage	KSAccountBalanceNotification	KSAccountBalanceNotificationDataKey
Kontostand	KSAccountBalanceNotification	KSAccountBalanceNotificationDataKey
Nutzer Upgrade	KSUserUpgradedNotificationDataKey	KSUserUpgradedNotificationDataKey
Zahlungsinformation	KSPaymentInfoNotification	KSPaymentInfoNotificationDataKey

6.2 Android

Für Android wurde eine JAR-Paket erstellt. In den folgenden Abschnitten wird erläutert, wie das Paket in Ihr Projekt einzubinden ist und wie die grundlegenden Funktionen der Library verwendet werden können.

6.2.1 Installation

1. Name des JAR-Pakets: keshlib-[version].jar (im Demo-Projekt enthalten im „libs“-Verzeichnis)
2. Fügen Sie das JAR-Paket in den Projekt-Einstellungen dem Java Build Path (über den Reiter "Libraries") hinzu.
3. Die "JRE System Library" sollte ebenfalls im Projekt verlinkt sein.

6.2.2 Konfiguration

Um die Funktionen der kesh-Schnittstelle nutzen und Anfragen erfolgreich an den Server schicken zu können, muss zunächst der KeshServiceManager konfiguriert werden.

6.2.2.1 Logging

Der KeshServiceManager bietet einige statische Methoden an, über die sich das Loggingverhalten der Library konfigurieren lässt.

```
// Show all messages (Initial level is java.util.logging.Level.OFF)
KeshServiceManager.setLogLevel(Level.FINEST);

// Add a custom log handler (The library uses a java.util.logging.ConsoleHandler
initial)
MyCustomHandler myHandler = new MyCustomHandler();
KeshServiceManager.addLogHandler(myHandler);

// Remove a log handler
KeshServiceManager.removeLogHandler(myHandler);

// The default handler can also be replaced
Handler[] myHandlers = { new MyCustomHandler1(), new MyCustomHandler2() };
KeshServiceManager.setLogHandlers(myHandlers);
```

6.2.2.2 KeshServiceManagerConfiguration

Um die Funktionen der kesh-Schnittstelle nutzen und Anfragen erfolgreich an den Server schicken zu können, muss zunächst der KeshServiceManager konfiguriert werden.

Hierzu wird zunächst eine KeshServiceManagerConfiguration mit den entsprechenden

Appdaten erstellt. Beispiel:

```
String appVersion = "0.0.1"; // Version of your app
String appType = "myApp";    // String identifier for your app

KeshServiceManagerConfiguration conf = new
    KeshServiceManagerConfiguration(androidContext, appVersion, appType);

// By default, the library tries to reconnect itself after an error.
// Turn off auto reconnect ( Can also be done later with
// KeshServiceManager.getInstance().setAutoReconnectEnabled(false) )
conf.setAutoReconnectEnabled(false);

// Initialize KeshServiceManager
KeshServiceManager.initializeManager(conf);

// After initialization, you can get the instance
KeshServiceManager serviceManager = KeshServiceManager.getInstance();
```

6.2.3 Verbindungsaufbau

Einmal konfiguriert, kann anschließend der Verbindungsaufbau erfolgen:

```
private void connect() {
    X509Certificate cert = null;
    KeyStore store = loadClientKeyStore(); // Load a KeyStore with the client cert
    String storePasswd = "storepasswd";    // Password for the key store

    try {
        cert = loadCertificateFromFile(); // Load the server cert to identify the server
    } catch (Exception e) {
        e.printStackTrace();
    }
    // Connect using client authentication
    serviceManager.connectToServer(SERVER_URL, cert, store, storePasswd);
}
```

In der Methode `loadClientKeyStore()` wird ein `KeyStore` mittels der benötigten .p12-Datei erstellt, `loadCertificateFromFile()` lädt das X.509-Server-Zertifikat. Die Implementierung dieser Methoden ist im Demo-Projekt ersichtlich.

6.2.4 Abschicken von Requests

Der nächste Schritt ist der Login des Nutzers. Durch den Login erhält die kesh-Bibliothek ein Session-Token, welches dann automatisch für die weiteren Requests genutzt wird:

```
public void sendLogin(String phoneNumber, String password) {  
    serviceManager.sendLoginRequest(phoneNumber, password, callback);  
}
```

Über das Callback wird der Erfolg oder Misserfolg des Requests mitgeteilt, die Response-Daten der jeweiligen Requests können im obigen Dokument eingesehen werden. Aufbau des Callbacks:

```
private OnResponseReceivedCallback callback = new OnResponseReceivedCallback() {  
    public void onRequestFinishedSuccessful(AbstractResponseData arg0) {  
        System.out.println("Request finished successful");  
    }  
    public void onRequestFinishedWithError(AbstractErrorData arg0) {  
        System.out.println("Request finished with error");  
        ((ErrorDataV2) arg0).getMessage();  
    }  
};
```

Eine Implementierung aller Requests ist im beigefügten Demo-Eclipse-Projekt (KeshlibDemoApp) ersichtlich.

6.2.5 Notifications

Bei einigen Ereignissen schickt der Kesh-Service eine Nachricht an einen Client, ohne dass dieser eine Anfrage gesendet hat. Um diese Nachrichten verarbeiten zu können bietet der KeshServiceManager Methoden mit denen OnNotificationReceivedListener an- und abgemeldet werden können. Alle registrierten Listener werden über eingehende Notifications informiert.

```
public void registerForNotification() {  
    // Assume current class implements OnNotificationReceivedListener,  
    // serviceManager is our KeshServiceManager instance  
    serviceManager.registerNotificationListener(this);  
}  
  
public void unregisterForNotification() {  
    // Assume current class implements OnNotificationReceivedListener,  
    // serviceManager is our KeshServiceManager instance  
    serviceManager.unregisterNotificationListener(this);  
}
```

6.2.6 Verbindungsstatus

Die Library bietet die Möglichkeit einen Listener für Änderungen am Verbindungsstatus zu registrieren. Wie auch die `OnNotificationReceivedListener` können `OnConnectivityChangedListener` an- und abgemeldet werden. Alle registrierten Listener werden über Änderungen informiert.

```
public void registerForConnectionChanges() {
    // Assume current class implements OnConnectivityChangedListener,
    // serviceManager is our KeshServiceManager instance
    serviceManager.registerConnectionStateChangeListener(this);
}

public void unregisterForConnectionChanges() {
    // Assume current class implements OnConnectivityChangedListener,
    // serviceManager is our KeshServiceManager instance
    serviceManager.unregisterConnectionStateChangeListener(this);
}
```

6.3 Verbindungsdaten

In den nachfolgenden Abschnitten befinden sich die Verbindungsdaten zu den einzelnen Systemen.

6.3.1 Produktion

- Host: **spush.kesh.de**
- Port: **443**
- SSL-Verschlüsselung: **Ja**
- Server-Zertifikat: **Beim Support zu erfragen**

6.3.2 Demo

- Host: **demo1.kesh.de**
- Port: **743**
- SSL-Verschlüsselung: **Ja**
- Server-Zertifikat: **demo1.kesh.de.der** (iOS), **demo1.kesh.de.crt** (Android)