

---

# **python-fints Documentation**

***Release 4.0***

**Raphael Michel**

**Apr 22, 2023**



---

## Contents

---

<b>1</b>	<b>Library user documentation content</b>	<b>3</b>
1.1	Getting started . . . . .	3
1.2	Reading operations . . . . .	4
1.3	The client object . . . . .	7
1.4	Working with TANs . . . . .	9
1.5	Sending SEPA transfers . . . . .	16
1.6	Creating SEPA debits . . . . .	18
1.7	Tested banks . . . . .	20
1.8	Upgrading from python-fints 3.x to 4.x . . . . .	21
1.9	Upgrading from python-fints 2.x to 3.x . . . . .	21
1.10	Upgrading from python-fints 1.x to 2.x . . . . .	22
1.11	Troubleshooting and bug reporting . . . . .	22
<b>2</b>	<b>Library developer documentation content</b>	<b>27</b>
2.1	Developer documentation/API . . . . .	27
	<b>Python Module Index</b>	<b>113</b>
	<b>Index</b>	<b>115</b>



This is a pure-python implementation of FinTS (formerly known as HBCI), a online-banking protocol commonly supported by German banks.



## 1.1 Getting started

### 1.1.1 Register for a product ID

As of September 14th, 2019, all FinTS client programs need to be registered with the ZKA. You need to fill out a PDF form and will be assigned a product ID that you can pass to this library. It can take up to two weeks for the product ID to be assigned.

The reason for this requirement is compliance with the European Unions 2nd Payment Services Directive (PSD2) which mandates that end-users can transparently see which applications are accessing their bank account.

You can find more information as well as the registration form on the [ZKA Website](#) (only available in German).

### 1.1.2 Start coding

First of all, you need to install the library:

```
$ pip3 install fints
```

Then, you can initialize a FinTS client by providing your bank's BLZ, your username and PIN as well as the HBCI endpoint of your bank. Logging in with a signature file or chip card is currently not supported. For example:

```
import logging
from datetime import date
import getpass
from fints.client import FinTS3PinTanClient

logging.basicConfig(level=logging.DEBUG)
f = FinTS3PinTanClient(
    '123456789', # Your bank's BLZ
    'myusername', # Your login name
    getpass.getpass('PIN:'), # Your banking PIN
```

(continues on next page)

(continued from previous page)

```
'https://hbc-pintan.gad.de/cgi-bin/hbciservlet',
product_id='Your product ID' # see above
)
```

Since the implementation of PSD2, you will in almost all cases need to be ready to deal with TANs. For a quick start, we included a minimal command-line utility to help choose a TAN method:

```
from fints.utils import minimal_interactive_cli_bootstrap
minimal_interactive_cli_bootstrap(f)
```

You can then open up a real communication dialog to the bank with a `with` statement and issue commands: commands using the client instance:

```
with f:
    # Since PSD2, a TAN might be needed for dialog initialization. Let's check if_
    ↪there is one required
    if f.init_tan_response:
        print("A TAN is required", f.init_tan_response.challenge)
        tan = input('Please enter TAN:')
        f.send_tan(f.init_tan_response, tan)

    # Fetch accounts
    accounts = f.get_sepa_accounts()
```

Go on to the next pages to find out what commands are supported!

## 1.2 Reading operations

---

**Note:** Starting from version 3, **all of the methods on this page** can return a `NeedTANResponse` instead of actual data if your bank requires a TAN. You should then enter a TAN, read our chapter *Working with TANs* to find out more.

---

### 1.2.1 Fetching your bank accounts

The most simple method allows you to get all bank accounts that your user has access to:

```
class fints.client.FinTS3Client (bank_identifier, user_id, customer_id=None, from_data:
                                bytes = None, product_id=None, product_version='4.0.0',
                                mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

```
    get_sepa_accounts()
```

Returns a list of SEPA accounts

**Returns** List of SEPAAccount objects.

This method will return a list of named tuples of the following type:

```
class fints.models.SEPAAccount (iban, bic, accountnumber, subaccount, blz)
```

You will need this account object for many further operations to show which account you want to operate on.



## 1.2.2 Fetching bank information

During the first interaction with the bank some meta information about the bank and your user is transmitted from the bank.

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:
                               bytes = None, product_id=None, product_version='4.0.0',
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

**get\_information()**

Return information about the connected bank.

Note: Can only be filled after the first communication with the bank. If in doubt, use a construction like:

```
f = FinTS3Client(...)
with f:
    info = f.get_information()
```

Returns a nested dictionary:

```
bank:
  name: Bank Name
  supported_operations: dict(FinTSOperations -> boolean)
  supported_formats: dict(FinTSOperation -> [
    ↪ 'urn:iso:std:iso:20022:tech:xsd:pain.001.003.03', ...])
  supported_sepa_formats: ['urn:iso:std:iso:20022:tech:xsd:pain.001.003.03',
    ↪ ...]
accounts:
  - iban: IBAN
    account_number: Account Number
    subaccount_number: Sub-Account Number
    bank_identifier: fints.formals.BankIdentifier(...)
    customer_id: Customer ID
    type: Account type
    currency: Currency
    owner_name: ['Owner Name 1', 'Owner Name 2 (optional)']
    product_name: Account product name
    supported_operations: dict(FinTSOperations -> boolean)
  - ...
```

## 1.2.3 Fetching account balances

You can fetch the current balance of an account with the `get_balance` operation.

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:
                               bytes = None, product_id=None, product_version='4.0.0',
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

**get\_balance** (*account: fints.models.SEPAAccount*)

Fetches an accounts current balance.

**Parameters** **account** – SEPA account to fetch the balance

**Returns** A `mt940.models.Balance` object

This method will return a list of `Balance` objects from the `mt-940` library. You can find more information in [their documentation](#).

## 1.2.4 Reading account transactions

You can fetch the banking statement of an account within a certain timeframe with the `get_transactions` operation.

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:  
                               bytes = None, product_id=None, product_version='4.0.0',  
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

```
get_transactions(account: fints.models.SEPAAccount, start_date: datetime.date = None,  
                 end_date: datetime.date = None)
```

Fetches the list of transactions of a bank account in a certain timeframe.

### Parameters

- **account** – SEPA
- **start\_date** – First day to fetch
- **end\_date** – Last day to fetch

**Returns** A list of `mt940.models.Transaction` objects

```
get_transactions_xml(account: fints.models.SEPAAccount, start_date: datetime.date = None,  
                    end_date: datetime.date = None) → list
```

Fetches the list of transactions of a bank account in a certain timeframe as camt.052.001.02 XML files. Returns both booked and pending transactions.

### Parameters

- **account** – SEPA
- **start\_date** – First day to fetch
- **end\_date** – Last day to fetch

**Returns** Two lists of bytestrings containing XML documents, possibly empty: first one for booked transactions, second for pending transactions

This method will return a list of `Transaction` objects from the `mt-940` library. You can find more information in [their documentation](#).

## 1.2.5 Fetching holdings

You can fetch the holdings of an account with the `get_holdings` method:

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:  
                               bytes = None, product_id=None, product_version='4.0.0',  
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

```
get_holdings(account: fints.models.SEPAAccount)
```

Retrieve holdings of an account.

**Parameters** **account** – `SEPAAccount` to retrieve holdings for.

**Returns** List of `Holding` objects

This method will return a list of `Holding` objects:

```
class fints.models.Holding(ISIN, name, market_value, value_symbol, valuation_date, pieces,  
                          total_value, acquisitionprice)
```

## 1.3 The client object

### 1.3.1 Storing and restoring client state

The `FinTS3Client` object keeps some internal state that's beneficial to keep across invocations. This includes

- A system identifier that uniquely identifies this particular FinTS endpoint
- The Bank Parameter Data (BPD) with information about the bank and its advertised capabilities
- The User Parameter Data (UPD) with information about the user account and allowed actions

```
class fints.client.FinTS3Client (bank_identifier, user_id, customer_id=None, from_data:
                                bytes = None, product_id=None, product_version='4.0.0',
                                mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

**deconstruct** (*including\_private: bool = False*) → bytes

Return state of this `FinTSClient` instance as an opaque datablob. You should not use this object after calling this method.

Information about the connection is implicitly retrieved from the bank and cached in the `FinTSClient`. This includes: system identifier, bank parameter data, user parameter data. It's not strictly required to retain this information across sessions, but beneficial. If possible, an API user **SHOULD** use this method to serialize the client instance before destroying it, and provide the serialized data next time an instance is constructed.

Parameter *including\_private* should be set to `True`, if the storage is sufficiently secure (with regards to confidentiality) to include private data, specifically, account numbers and names. Most often this is the case.

Note: No connection information is stored in the datablob, neither is the PIN.

**set\_data** (*blob: bytes*)

Restore a datablob created with `deconstruct()`.

You should only call this method once, and only immediately after constructing the object and before calling any other method or functionality (e.g. `__enter__()`). For convenience, you can pass the *from\_data* parameter to `__init__()`.

Using the `deconstruct()/set_data()` facility is purely optional for reading operations, but may speed up the process because the BPD/UPD can be cached and need not be transmitted again.

It may be required to use the facility for transaction operations if both parts of a two-step transaction cannot be completed with the same `FinTS3Client` object.

The `deconstruct()` parameter *include\_private* (defaults to `False`) enables including the User Parameter Data in the datablob. Set this to `True` if you can sufficiently ensure the privacy of the returned datablob (mostly: user name and account numbers).

If your system manages multiple users/identity contexts, you **SHOULD** keep distinct datablobs per user or context.

You **SHOULD NOT** call any other methods on the `FinTS3Client` object after calling `deconstruct()`.

### 1.3.2 Keeping the dialog open

All FinTS operations happen in the context of a so-called “dialog”. The simple reading operations of this library will automatically open and close the dialog when necessary, but each opening and each closing takes one FinTS roundtrip.

For the case where multiple operations are to be performed one after the other you can indicate to the library that you want to open a standing dialog and keep it open explicitly by entering the `FinTS3Client` as a context handler.

This can, and should be, complemented with the client state facility as follows:

```
datablob = ... # get from backend storage, or set to None
client = FinTS3PinTanClient(..., from_data=datablob)

with client:
    accounts = client.get_sepa_accounts()
    balance = client.get_balance(accounts[0])
    transactions = client.get_transactions(accounts[0])

datablob = client.deconstruct()
# Store datablob to backend storage
```

For transactions involving TANs it may be required by the bank to issue both steps for one transaction within the same dialog. In this case it's mandatory to use a standing dialog, because otherwise each step would be issued in its own, implicit, dialog.

### 1.3.3 Storing and restoring dialog state

```
class fints.client.FinTS3Client (bank_identifier, user_id, customer_id=None, from_data:
                                bytes = None, product_id=None, product_version='4.0.0',
                                mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)
```

**pause\_dialog()**

Pause a standing dialog and return the saved dialog state.

Sometimes, for example in a web app, it's not possible to keep a context open during user input. In some cases, though, it's required to send a response within the same dialog that issued the original task (f.e. TAN with `TANTimeDialogAssociation.NOT_ALLOWED`). This method freezes the current standing dialog (started with `FinTS3Client.__enter__()`) and returns the frozen state.

Commands **MUST NOT** be issued in the dialog after calling this method.

**MUST** be used in conjunction with `deconstruct()/set_data()`.

Caller **SHOULD** ensure that the dialog is resumed (and properly ended) within a reasonable amount of time.

#### Example

```
client = FinTS3PinTanClient(..., from_data=None)
with client:
    challenge = client.sepa_transfer(...)

    dialog_data = client.pause_dialog()

    # dialog is now frozen, no new commands may be issued
    # exiting the context does not end the dialog

client_data = client.deconstruct()

# Store dialog_data and client_data out-of-band somewhere
# ... Some time passes ...
# Later, possibly in a different process, restore the state

client = FinTS3PinTanClient(..., from_data=client_data)
with client.resume_dialog(dialog_data):
    client.send_tan(...)
```

(continues on next page)

(continued from previous page)

```
# Exiting the context here ends the dialog, unless frozen with pause_
↪dialog() again.
```

```
resume_dialog(dialog_data)
```

## 1.4 Working with TANs

Many operations in FinTS will require a form of two-step authentication, called TANs. TANs are mostly required for operations that move money or change details of a bank account. TANs can be generated with a multitude of methods, including paper lists, smartcard readers, SMS messages, and smartphone apps.

### 1.4.1 TAN methods

Before doing any operations involving TANs, you should get a list of supported TAN mechanisms:

```
mechanisms = client.get_tan_mechanisms()
```

The returned dictionary maps identifiers (generally: three-digit numerals) to instances of a *TwoStepParametersCommon()* subclass with varying fields, depending on the version of the two-step process and the bank.

The *name* field of these objects provides a user-friendly name of the TAN mechanism that you can display to the user to choose from. To select a TAN mechanism, you can use *set\_tan\_mechanism()*, which takes the identifier used as key in the *get\_tan\_mechanisms()* return value.

If the *description\_required* attribute for the TAN mechanism is *MUST*, you will need to get a list of TAN media with *get\_tan\_media()* and select the appropriate one with *set\_tan\_medium()*.

Have a look at the source code of *minimal\_interactive\_cli\_bootstrap()* for an example on how to ask the user for these properties.

You may not change the active TAN mechanism or TAN medium within a standing dialog (see *Storing and restoring dialog state*).

The selection of the active TAN mechanism/medium is stored with the persistent client data (see *Storing and restoring client state*).

```
class fints.client.FinTS3PinTanClient(bank_identifier, user_id, pin, server, cus-
                                     tomer_id=None, *args, **kwargs)
```

```
get_current_tan_mechanism()
```

```
get_tan_mechanisms()
```

Get the available TAN mechanisms.

Note: Only checks for HITANS versions listed in *IMPLEMENTED\_HKTAN\_VERSIONS*.

**Returns** Dictionary of security\_function: TwoStepParameters objects.

```
get_tan_media(media_type=fints.formals.TANMediaType2.ALL, me-
               dia_class=fints.formals.TANMediaClass4.ALL)
```

Get information about TAN lists/generators.

Returns tuple of *fints.formals.TANUsageOption* and a list of *fints.formals.TANMedia4* or *fints.formals.TANMedia5* objects.

```
set_tan_mechanism (security_function)
```

```
set_tan_medium (tan_medium)
```

## 1.4.2 TAN challenges

When you try to perform an operation that requires a TAN to proceed, you will receive an object containing the bank's challenge (and some internal data to continue the operation once the TAN has been processed):

```
class fints.client.NeedTANResponse (command_seg, tan_request, resume_method=None,
                                     tan_request_structured=False)
```

```
challenge = None
```

Textual challenge to be displayed to the user

```
challenge_hhd_uc = None
```

HHD\_UC challenge to be transmitted to the TAN generator

```
challenge_html = None
```

HTML-safe challenge text, possibly with formatting

```
challenge_matrix = None
```

Matrix code challenge: tuple(mime\_type, data)

```
challenge_raw = None
```

Raw challenge as received by the bank

```
get_data () → bytes
```

Return a compressed datablob representing this object.

To restore the object, use `fints.client.NeedRetryResponse.from_data()`.

The `challenge` attribute will contain human-readable instructions on how to proceed.

The `challenge_html` attribute will possibly contain a nicer, formatted, HTML version of the challenge text that you should prefer if your primary interface can render HTML. The contents are guaranteed to be proper and clean (by using the *bleach* library): They can be used with *mark\_safe* in Django.

The `challenge_hhd_uc` attribute will contain the challenge to be used with a TAN generator device using the Hand Held Device Unidirectional Coupling specification (such as a Flicker-Code).

## 1.4.3 Flicker-Code / optiTAN

If you want to use chipTAN with an optical TAN device, we provide utilities to print the flicker code on a unix terminal. Just pass the `challenge_hhd_uc` value to this method:

```
fints.hhd.flicker.terminal_flicker_unix (code, field_width=3, space_width=3, height=1,
                                          clear=False, wait=0.05)
```

Re-encodes a flicker code and prints it on a unix terminal.

### Parameters

- **code** – Challenge value
- **field\_width** – Width of fields in characters (default: 3).
- **space\_width** – Width of spaces in characters (default: 3).
- **height** – Height of fields in characters (default: 1).
- **clear** – Clear terminal after every line (default: False).

- **wait** – Waiting interval between lines (default: 0.05).

You should probably catch for `KeyboardInterrupts` to allow the user to abort the displaying and to continue with the TAN:

```
try:
    terminal_flicker_unix(result.challenge_hhduc)
except KeyboardInterrupt:
    pass
```

### 1.4.4 photoTAN

If you want to use photoTAN, use the `challenge_matrix` attribute to access the image file, e.g. by writing it to a file:

```
with open("tan.png", "wb") as writer:
    writer.write(result.challenge_matrix[1])
writer.close()
```

### 1.4.5 Sending the TAN

Once obtained the TAN, you can send it with the `send_tan` client method:

```
class fints.client.FintS3PinTanClient (bank_identifier, user_id, pin, server, cus-
                                     tomer_id=None, *args, **kwargs)
```

**send\_tan** (*challenge: fints.client.NeedTANResponse, tan: str*)

Sends a TAN to confirm a pending operation.

#### Parameters

- **challenge** – NeedTANResponse to respond to
- **tan** – TAN value

**Returns** Currently no response

For example:

```
tan = input('Please enter the TAN code: ')
result = client.send_tan(result, tan)
```

### 1.4.6 Storing and restoring TAN state

The `get_data()` method and `from_data()` factory method can be used to store and restore a TAN state object between steps.

```
class fints.client.NeedRetryResponse
```

Base class for Responses that need the operation to be externally retried.

A concrete subclass of this class is returned, if an operation cannot be completed and needs a retry/completion. Typical (and only) example: Requiring a TAN to be provided.

```
classmethod from_data (blob)
```

Restore an object instance from a compressed datablob.

Returns an instance of a concrete subclass.

You SHOULD use this facility together with the client and dialog state restoration facilities:

Listing 1: First step

```
client = FinTS3PinTanClient(...)
# Optionally: choose a tan mechanism with
# client.set_tan_mechanism(...)

with client:
    response = client.sepa_transfer(...)

    dialog_data = client.pause_dialog()
client_data = client.deconstruct()
tan_data = response.get_data()
```

Listing 2: Second step

```
tan_request = NeedRetryResponse.from_data(tan_data)
print("TAN request: {}".format(tan_request.challenge))
tan = input('Enter TAN: ')
```

Listing 3: Third step

```
tan_request = NeedRetryResponse.from_data(tan_data)
client = FinTS3PinTanClient(..., from_data=client_data)
with client.resume_dialog(dialog_data):
    response = client.send_tan(tan_request, tan)

print(response.status)
print(response.responses)
```

## 1.4.7 Reference

**class** `fints.formals.TwoStepParameters2(*args, **kwargs)`

**name**

Name des Zwei-Schritt-Verfahrens

**Type** `str`

**max\_length\_input**

Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

**Type** `int`

**allowed\_format**

Erlaubtes Format im Zwei-Schritt-Verfahren

**Type** `fints.formals.AllowedFormat`

**text\_return\_value**

Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

**Type** `str`

**max\_length\_return\_value**

Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren



```

    Type int
number_of_supported_lists
    Anzahl unterstützter aktiver TAN-Listen

    Type int
multiple_tans_allowed
    Mehrfach-TAN erlaubt

    Type bool
tan_time_dialog_association
    TAN Zeit- und Dialogbezug

    Type fints.formals.TANTimeDialogAssociation
tan_list_number_required
    TAN-Listennummer erforderlich

    Type fints.formals.TANListNumberRequired
cancel_allowed
    Auftragsstorno erlaubt

    Type bool
challenge_class_required
    Challenge-Klasse erforderlich

    Type bool
challenge_value_required
    Challenge-Betrag erforderlich

    Type bool
VERSION
    TAN mechanism version

class fints.formals.TwoStepParameters3 (*args, **kwargs)

    name
    Name des Zwei-Schritt-Verfahrens

    Type str
max_length_input
    Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

    Type int
allowed_format
    Erlaubtes Format im Zwei-Schritt-Verfahren

    Type fints.formals.AllowedFormat
text_return_value
    Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

    Type str
max_length_return_value
    Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

    Type int

```

**number\_of\_supported\_lists**  
Anzahl unterstützter aktiver TAN-Listen

Type `int`

**multiple\_tans\_allowed**  
Mehrfach-TAN erlaubt

Type `bool`

**tan\_time\_dialog\_association**  
TAN Zeit- und Dialogbezug

Type `fints.formals.TANTimeDialogAssociation`

**tan\_list\_number\_required**  
TAN-Listennummer erforderlich

Type `fints.formals.TANListNumberRequired`

**cancel\_allowed**  
Auftragsstorno erlaubt

Type `bool`

**challenge\_class\_required**  
Challenge-Klasse erforderlich

Type `bool`

**challenge\_value\_required**  
Challenge-Betrag erforderlich

Type `bool`

**initialization\_mode**  
Initialisierungsmodus

Type `fints.formals.InitializationMode`

**description\_required**  
Bezeichnung des TAN-Medium erforderlich

Type `fints.formals.DescriptionRequired`

**supported\_media\_number**  
Anzahl unterstützter aktiver TAN-Medien

Type `int`

**VERSION**  
TAN mechanism version

**class** `fints.formals.TwoStepParameters5(*args, **kwargs)`

**zka\_id**  
ZKA TAN-Verfahren

Type `str`

**zka\_version**  
Version ZKA TAN-Verfahren

Type `str`

**name**  
Name des Zwei-Schritt-Verfahrens  
Type `str`

**max\_length\_input**  
Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren  
Type `int`

**allowed\_format**  
Erlaubtes Format im Zwei-Schritt-Verfahren  
Type `fints.formals.AllowedFormat`

**text\_return\_value**  
Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren  
Type `str`

**max\_length\_return\_value**  
Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren  
Type `int`

**number\_of\_supported\_lists**  
Anzahl unterstützter aktiver TAN-Listen  
Type `int`

**multiple\_tans\_allowed**  
Mehrfach-TAN erlaubt  
Type `bool`

**tan\_time\_dialog\_association**  
TAN Zeit- und Dialogbezug  
Type `fints.formals.TANTimeDialogAssociation`

**tan\_list\_number\_required**  
TAN-Listennummer erforderlich  
Type `fints.formals.TANListNumberRequired`

**cancel\_allowed**  
Auftragsstorno erlaubt  
Type `bool`

**sms\_charge\_account\_required**  
SMS-Abbuchungskonto erforderlich  
Type `fints.formals.SMSChargeAccountRequired`

**principal\_account\_required**  
Auftraggeberkonto erforderlich  
Type `fints.formals.PrincipalAccountRequired`

**challenge\_class\_required**  
Challenge-Klasse erforderlich  
Type `bool`

**challenge\_structured**  
Challenge strukturiert

**Type** `bool`

**initialization\_mode**  
Initialisierungsmodus

**Type** `fints.formals.InitializationMode`

**description\_required**  
Bezeichnung des TAN-Medium erforderlich

**Type** `fints.formals.DescriptionRequired`

**supported\_media\_number**  
Anzahl unterstützter aktiver TAN-Medien

**Type** `int`

**VERSION**  
TAN mechanism version

## 1.5 Sending SEPA transfers

### 1.5.1 Simple mode

You can create a simple SEPA transfer using this convenient client method:

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:  
                               bytes = None, product_id=None, product_version='4.0.0',  
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)  
  
    simple_sepa_transfer(account: fints.models.SEPAAccount, iban: str, bic: str, recipient_name:  
                        str, amount: decimal.Decimal, account_name: str, reason: str, in-  
                        stant_payment=False, endtoend_id='NOTPROVIDED')
```

Simple SEPA transfer.

#### Parameters

- **account** – SEPAAccount to start the transfer from.
- **iban** – Recipient's IBAN
- **bic** – Recipient's BIC
- **recipient\_name** – Recipient name
- **amount** – Amount as a `Decimal`
- **account\_name** – Sender account name
- **reason** – Transfer reason
- **instant\_payment** – Whether to use instant payment (defaults to `False`)
- **endtoend\_id** – End-to-end-Id (defaults to `NOTPROVIDED`)

**Returns** Returns either a `NeedRetryResponse` or `TransactionResponse`

You should then enter a TAN, read our chapter [Working with TANs](#) to find out more.

### 1.5.2 Advanced mode

If you want to use advanced methods, you can supply your own SEPA XML:

```
class fints.client.FinTS3Client(bank_identifier, user_id, customer_id=None, from_data:
                               bytes = None, product_id=None, product_version='4.0.0',
                               mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)

    sepa_transfer(account: fints.models.SEPAAccount, pain_message: str, multi-
                  ple=False, control_sum=None, currency='EUR', book_as_single=False,
                  pain_descriptor='urn:iso:std:iso:20022:tech:xsd:pain.001.001.03', in-
                  stant_payment=False)
    Custom SEPA transfer.
```

#### Parameters

- **account** – SEPAAccount to send the transfer from.
- **pain\_message** – SEPA PAIN message containing the transfer details.
- **multiple** – Whether this message contains multiple transfers.
- **control\_sum** – Sum of all transfers (required if there are multiple)
- **currency** – Transfer currency
- **book\_as\_single** – Kindly ask the bank to put multiple transactions as separate lines on the bank statement (defaults to `False`)
- **pain\_descriptor** – URN of the PAIN message schema used.
- **instant\_payment** – Whether this is an instant transfer (defaults to `False`)

**Returns** Returns either a `NeedRetryResponse` or `TransactionResponse`

### 1.5.3 Full example

```
client = FinTS3PinTanClient(...)
minimal_interactive_cli_bootstrap(client)

with client:
    if client.init_tan_response:
        print("A TAN is required", client.init_tan_response.challenge)

        if getattr(client.init_tan_response, 'challenge_hhduc', None):
            try:
                terminal_flicker_unix(client.init_tan_response.challenge_hhduc)
            except KeyboardInterrupt:
                pass

        tan = input('Please enter TAN:')
        client.send_tan(client.init_tan_response, tan)

res = client.simple_sepa_transfer(
    account=accounts[0],
    iban='DE12345',
    bic='BIC12345',
    amount=Decimal('7.00'),
    recipient_name='Foo',
    account_name='Test',
```

(continues on next page)

(continued from previous page)

```

        reason='Birthday gift',
        endtoend_id='NOTPROVIDED',
    )

    if isinstance(res, NeedTANResponse):
        print("A TAN is required", res.challenge)

        if getattr(res, 'challenge_hhduc', None):
            try:
                terminal_flicker_unix(res.challenge_hhduc)
            except KeyboardInterrupt:
                pass

        tan = input('Please enter TAN:')
        res = client.send_tan(res, tan)

    print(res.status)
    print(res.responses)

```

## 1.6 Creating SEPA debits

You can submit a SEPA debit XML file to the bank with the `sepa_debit` method:

```

class fints.client.FinTS3Client (bank_identifier, user_id, customer_id=None, from_data:
                                bytes = None, product_id=None, product_version='4.0.0',
                                mode=<FinTSClientMode.INTERACTIVE: 'interactive'>)

```

```

    sepa_debit (account: fints.models.SEPAAccount, pain_message: str, multiple=False,
                cor1=False, control_sum=None, currency='EUR', book_as_single=False,
                pain_descriptor='urn:iso:std:iso:20022:tech:xsd:pain.008.003.01')

```

Custom SEPA debit.

### Parameters

- **account** – SEPAAccount to send the debit from.
- **pain\_message** – SEPA PAIN message containing the debit details.
- **multiple** – Whether this message contains multiple debits.
- **cor1** – Whether to use COR1 debit (lead time reduced to 1 day)
- **control\_sum** – Sum of all debits (required if there are multiple)
- **currency** – Debit currency
- **book\_as\_single** – Kindly ask the bank to put multiple transactions as separate lines on the bank statement (defaults to False)
- **pain\_descriptor** – URN of the PAIN message schema used. Defaults to `urn:iso:std:iso:20022:tech:xsd:pain.008.003.01`.

**Returns** Returns either a `NeedRetryResponse` or `TransactionResponse` (with `data['task_id']` set, if available)

You should then enter a TAN, read our chapter [Working with TANs](#) to find out more.

### 1.6.1 Full example

You can easily generate XML using the `sepaxml` python library:

```
from sepaxml import SepaDD

config = {
    "name": "Test Company",
    "IBAN": "DE12345",
    "BIC": "BIC12345",
    "batch": False,
    "creditor_id": "TESTCORPID",
    "currency": "EUR",
}

sepa = SepaDD(config, schema="pain.008.002.02")
sepa.add_payment({
    "name": "Customer",
    "IBAN": "DE12345",
    "BIC": "BIC12345",
    "amount": 100,
    "type": "OOFF", # FRST, RCUR, OOFF, FNAL
    "collection_date": datetime.date.today() + datetime.timedelta(days=3),
    "mandate_id": "FINTSTEST1",
    "mandate_date": datetime.date(2018, 7, 26),
    "description": "FinTS Test transaction",
})
pain_message = sepa.export().decode()

client = FinTS3PinTanClient(...)
minimal_interactive_cli_bootstrap(client)

with client:
    if client.init_tan_response:
        print("A TAN is required", client.init_tan_response.challenge)

        if getattr(client.init_tan_response, 'challenge_hhduc', None):
            try:
                terminal_flicker_unix(client.init_tan_response.challenge_hhduc)
            except KeyboardInterrupt:
                pass

        tan = input('Please enter TAN:')
        client.send_tan(client.init_tan_response, tan)

    res = client.sepa_debit(
        account=accounts[0],
        data=pain_message,
        multiple=False,
        control_sum=Decimal('1.00'),
        pain_descriptor='urn:iso:std:iso:20022:tech:xsd:pain.008.002.02'
    )

    if isinstance(res, NeedTANResponse):
        print("A TAN is required", res.challenge)

        if getattr(res, 'challenge_hhduc', None):
            try:
```

(continues on next page)

(continued from previous page)

```
        terminal_flicker_unix(res.challenge_hhduc)
    except KeyboardInterrupt:
        pass

    tan = input('Please enter TAN:')
    res = client.send_tan(res, tan)

print(res.status)
print(res.responses)
```

## 1.7 Tested banks

The following banks have been tested with version 3.x of this library:

Bank	Transactions and Balance	Holdings	Transfer	Debits
Postbank	Yes			
BBBank eG	Yes		Yes	
Sparkasse Heidelberg	Yes			
comdirect	Yes			

### 1.7.1 Tested security functions

- 902 “photoTAN”
- 921 “pushTAN”
- 930 “mobile TAN”
- 942 “mobile TAN”
- 962 “Smart-TAN plus manuell”
- 972 “Smart-TAN plus optisch”

### 1.7.2 Legacy results

The following banks have been tested with the old version 1.x of this library:



Bank	Statements	Holdings	Transfer	Debits
BBBank eG	Yes		Yes	
CortalConsors	Yes	Yes		
comdirect		Yes		
GLS Bank eG	Yes		Yes	Yes
DKB	Yes			
ING DiBa	Yes			
netbank	Yes			
NIBC Direct	Yes			
Postbank	Yes			
Sparkasse	Yes			
Triodos Bank	Yes			
Volksbank (Fiducia)	Yes			
Wüstenrot	Yes			
1822direkt	Yes	Yes		

The following banks have been tested with the old version 2.x of this library:

Bank	Transactions and Balance	Holdings	Transfer	Debits
GLS Bank eG	Yes		Yes	Yes
Postbank	Yes			
Triodos Bank	Yes		Yes	
Volksbank Darmstadt-Süd Hessen	Yes		Yes	
Deutsche Skatbank	Yes		Yes	
BBBank eG	Yes		Yes	
MLP Banking AG	Yes			

## 1.8 Upgrading from python-fints 3.x to 4.x

Release 4.0 of this library was made to introduce a breaking change:

- You now need to register your application with the Deutsche Kreditwirtschaft (German banking association) and supply your assigned product IT when initializing the library.

The library used to have a built-in product ID that was used as a default if you didn't. This was very useful, but Deutsche Kreditwirtschaft asked us to stop doing this, since it undermines the whole point of the product registration. The ID included in prior versions of the library will be deactivated at some point and stop working.

## 1.9 Upgrading from python-fints 2.x to 3.x

Release 3.0 of this library was made to adjust to changes made by the banks as part of their PSD2 implementation in 2019. Here's what you should know when porting your code:

- A TAN can now be required for dialog initialization. In this case, `client.init_tan_response` will contain a `NeedTANResponse`.
- Basically every method of the client class can now return a `NeedTANResponse`, so you should always expect this case and handle it gracefully.
- Since everything can require a TAN, everything requires a standing dialog. Issuing interactive commands outside of a `with client:` statement is now deprecated. It still might work in very few cases, so we didn't disable

it, but we do not support it any longer. This affects you mostly when you work with this on a Python REPL or e.g. in a Notebook.

## 1.10 Upgrading from python-fints 1.x to 2.x

This library has seen a major rewrite in version 2.0 and the API has changed in a lot of places. These are the most important changes to know:

- The `get_statement` method was renamed to `get_transactions`. → [Reading account transactions](#)
- The `start_simple_sepa_transfer` method was renamed to `simple_sepa_transfer` and no longer takes a TAN method and TAN medium description as an argument. → [Sending SEPA transfers](#)
- The `start_sepa_transfer` method was renamed to `sepa_transfer` and no longer takes a TAN method and TAN medium description as an argument. The new parameter `pain_descriptor` should be passed with the version of the PAIN format, e.g. `urn:iso:std:iso:20022:tech:xsd:pain.001.001.03`. → [Sending SEPA transfers](#)
- The `start_sepa_debit` method was renamed to `sepa_debit` and no longer takes a TAN method and TAN medium description as an argument. The new parameter `pain_descriptor` should be passed with the version of the PAIN format, e.g. `urn:iso:std:iso:20022:tech:xsd:pain.008.003.01`. Also, a new parameter `cor1` is optionally available. → [Creating SEPA debits](#)
- Working with TANs has changed a lot. `get_tan_methos` has been renamed to `get_tan_mechanisms` and has a new return data type. The chosen TAN method is now set on a client level with `set_tan_mechanism` and `set_tan_medium`. You can find more information in the chapter [Working with TANs](#) and a full example in the chapter [Sending SEPA transfers](#).
- Debug logging output now contains parsed syntax structures instead of data blobs and is much easier to read.
- A new parser for FinTS has been added that is more robust and performs more validation.

In exchange, you get a couple of great new features:

- A new method `fints.client.FinTS3Client.get_information()` was added. → [Fetching bank information](#)
- It is now possible to serialize and store the state of the client to enable multi-step operations in a stateless environment. → [Storing and restoring client state](#)

## 1.11 Troubleshooting and bug reporting

The FinTS specification is long and complicated and in many parts leaves things open to interpretation – or sometimes implementors interpret things differently even though they're not really open to interpretation. This is valid for us, but also for the banks. Making the library work with many different banks is hard, and often impossible without access to a test account. Therefore, we ask you for patience when reporting issues with different banks – and you need to be ready that we might not be able to help you because we do not have the time or bank account required to dig deeper.

Therefore, if you run into trouble with this library, you first need to ask yourself a very important question: **Is it me or the library?** To answer this question for most cases, we have attached a script below, that we ask you to use to try the affected feature of the library in a well-documented way. Apart from changing the arguments (i.e. your bank's parameters and your credentials) at the top, we ask you **not to make any modifications**. Pasting this bit by bit into a Jupyter notebook **is a modification**. If your issue does not include information as to whether the script below works or does not work for your bank, **we will close your issue without further comment**.

If the script below does not work for you, there is probably a compatibility issue between this library and your bank. Feel free to open an issue, but make sure the issue title includes the name of the bank and the text includes what operations specifically fail.

If the script below does work for you, there is probably something wrong with your usage of the library or our documentation. Feel free to open an issue, but **include full working example code** that is necessary to reproduce the problem.

---

**Note:** Before posting anything on GitHub, make sure it does not contain your username, PIN, IBAN, or similarly sensitive data.

---

```
import datetime
import getpass
import logging
import sys
from decimal import Decimal

from fints.client import FinTS3PinTanClient, NeedTANResponse, \
    FinTSUnsupportedOperation
from fints.hhd.flicker import terminal_flicker_unix
from fints.utils import minimal_interactive_cli_bootstrap

logging.basicConfig(level=logging.DEBUG)

client_args = (
    'REPLACEME', # BLZ
    'REPLACEME', # USER
    getpass.getpass('PIN: '),
    'REPLACEME' # ENDPOINT
)

f = FinTS3PinTanClient(*client_args)
minimal_interactive_cli_bootstrap(f)

def ask_for_tan(response):
    print("A TAN is required")
    print(response.challenge)
    if getattr(response, 'challenge_hhduc', None):
        try:
            terminal_flicker_unix(response.challenge_hhduc)
        except KeyboardInterrupt:
            pass
    tan = input('Please enter TAN:')
    return f.send_tan(response, tan)

# Open the actual dialog
with f:
    # Since PSD2, a TAN might be needed for dialog initialization. Let's check if
    # there is one required
    if f.init_tan_response:
        ask_for_tan(f.init_tan_response)

    # Fetch accounts
    accounts = f.get_sepa_accounts()
```

(continues on next page)

(continued from previous page)

```

if isinstance(accounts, NeedTANResponse):
    accounts = ask_for_tan(accounts)
if len(accounts) == 1:
    account = accounts[0]
else:
    print("Multiple accounts available, choose one")
    for i, mm in enumerate(accounts):
        print(i, mm.iban)
    choice = input("Choice: ").strip()
    account = accounts[int(choice)]

# Test pausing and resuming the dialog
dialog_data = f.pause_dialog()

client_data = f.deconstruct(including_private=True)

f = FinTS3PinTanClient(*client_args, from_data=client_data)
with f.resume_dialog(dialog_data):
    while True:
        operations = [
            "End dialog",
            "Fetch transactions of the last 30 days",
            "Fetch transactions of the last 120 days",
            "Fetch transactions XML of the last 30 days",
            "Fetch transactions XML of the last 120 days",
            "Fetch information",
            "Fetch balance",
            "Fetch holdings",
            "Fetch scheduled debits",
            "Fetch status protocol",
            "Make a simple transfer"
        ]

        print("Choose an operation")
        for i, o in enumerate(operations):
            print(i, o)
        choice = int(input("Choice: ").strip())
        try:
            if choice == 0:
                break
            elif choice == 1:
                res = f.get_transactions(account, datetime.date.today() - datetime.
↳ timedelta(days=30),
                                datetime.date.today())
                while isinstance(res, NeedTANResponse):
                    res = ask_for_tan(res)
                print("Found", len(res), "transactions")
            elif choice == 2:
                res = f.get_transactions(account, datetime.date.today() - datetime.
↳ timedelta(days=120),
                                datetime.date.today())
                while isinstance(res, NeedTANResponse):
                    res = ask_for_tan(res)
                print("Found", len(res), "transactions")
            elif choice == 3:
                res = f.get_transactions_xml(account, datetime.date.today() -
↳ datetime.timedelta(days=30),

```

(continues on next page)

(continued from previous page)

```

                                datetime.date.today())
        while isinstance(res, NeedTANResponse):
            res = ask_for_tan(res)
            print("Found", len(res[0]) + len(res[1]), "XML documents")
        elif choice == 4:
            res = f.get_transactions_xml(account, datetime.date.today() -
↳datetime.timedelta(days=120),
                                datetime.date.today())
            while isinstance(res, NeedTANResponse):
                res = ask_for_tan(res)
                print("Found", len(res[0]) + len(res[1]), "XML documents")
        elif choice == 5:
            print(f.get_information())
        elif choice == 6:
            res = f.get_balance(account)
            while isinstance(res, NeedTANResponse):
                res = ask_for_tan(res)
            print(res)
        elif choice == 7:
            res = f.get_holdings(account)
            while isinstance(res, NeedTANResponse):
                res = ask_for_tan(res)
            print(res)
        elif choice == 8:
            res = f.get_scheduled_debits(account)
            while isinstance(res, NeedTANResponse):
                res = ask_for_tan(res)
            print(res)
        elif choice == 9:
            res = f.get_status_protocol()
            while isinstance(res, NeedTANResponse):
                res = ask_for_tan(res)
            print(res)
        elif choice == 10:
            res = f.simple_sepa_transfer(
                account=accounts[0],
                iban=input('Target IBAN:'),
                bic=input('Target BIC:'),
                amount=Decimal(input('Amount:')),
                recipient_name=input('Recipient name:'),
                account_name=input('Your name:'),
                reason=input('Reason:'),
                endtoend_id='NOTPROVIDED',
            )

            if isinstance(res, NeedTANResponse):
                ask_for_tan(res)
    except FintSUnsupportedOperation as e:
        print("This operation is not supported by this bank:", e)

```



---

Library developer documentation content

---

## 2.1 Developer documentation/API

This part of the documentation is for you if you want to improve python-fints, but also if you just want to look behind the curtain.

### 2.1.1 Parsing and serialization

**class** `fints.parser.FinTS3Parser`

Parser for FinTS/HBCI 3.0 messages

**parse\_message** (*data: bytes*) → `fints.types.SegmentSequence`

Takes a FinTS 3.0 message as byte array, and returns a parsed segment sequence

**class** `fints.parser.FinTS3Serializer`

Serializer for FinTS/HBCI 3.0 messages

**serialize\_message** (*message: fints.types.SegmentSequence*) → `bytes`

Serialize a message (as `SegmentSequence`, list of `FinTS3Segment`, or `FinTS3Segment`) into a byte array

Example usage:

```
>>> message = (b
↳ 'HNHBK:1:3+000000000428+300+430711670077=043999659571CN9D=+2+430711670077=043'
...      b"999659571CN9D=:2'HNVS:998:3+PIN:1+998+1+2::oIm3B1Hv6mQBAADYgbPpp?
↳ +kWrAQA+1+"
...      b"2:2:13:@8@00000000:5:1+280:15050500:hermes:S:0:0+0
↳ 'HNVSD:999:1+@195@HNSHK:2:"
...      b'4+PIN:1+999+9166926+1+1+2::oIm3B1Hv6mQBAADYgbPpp?
↳ +kWrAQA+1+1+1:999:1+6:10:16'
...      b"+280:15050500:hermes:S:0:0'HIRMG:3:2+0010::Nachricht_
↳ entgegengenommen.+0100:"
...      b":Dialog beendet.'HNSHA:4:2+9166926'HNHBS:5:1+2'")
>>> from fints.parser import FinTS3Parser
```

(continues on next page)

(continued from previous page)

```

>>> s = FinTS3Parser().parse_message(message)
>>> s
SegmentSequence([fints.segments.HNHBK3(header=fints.formals.SegmentHeader('HNHBK', 1,
↳3), message_size='000000000428', hbci_version=300, dialog_id=
↳'430711670077=043999659571CN9D=', message_number=2, reference_message=fints.formals.
↳ReferenceMessage(dialog_id='430711670077=043999659571CN9D=', message_number=2)),
↳fints.segments.HNVSK3(header=fints.formals.SegmentHeader('HNVSK', 998, 3), security_
↳profile=fints.formals.SecurityProfile(security_method='PIN', security_method_
↳version=1), security_function='998', security_role='1', security_identification_
↳details=fints.formals.SecurityIdentificationDetails(name_party='2', cid=None,
↳identifier_party='oIm3BlHv6mQBAADYgbPpp+kWrQA'), security_datetime=fints.formals.
↳SecurityDateTime(datetime_type='1'), encryption_algorithm=fints.formals.
↳EncryptionAlgorithm(usage_encryption='2', operation_mode='2', encryption_algorithm=
↳'13', algorithm_parameter_value=b'00000000', algorithm_parameter_name='5',
↳algorithm_parameter_iv_name='1'), key_name=fints.formals.KeyName(bank_
↳identifier=fints.formals.BankIdentifier(country_identifier='280', bank_code=
↳'15050500'), user_id='hermes', key_type='S', key_number=0, key_version=0),
↳compression_function='0'), fints.segments.HNVSD1(header=fints.formals.SegmentHeader(
↳'HNVSD', 999, 1), data=SegmentSequence([fints.segments.HNSHK4(header=fints.formals.
↳SegmentHeader('HNSHK', 2, 4), security_profile=fints.formals.
↳SecurityProfile(security_method='PIN', security_method_version=1), security_
↳function='999', security_reference='9166926', security_application_area='1',
↳security_role='1', security_identification_details=fints.formals.
↳SecurityIdentificationDetails(name_party='2', cid=None, identifier_party=
↳'oIm3BlHv6mQBAADYgbPpp+kWrQA'), security_reference_number=1, security_
↳datetime=fints.formals.SecurityDateTime(datetime_type='1'), hash_algorithm=fints.
↳formals.HashAlgorithm(usage_hash='1', hash_algorithm='999', algorithm_parameter_
↳name='1'), signature_algorithm=fints.formals.SignatureAlgorithm(usage_signature='6',
↳signature_algorithm='10', operation_mode='16'), key_name=fints.formals.
↳KeyName(bank_identifier=fints.formals.BankIdentifier(country_identifier='280', bank_
↳code='15050500'), user_id='hermes', key_type='S', key_number=0, key_version=0)),
↳fints.segments.HIRMG2(header=fints.formals.SegmentHeader('HIRMG', 3, 2),
↳responses=[fints.formals.Response(code='0010', reference_element=None, text=
↳'Nachricht entgegengenommen.'), fints.formals.Response(code='0100', reference_
↳element=None, text='Dialog beendet.')])), fints.segments.HNSHA2(header=fints.formals.
↳SegmentHeader('HNSHA', 4, 2), security_reference='9166926'))], fints.segments.
↳HNHBS1(header=fints.formals.SegmentHeader('HNHBS', 5, 1), message_number=2)])
>>> from fints.parser import FinTS3Serializer
>>> FinTS3Serializer().serialize_message(s)
b
↳"HNHBK:1:3+000000000428+300+430711670077=043999659571CN9D=+2+430711670077=043999659571CN9D=:2
↳'HNVSK:998:3+PIN:1+998+1+2::oIm3BlHv6mQBAADYgbPpp?
↳+kWrQA+1+2:2:13:@8@00000000:5:1+280:15050500:hermes:S:0:0+0
↳'HNVSD:999:1+@195@HNSHK:2:4+PIN:1+999+9166926+1+1+2::oIm3BlHv6mQBAADYgbPpp?
↳+kWrQA+1+1+1:999:1+6:10:16+280:15050500:hermes:S:0:0'HIRMG:3:2+0010::Nachricht_
↳entgegenommen.+0100::Dialog beendet.'HNSHA:4:2+9166926'HNHBS:5:1+2'"

```

**Note:** In general parsing followed by serialization is not idempotent: A message may contain empty list elements at the end, but our serializer will never generate them.

## 2.1.2 FinTS Segments

A segment is the core communication workhorse in FinTS. Each segment has a header of fixed format, which includes the segment type (“Segmentkennung”), number within the message, version, and, optionally, the number of the



segment of another message it is in response or relation to (“Bezugssegment”).

The header is followed by a nested structure of fields and groups of fields, the exact specification of which depends on the segment type and version.

All segment classes derive from `FinTS3Segment`, which specifies the header attribute of `SegmentHeader` type.

```
class fints.segments.base.FinTS3Segment(*args, **kwargs)
```

#### TYPE

Segment type. Will be determined from the class name in subclasses, if the class name consists only of uppercase characters followed by decimal digits. Subclasses may explicitly set a class attribute instead.

#### VERSION

Segment version. Will be determined from the class name in subclasses, if the class name consists only of uppercase characters followed by decimal digits. Subclasses may explicitly set a class attribute instead.

```
classmethod find_subclass(segment: list)
```

Parse the given `segment` parameter as a `SegmentHeader` and return a subclass with matching type and version class attributes.

#### header

Segmentkopf

Type `fints.formals.SegmentHeader`

```
print_nested(stream=None, level=0, indent=' ', prefix="", first_level_indent=True, trailer="",
              print_doc=True, first_line_suffix="")
```

Structured nested print of the object to the given stream.

The print-out is eval()able to reconstruct the object.

The `FinTS3Segment` class and its base classes employ a number of dynamic programming techniques so that derived classes need only specify the name, order and type of fields. All type conversion, construction etc. will take place automatically. All derived classes basically should behave “as expected”, returning only native Python datatypes.

Consider this example segment class:

```
class HNHBS1(FinTS3Segment):
    message_number = DataElementField(type='num', max_length=4)
```

Calling `print_nested` on an instance of this class might output:

```
fints.segments.HNHBS1(
    header = fints.formals.SegmentHeader('HNHBS', 4, 1),
    message_number = 1,
)
```

## All Segments

### fints.segments.accounts module

```
class fints.segments.accounts.HISPA1(*args, **kwargs)
```

Bases: `fints.segments.base.FinTS3Segment`

SEPA-Kontoverbindung rückmelden, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

```
TYPE = 'HISPA'
VERSION = 1

accounts
    SEPA-Kontoverbindung
        Type fints.formals.KTZ1

class fints.segments.accounts.HISPAS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment
    SEPA-Kontoverbindung anfordern, Parameter, version 1
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle
    TYPE = 'HISPAS'
    VERSION = 1
    parameter
        Parameter SEPA-Kontoverbindung anfordern
        Type fints.formals.GetSEPAAccountParameter1

class fints.segments.accounts.HKSPA1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    SEPA-Kontoverbindung anfordern, version 1
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle
    TYPE = 'HKSPA'
    VERSION = 1
    accounts
        Kontoverbindung
            Type fints.formals.Account3
```

## fints.segments.auth module

```
class fints.segments.auth.HIPINS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment
    PIN/TAN-spezifische Informationen, version 1
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN
    TYPE = 'HIPINS'
    VERSION = 1
    parameter
        Parameter PIN/TAN-spezifische Informationen
        Type fints.formals.ParameterPinTan

class fints.segments.auth.HITAB4(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    TAN-Generator/Liste anzeigen Bestand Rückmeldung, version 4
```

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HITAB'

**VERSION** = 4

**tan\_media\_list**

TAN-Medium-Liste

Type *fints.formals.TANMedia4*

**tan\_usage\_option**

TAN\_Einsatzoption

Type *fints.formals.TANUsageOption*

**class** *fints.segments.auth.HITAB5* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

TAN-Generator/Liste anzeigen Bestand Rückmeldung, version 5

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HITAB'

**VERSION** = 5

**tan\_media\_list**

TAN-Medium-Liste

Type *fints.formals.TANMedia5*

**tan\_usage\_option**

TAN\_Einsatzoption

Type *fints.formals.TANUsageOption*

**class** *fints.segments.auth.HITAN2* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Zwei-Schritt-TAN-Einreichung Rückmeldung, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HITAN'

**VERSION** = 2

**ben**

BEN

Type *str*

**challenge**

Challenge

Type *str*

**challenge\_valid\_until**

Gültigkeitsdatum und -uhrzeit für Challenge

Type *fints.formals.ChallengeValidUntil*

**tan\_list\_number**

TAN-Listennummer

Type *str*

```
tan_process
    TAN-Prozess
        Type str

task_hash_value
    Auftrags-Hashwert
        Type bytes

task_reference
    Auftragsreferenz
        Type str

class fints.segments.auth.HITAN3(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Zwei-Schritt-TAN-Einreichung Rückmeldung, version 3
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN
    TYPE = 'HITAN'
    VERSION = 3
    ben
        BEN
            Type str
    challenge
        Challenge
            Type str
    challenge_valid_until
        Gültigkeitsdatum und -uhrzeit für Challenge
            Type fints.formals.ChallengeValidUntil
    tan_list_number
        TAN-Listennummer
            Type str
    tan_medium_name
        Bezeichnung des TAN-Mediums
            Type str
    tan_process
        TAN-Prozess
            Type str
    task_hash_value
        Auftrags-Hashwert
            Type bytes
    task_reference
        Auftragsreferenz
            Type str
```

```

class fints.segments.auth.HITAN5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Zwei-Schritt-TAN-Einreichung Rückmeldung, version 5
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN
    TYPE = 'HITAN'
    VERSION = 5
    ben
        BEN
        Type str
    challenge
        Challenge
        Type str
    challenge_hhduc
        Challenge HHD_UC
        Type bytes
    challenge_valid_until
        Gültigkeitsdatum und -uhrzeit für Challenge
        Type fints.formals.ChallengeValidUntil
    tan_list_number
        TAN-Listennummer
        Type str
    tan_medium_name
        Bezeichnung des TAN-Mediums
        Type str
    tan_process
        TAN-Prozess
        Type str
    task_hash_value
        Auftrags-Hashwert
        Type bytes
    task_reference
        Auftragsreferenz
        Type str

class fints.segments.auth.HITAN6(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Zwei-Schritt-TAN-Einreichung Rückmeldung, version 6
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN
    TYPE = 'HITAN'
    VERSION = 6

```

**challenge**  
Challenge  
Type `str`

**challenge\_hhduc**  
Challenge HHD\_UC  
Type `bytes`

**challenge\_valid\_until**  
Gültigkeitsdatum und -uhrzeit für Challenge  
Type `fints.formals.ChallengeValidUntil`

**tan\_medium\_name**  
Bezeichnung des TAN-Mediums  
Type `str`

**tan\_process**  
TAN-Prozess  
Type `str`

**task\_hash\_value**  
Auftrags-Hashwert  
Type `bytes`

**task\_reference**  
Auftragsreferenz  
Type `str`

**class** `fints.segments.auth.HITANS1(*args, **kwargs)`  
Bases: `fints.segments.auth.HITANSBase`  
**TYPE** = `'HITANS'`  
**VERSION** = `1`  
**parameter**  
Type `fints.formals.ParameterTwostepTAN1`

**class** `fints.segments.auth.HITANS2(*args, **kwargs)`  
Bases: `fints.segments.auth.HITANSBase`  
**TYPE** = `'HITANS'`  
**VERSION** = `2`  
**parameter**  
Type `fints.formals.ParameterTwostepTAN2`

**class** `fints.segments.auth.HITANS3(*args, **kwargs)`  
Bases: `fints.segments.auth.HITANSBase`  
**TYPE** = `'HITANS'`  
**VERSION** = `3`  
**parameter**  
Type `fints.formals.ParameterTwostepTAN3`

```

class fints.segments.auth.HITANS4(*args, **kwargs)
    Bases: fints.segments.auth.HITANSBase

    TYPE = 'HITANS'

    VERSION = 4

    parameter
        Type fints.formals.ParameterTwostepTAN4

class fints.segments.auth.HITANS5(*args, **kwargs)
    Bases: fints.segments.auth.HITANSBase

    TYPE = 'HITANS'

    VERSION = 5

    parameter
        Type fints.formals.ParameterTwostepTAN5

class fints.segments.auth.HITANS6(*args, **kwargs)
    Bases: fints.segments.auth.HITANSBase

    TYPE = 'HITANS'

    VERSION = 6

    parameter
        Type fints.formals.ParameterTwostepTAN6

class fints.segments.auth.HITANSBase(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    TYPE = None

    VERSION = None

class fints.segments.auth.HKIDN2(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Identifikation, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    TYPE = 'HKIDN'

    VERSION = 2

    bank_identifier
        Kreditinstitutskennung
        Type fints.formals.BankIdentifier

    customer_id
        Kunden-ID
        Type str

    system_id
        Kundensystem-ID
        Type str

    system_id_status
        Kundensystem-Status

```

Type *fints.formals.SystemIDStatus*

**class** *fints.segments.auth.HKTAB4* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

TAN-Generator/Liste anzeigen Bestand, version 4

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HKTAB'

**VERSION** = 4

**tan\_media\_class**

TAN-Medium-Klasse

Type *fints.formals.TANMediaClass3*

**tan\_media\_type**

TAN-Medium-Art

Type *fints.formals.TANMediaType2*

**class** *fints.segments.auth.HKTAB5* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

TAN-Generator/Liste anzeigen Bestand, version 5

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HKTAB'

**VERSION** = 5

**tan\_media\_class**

TAN-Medium-Klasse

Type *fints.formals.TANMediaClass4*

**tan\_media\_type**

TAN-Medium-Art

Type *fints.formals.TANMediaType2*

**class** *fints.segments.auth.HKTAN2* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Zwei-Schritt-TAN-Einreichung, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = 'HKTAN'

**VERSION** = 2

**cancel\_task**

Auftrag stornieren

Type *bool*

**challenge\_class**

Challenge-Klasse

Type *int*

**further\_tan\_follows**

Weitere TAN folgt



**Type** `bool`

**parameter\_challenge\_class**  
Parameter Challenge-Klasse

**Type** `fints.formals.ParameterChallengeClass`

**tan\_list\_number**  
TAN-Listennummer

**Type** `str`

**tan\_process**  
TAN-Prozess

**Type** `str`

**task\_hash\_value**  
Auftrags-Hashwert

**Type** `bytes`

**task\_reference**  
Auftragsreferenz

**Type** `str`

**class** `fints.segments.auth.HKTAN3(*args, **kwargs)`  
Bases: `fints.segments.base.FinTS3Segment`  
Zwei-Schritt-TAN-Einreichung, version 3  
Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**TYPE** = `'HKTAN'`

**VERSION** = `3`

**cancel\_task**  
Auftrag stornieren

**Type** `bool`

**challenge\_class**  
Challenge-Klasse

**Type** `int`

**further\_tan\_follows**  
Weitere TAN folgt

**Type** `bool`

**parameter\_challenge\_class**  
Parameter Challenge-Klasse

**Type** `fints.formals.ParameterChallengeClass`

**tan\_list\_number**  
TAN-Listennummer

**Type** `str`

**tan\_medium\_name**  
Bezeichnung des TAN-Mediums

**Type** `str`

**tan\_process**  
TAN-Prozess  
**Type** `str`

**task\_hash\_value**  
Auftrags-Hashwert  
**Type** `bytes`

**task\_reference**  
Auftragsreferenz  
**Type** `str`

**class** `fints.segments.auth.HKTAN5(*args, **kwargs)`  
Bases: `fints.segments.base.FinTS3Segment`  
Zwei-Schritt-TAN-Einreichung, version 5  
Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN  
**TYPE** = `'HKTAN'`  
**VERSION** = `5`

**account**  
Kontoverbindung international Auftraggeber  
**Type** `fints.formals.KTI1`

**cancel\_task**  
Auftrag stornieren  
**Type** `bool`

**challenge\_class**  
Challenge-Klasse  
**Type** `int`

**further\_tan\_follows**  
Weitere TAN folgt  
**Type** `bool`

**parameter\_challenge\_class**  
Parameter Challenge-Klasse  
**Type** `fints.formals.ParameterChallengeClass`

**segment\_type**  
Segmentkennung  
**Type** `str`

**sms\_charge\_account**  
SMS-Abbuchungskonto  
**Type** `fints.formals.KTI1`

**tan\_list\_number**  
TAN-Listennummer  
**Type** `str`

**tan\_medium\_name**  
Bezeichnung des TAN-Mediums  
Type `str`

**tan\_process**  
TAN-Prozess  
Type `str`

**task\_hash\_value**  
Auftrags-Hashwert  
Type `bytes`

**task\_reference**  
Auftragsreferenz  
Type `str`

**class** `fints.segments.auth.HKTAN6(*args, **kwargs)`  
Bases: `fints.segments.base.FinTS3Segment`  
Zwei-Schritt-TAN-Einreichung, version 6  
Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN  
**TYPE** = `'HKTAN'`  
**VERSION** = `6`

**account**  
Kontoverbindung international Auftraggeber  
Type `fints.formals.KTI1`

**cancel\_task**  
Auftrag stornieren  
Type `bool`

**challenge\_class**  
Challenge-Klasse  
Type `int`

**further\_tan\_follows**  
Weitere TAN folgt  
Type `bool`

**parameter\_challenge\_class**  
Parameter Challenge-Klasse  
Type `fints.formals.ParameterChallengeClass`

**response\_hhd\_uc**  
Antwort HHD\_UC  
Type `fints.formals.ResponseHHDUC`

**segment\_type**  
Segmentkennung  
Type `str`

**sms\_charge\_account**  
SMS-Abbuchungskonto

Type *fints.formals.KTI1*

**tan\_medium\_name**  
Bezeichnung des TAN-Mediums

Type *str*

**tan\_process**  
TAN-Prozess

Type *str*

**task\_hash\_value**  
Auftrags-Hashwert

Type *bytes*

**task\_reference**  
Auftragsreferenz

Type *str*

**class** *fints.segments.auth.HKVVB3* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Verarbeitungsvorbereitung, version 3

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HKVVB'

**VERSION** = 3

**bpd\_version**  
BPD-Version

Type *int*

**language**  
Dialogsprache

Type *fints.formals.Language2*

**product\_name**  
Produktbezeichnung

Type *str*

**product\_version**  
Produktversion

Type *str*

**upd\_version**  
UPD-Version

Type *int*

## **fints.segments.bank module**

**class** *fints.segments.bank.HIBPA3* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Bankparameter allgemein, version 3

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HIBPA'

**VERSION** = 3

**bank\_identifier**

Kreditinstitutskennung

Type *fints.formals.BankIdentifier*

**bank\_name**

Kreditinstitutsbezeichnung

Type *str*

**bpd\_version**

BPD-Version

Type *int*

**max\_message\_length**

Maximale Nachrichtengröße

Type *int*

**max\_timeout**

Maximaler Timeout-Wert

Type *int*

**min\_timeout**

Minimaler Timeout-Wert

Type *int*

**number\_tasks**

Anzahl Geschäftsvorfallarten pro Nachricht

Type *int*

**supported\_hbci\_version**

Unterstützte HBCI-Versionen

Type *fints.formals.SupportedHBCIVersions2*

**supported\_languages**

Unterstützte Sprachen

Type *fints.formals.SupportedLanguages2*

**class** *fints.segments.bank.HIKOM4* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Kommunikationszugang rückmelden, version 4

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HIKOM'

**VERSION** = 4

**bank\_identifier**

Kreditinstitutskennung

Type *fints.formals.BankIdentifier*

**communication\_parameters**

Kommunikationsparameter

Type *fints.formals.CommunicationParameter2*

**default\_language**

Standardsprache

Type *fints.formals.Language2*

**class** *fints.segments.bank.HIUPA4* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Userparameter allgemein

**TYPE** = 'HIUPA'

**VERSION** = 4

**extension**

Erweiterung, allgemein

Type *str*

**upd\_usage**

UPD-Verwendung

Type *fints.formals.UPDUsage*

**upd\_version**

UPD-Version

Type *int*

**user\_identifier**

Benutzerkennung

Type *str*

**username**

Benutzername

Type *str*

**class** *fints.segments.bank.HIUPD6* (\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Kontoinformationen

**TYPE** = 'HIUPD'

**VERSION** = 6

**account\_currency**

Kontowährung

Type *str*

**account\_information**

Kontoverbindung

Type *fints.formals.AccountInformation*

**account\_limit**

Kontolimit

Type *fints.formals.AccountLimit*

**account\_product\_name**  
Kontoproduktbezeichnung  
**Type** `str`

**account\_type**  
Kontoart  
**Type** `int`

**allowed\_transactions**  
Erlaubte Geschäftsvorfälle  
**Type** `fints.formals.AllowedTransaction`

**customer\_id**  
Kunden-ID  
**Type** `str`

**extension**  
Erweiterung, kontobezogen  
**Type** `str`

**iban**  
IBAN  
**Type** `str`

**name\_account\_owner\_1**  
Name des Kontoinhabers 1  
**Type** `str`

**name\_account\_owner\_2**  
Name des Kontoinhabers 2  
**Type** `str`

**class** `fints.segments.bank.HKKOM4(*args, **kwargs)`  
Bases: `fints.segments.base.FinTS3Segment`  
Kommunikationszugang anfordern, version 4  
Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals  
**TYPE** = `'HKKOM'`  
**VERSION** = `4`

**end\_bank\_identifier**  
Bis Kreditinstitutskennung  
**Type** `fints.formals.BankIdentifier`

**max\_number\_responses**  
Maximale Anzahl Einträge  
**Type** `int`

**start\_bank\_identifier**  
Von Kreditinstitutskennung  
**Type** `fints.formals.BankIdentifier`

**touchdown\_point**  
Aufsetzpunkt  
  
Type `str`

### fints.segments.base module

```
class fints.segments.base.FinTS3SegmentMeta
    Bases: fints.types.ContainerMeta

    mro()
        Return a type's method resolution order.

class fints.segments.base.ParameterSegment(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    TYPE = None

    VERSION = None

    max_number_tasks
        Maximale Anzahl Aufträge

        Type int

    min_number_signatures
        Anzahl Signaturen mindestens

        Type int

    security_class
        Sicherheitsklasse

        Type fints.formals.SecurityClass

class fints.segments.base.ParameterSegment_22(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    TYPE = None

    VERSION = None

    max_number_tasks
        Maximale Anzahl Aufträge

        Type int

    min_number_signatures
        Anzahl Signaturen mindestens

        Type int
```

### fints.segments.debit module

```
class fints.segments.debit.BatchDebitBase(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    TYPE = None

    VERSION = None
```



```

account
    Kontoverbindung international

    Type fints.formals.KTI1

request_single_booking
    Einzelbuchung gewünscht

    Type bool

sepa_descriptor
    SEPA Descriptor

    Type str

sepa_pain_message
    SEPA pain message

    Type bytes

sum_amount
    Summenfeld

    Type fints.formals.Amount1

class fints.segments.debit.DebitResponseBase(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    TYPE = None

    VERSION = None

    task_id
        Auftragsidentifikation

        Type str

class fints.segments.debit.HIDBS1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Bestand terminierter SEPA-Einzellastschriften rückmelden, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDBS'

    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    sepa_descriptor
        SEPA Descriptor

        Type str

    sepa_pain_message
        SEPA pain message

        Type bytes

    task_cancelable
        Auftrag löschar

```

```
        Type bool
task_changeable
    Auftrag änderbar
        Type bool
task_id
    Auftragsidentifikation
        Type str
class fints.segments.debit.HIDBS2(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Bestand terminierter SEPA-Einzellastschriften rückmelden, version 2
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle
    TYPE = 'HIDBS'
    VERSION = 2
    account
        Kontoverbindung international
        Type fints.formals.KTI1
    sepa_c_code
        SEPA-C-Code
        Type fints.formals.SEPACCode1
    sepa_descriptor
        SEPA Descriptor
        Type str
    sepa_pain_message
        SEPA pain message
        Type bytes
    status_sepa_task
        Status SEPA-Auftrag
        Type fints.formals.StatusSEPATask1
    task_changeable
        Auftrag änderbar
        Type bool
    task_id
        Auftragsidentifikation
        Type str
class fints.segments.debit.HIDBSS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment
    Bestand terminierter SEPA-Einzellastschriften Parameter, version 1
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle
```

```

TYPE = 'HIDBSS'
VERSION = 1

parameter
    Parameter Bestand terminierter SEPA-Einzellastschriften

    Type fints.formals.QueryScheduledDebitParameter1

class fints.segments.debit.HIDBSS2(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Bestand terminierter SEPA-Einzellastschriften Parameter, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDBSS'
    VERSION = 2

    parameter
        Parameter Bestand terminierter SEPA-Einzellastschriften

        Type fints.formals.QueryScheduledDebitParameter2

class fints.segments.debit.HIDMB1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Bestand terminierter SEPA-Sammellastschriften rückmelden, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMB'
    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    date_booked
        Ausführungsdatum

        Type datetime.date

    date_entered
        Einreichungsdatum

        Type datetime.date

    debit_count
        Anzahl der Aufträge

        Type int

    sum_amount
        Summe der Beträge

        Type fints.formals.Amount1

    task_id
        Auftragsidentifikation

        Type str

```

```
class fints.segments.debit.HIDMBS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Bestand terminierter SEPA-Sammellastschriften Parameter, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMBS'

    VERSION = 1

    parameter
        Parameter Bestand terminierter SEPA-Sammellastschriften

        Type fints.formals.QueryScheduledBatchDebitParameter1
```

```
class fints.segments.debit.HIDMC1(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-COR1-Sammellastschrift bestätigen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMC'

    VERSION = 1
```

```
class fints.segments.debit.HIDMCS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-COR1-Sammellastschrift Parameter, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMCS'

    VERSION = 1

    parameter
        Parameter terminierte SEPA-COR1-Sammellastschrift

        Type fints.formals.ScheduledCOR1BatchDebitParameter1
```

```
class fints.segments.debit.HIDME1(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-Sammellastschrift bestätigen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDME'

    VERSION = 1
```

```
class fints.segments.debit.HIDME2(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-Sammellastschrift bestätigen, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDME'
```

```

VERSION = 2

class fints.segments.debit.HIDMES1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-Sammellastschrift einreichen Parameter, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMES'

    VERSION = 1

    parameter
        Parameter terminierte SEPA-Sammellastschrift einreichen

        Type fints.formals.ScheduledBatchDebitParameter1

class fints.segments.debit.HIDMES2(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-Sammellastschrift einreichen Parameter, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDMES'

    VERSION = 2

    parameter
        Parameter terminierte SEPA-Sammellastschrift einreichen

        Type fints.formals.ScheduledBatchDebitParameter2

class fints.segments.debit.HIDSC1(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-COR1-Einzellastschrift bestätigen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSC'

    VERSION = 1

class fints.segments.debit.HIDSCS1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-COR1-Einzellastschrift Parameter, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSCS'

    VERSION = 1

    parameter
        Parameter terminierte SEPA-COR1-Einzellastschrift

        Type fints.formals.ScheduledCOR1DebitParameter1

```

```
class fints.segments.debit.HIDSE1(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-Einzellastschrift bestätigen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSE'

    VERSION = 1
```

```
class fints.segments.debit.HIDSE2(*args, **kwargs)
    Bases: fints.segments.debit.DebitResponseBase

    Einreichung terminierter SEPA-Einzellastschrift bestätigen, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSE'

    VERSION = 2
```

```
class fints.segments.debit.HIDSES1(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-Einzellastschrift einreichen Parameter, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSES'

    VERSION = 1

    parameter
        Parameter terminierte SEPA-Sammellastschrift einreichen

        Type fints.formals.ScheduledDebitParameter1
```

```
class fints.segments.debit.HIDSES2(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Terminierte SEPA-Einzellastschrift einreichen Parameter, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIDSES'

    VERSION = 2

    parameter
        Parameter terminierte SEPA-Sammellastschrift einreichen

        Type fints.formals.ScheduledDebitParameter2
```

```
class fints.segments.debit.HKDBS1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Bestand terminierter SEPA-Einzellastschriften anfordern, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDBS'
```

```

VERSION = 1

account
    Kontoverbindung international

    Type fints.formals.KTI1

date_end
    Bis Datum

    Type datetime.date

date_start
    Von Datum

    Type datetime.date

max_number_responses
    Maximale Anzahl Einträge

    Type int

supported_sepa_pain_messages
    Unterstützte SEPA pain messages

    Type fints.formals.SupportedSEPAPainMessages1

touchdown_point
    Aufsetzpunkt

    Type str

class fints.segments.debit.HKDBS2 (*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Bestand terminierter SEPA-Einzellastschriften anfordern, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDBS'

    VERSION = 2

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    date_end
        Bis Datum

        Type datetime.date

    date_start
        Von Datum

        Type datetime.date

    max_number_responses
        Maximale Anzahl Einträge

        Type int

    supported_sepa_pain_messages
        Unterstützte SEPA pain messages

```

Type *fints.formals.SupportedSEPA PainMessages1*

**touchdown\_point**

Aufsetzpunkt

Type *str*

**class** *fints.segments.debit.HKDMB1*(\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Bestand terminierter SEPA-Sammellastschriften anfordern, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKDMB'

**VERSION** = 1

**account**

Kontoverbindung international

Type *fints.formals.KTI1*

**date\_end**

Bis Datum

Type *datetime.date*

**date\_start**

Von Datum

Type *datetime.date*

**max\_number\_responses**

Maximale Anzahl Einträge

Type *int*

**touchdown\_point**

Aufsetzpunkt

Type *str*

**class** *fints.segments.debit.HKDMC1*(\*args, \*\*kwargs)

Bases: *fints.segments.debit.BatchDebitBase*

Terminierte SEPA-COR1-Sammellastschrift einreichen, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKDMC'

**VERSION** = 1

**class** *fints.segments.debit.HKDME1*(\*args, \*\*kwargs)

Bases: *fints.segments.debit.BatchDebitBase*

Einreichung terminierter SEPA-Sammellastschrift, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKDME'

**VERSION** = 1



```

class fints.segments.debit.HKDME2(*args, **kwargs)
    Bases: fints.segments.debit.BatchDebitBase

    Einreichung terminierter SEPA-Sammellastschrift, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDME'

    VERSION = 2

class fints.segments.debit.HKDSC1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Terminierte SEPA-COR1-Einzellastschrift einreichen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDSC'

    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    sepa_descriptor
        SEPA Descriptor

        Type str

    sepa_pain_message
        SEPA pain message

        Type bytes

class fints.segments.debit.HKDSE1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Terminierte SEPA-Einzellastschrift einreichen, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDSE'

    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    sepa_descriptor
        SEPA Descriptor

        Type str

    sepa_pain_message
        SEPA pain message

        Type bytes

```

```
class fints.segments.debit.HKDSE2(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Terminierte SEPA-Einzellastschrift einreichen, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKDSE'

    VERSION = 2

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    sepa_descriptor
        SEPA Descriptor

        Type str

    sepa_pain_message
        SEPA pain message

        Type bytes
```

### fints.segments.depot module

```
class fints.segments.depot.HIWPD5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Depotaufstellung rückmelden, version 5

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HIWPD'

    VERSION = 5

    holdings
        Depotaufstellung

        Type bytes

class fints.segments.depot.HIWPD6(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Depotaufstellung rückmelden, version 6

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIWPD'

    VERSION = 6

    holdings
        Depotaufstellung

        Type bytes

class fints.segments.depot.HKWPD5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
```

Depotaufstellung anfordern, version 5

Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

**TYPE** = 'HKWPD'

**VERSION** = 5

**account**

Depot

Type *fints.formals.Account2*

**currency**

Währung der Depotaufstellung

Type *str*

**max\_number\_responses**

Maximale Anzahl Einträge

Type *int*

**quality**

Kursqualität

Type *int*

**touchdown\_point**

Aufsetzpunkt

Type *str*

**class** *fints.segments.depot.HKWPD6*(\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Depotaufstellung anfordern, version 6

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKWPD'

**VERSION** = 6

**account**

Depot

Type *fints.formals.Account3*

**currency**

Währung der Depotaufstellung

Type *str*

**max\_number\_responses**

Maximale Anzahl Einträge

Type *int*

**quality**

Kursqualität

Type *str*

**touchdown\_point**

Aufsetzpunkt

Type `str`

### **fints.segments.dialog module**

**class** `fints.segments.dialog.HIRMG2(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Rückmeldungen zur Gesamtnachricht

**TYPE** = `'HIRMG'`

**VERSION** = `2`

**responses**

Rückmeldung

Type `fints.formals.Response`

**class** `fints.segments.dialog.HIRMS2(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Rückmeldungen zu Segmenten

**TYPE** = `'HIRMS'`

**VERSION** = `2`

**responses**

Rückmeldung

Type `fints.formals.Response`

**class** `fints.segments.dialog.HISYN4(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Synchronisierungsantwort

**TYPE** = `'HISYN'`

**VERSION** = `4`

**message\_number**

Nachrichtennummer

Type `int`

**security\_reference\_digital\_signature**

Sicherheitsreferenznummer für Digitale Signatur

Type `int`

**security\_reference\_signature\_key**

Sicherheitsreferenznummer für Signierschlüssel

Type `int`

**system\_id**

Kundensystem-ID

Type `str`

**class** `fints.segments.dialog.HKEND1(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Dialogende, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HKEND'

**VERSION** = 1

**dialog\_id**  
Dialog-ID

**Type** *str*

**class** *fints.segments.dialog.HKSYN3(\*args, \*\*kwargs)*

Bases: *fints.segments.base.FinTS3Segment*

Synchronisierung, version 3

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HKSYN'

**VERSION** = 3

**synchronization\_mode**

**Type** *fints.formals.SynchronizationMode*

## **fints.segments.journal module**

**class** *fints.segments.journal.HIPRO3(\*args, \*\*kwargs)*

Bases: *fints.segments.base.FinTS3Segment*

Statusprotokoll rückmelden, version 3

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = 'HIPRO'

**VERSION** = 3

**date**  
Datum

**Type** *datetime.date*

**reference**  
Bezugssegment

**Type** *int*

**reference\_message**  
Bezugsnachricht

**Type** *fints.formals.ReferenceMessage*

**responses**  
Rückmeldung

**Type** *fints.formals.Response*

**time**  
Uhrzeit

**Type** *datetime.time*

```
class fints.segments.journal.HIPRO4(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Statusprotokoll rückmelden, version 4

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    TYPE = 'HIPRO'

    VERSION = 4

    date
        Datum

        Type datetime.date

    reference
        Bezugssegment

        Type int

    reference_message
        Bezugsnachricht

        Type fints.formals.ReferenceMessage

    responses
        Rückmeldung

        Type fints.formals.Response

    time
        Uhrzeit

        Type datetime.time

class fints.segments.journal.HIPROS3(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment_22

    Statusprotokoll Parameter, version 3

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    TYPE = 'HIPROS'

    VERSION = 3

class fints.segments.journal.HIPROS4(*args, **kwargs)
    Bases: fints.segments.base.ParameterSegment

    Statusprotokoll Parameter, version 4

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    TYPE = 'HIPROS'

    VERSION = 4

class fints.segments.journal.HKPRO3(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Statusprotokoll anfordern, version 3

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    TYPE = 'HKPRO'

    VERSION = 3
```

**date\_end**  
Bis Datum

Type `datetime.date`

**date\_start**  
Von Datum

Type `datetime.date`

**max\_number\_responses**  
Maximale Anzahl Einträge

Type `int`

**touchdown\_point**  
Aufsetzpunkt

Type `str`

**class** `fints.segments.journal.HKPRO4(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Statusprotokoll anfordern, version 4

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**TYPE** = `'HKPRO'`

**VERSION** = `4`

**date\_end**  
Bis Datum

Type `datetime.date`

**date\_start**  
Von Datum

Type `datetime.date`

**max\_number\_responses**  
Maximale Anzahl Einträge

Type `int`

**touchdown\_point**  
Aufsetzpunkt

Type `str`

## **fints.segments.message module**

**class** `fints.segments.message.HNHBK3(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Nachrichtenkopf

**TYPE** = `'HNHBK'`

**VERSION** = `3`

**dialog\_id**  
Dialog-ID

```
    Type str
hbciversion
    HBCI-Version

    Type int
message_number
    Nachrichtennummer

    Type int
message_size
    Größe der Nachricht (nach Verschlüsselung und Komprimierung)

    Type int
reference_message
    Bezugsnachricht

    Type fints.formals.ReferenceMessage
class fints.segments.message.HNHBS1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Nachrichtenabschluss

    TYPE = 'HNHBS'
    VERSION = 1

    message_number
    Nachrichtennummer

    Type int
class fints.segments.message.HNSHA2(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Signaturabschluss, version 2

    Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

    TYPE = 'HNSHA'
    VERSION = 2

    security_reference
    Sicherheitskontrollreferenz

    Type str
user_defined_signature
    Benutzerdefinierte Signatur

    Type fints.formals.UserDefinedSignature
validation_result
    Validierungsergebnis

    Type bytes
class fints.segments.message.HNSHK4(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Signaturkopf, version 4

    Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI
```



```

TYPE = 'HNSHK'

VERSION = 4

certificate
    Zertifikat

    Type fints.formals.Certificate

hash_algorithm
    Hashalgorithmus

    Type fints.formals.HashAlgorithm

key_name
    Schlüsselname

    Type fints.formals.KeyName

security_application_area
    Bereich der Sicherheitsapplikation, kodiert

    Type fints.formals.SecurityApplicationArea

security_datetime
    Sicherheitsdatum und -uhrzeit

    Type fints.formals.SecurityDateTime

security_function
    Sicherheitsfunktion, kodiert

    Type str

security_identification_details
    Sicherheitsidentifikation, Details

    Type fints.formals.SecurityIdentificationDetails

security_profile
    Sicherheitsprofil

    Type fints.formals.SecurityProfile

security_reference
    Sicherheitskontrollreferenz

    Type str

security_reference_number
    Sicherheitsreferenznummer

    Type int

security_role
    Rolle des Sicherheitslieferanten, kodiert

    Type fints.formals.SecurityRole

signature_algorithm
    Signaturalgorithmus

    Type fints.formals.SignatureAlgorithm

class fints.segments.message.HNVSD1 (*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

```

Verschlüsselte Daten, version 1

Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

**TYPE** = 'HNVSD'

**VERSION** = 1

**data**

Daten, verschlüsselt

**class** `fints.segments.message.HNVSK3(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Verschlüsselungskopf, version 3

Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

**TYPE** = 'HNVSK'

**VERSION** = 3

**certificate**

Zertifikat

**Type** `fints.formals.Certificate`

**compression\_function**

Komprimierungsfunktion

**Type** `fints.formals.CompressionFunction`

**encryption\_algorithm**

Verschlüsselungsalgorithmus

**Type** `fints.formals.EncryptionAlgorithm`

**key\_name**

Schlüsselname

**Type** `fints.formals.KeyName`

**security\_datetime**

Sicherheitsdatum und -uhrzeit

**Type** `fints.formals.SecurityDateTime`

**security\_function**

Sicherheitsfunktion, kodiert

**Type** `str`

**security\_identification\_details**

Sicherheitsidentifikation, Details

**Type** `fints.formals.SecurityIdentificationDetails`

**security\_profile**

Sicherheitsprofil

**Type** `fints.formals.SecurityProfile`

**security\_role**

Rolle des Sicherheitslieferanten, kodiert

**Type** `fints.formals.SecurityRole`

**fints.segments.saldo module**

```

class fints.segments.saldo.HISAL5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment
    Saldenrückmeldung, version 5
    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HISAL'

    VERSION = 5

    account
        Kontoverbindung Auftraggeber
        Type fints.formals.Account2

    account_product
        Kontoproduktbezeichnung
        Type str

    available_amount
        Verfügbarer Betrag
        Type fints.formals.Amount1

    balance_booked
        Gebuchter Saldo
        Type fints.formals.Balance1

    balance_pending
        Saldo der vorgemerkten Umsätze
        Type fints.formals.Balance1

    booking_date
        Buchungsdatum des Saldos
        Type datetime.date

    booking_time
        Buchungsuhrzeit des Saldos
        Type datetime.time

    currency
        Kontowährung
        Type str

    date_due
        Fälligkeit
        Type datetime.date

    line_of_credit
        Kreditlinie
        Type fints.formals.Amount1

    used_amount
        Bereits verfügbarer Betrag
        Type fints.formals.Amount1

```

```
class fints.segments.saldo.HISAL6(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Saldenrückmeldung, version 6

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HISAL'

    VERSION = 6

    account
        Kontoverbindung Auftraggeber

        Type fints.formals.Account3

    account_product
        Kontoproduktbezeichnung

        Type str

    available_amount
        Verfügbarer Betrag

        Type fints.formals.Amount1

    balance_booked
        Gebuchter Saldo

        Type fints.formals.Balance2

    balance_pending
        Saldo der vorgemerkten Umsätze

        Type fints.formals.Balance2

    booking_timestamp
        Buchungszeitpunkt

        Type fints.formals.Timestamp1

    currency
        Kontowährung

        Type str

    date_due
        Fälligkeit

        Type datetime.date

    line_of_credit
        Kreditlinie

        Type fints.formals.Amount1

    overdraft
        Überziehung

        Type fints.formals.Amount1

    used_amount
        Bereits verfügbarer Betrag

        Type fints.formals.Amount1
```

```
class fints.segments.saldo.HISAL7(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Saldenrückmeldung, version 7

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HISAL'

    VERSION = 7

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    account_product
        Kontoproduktbezeichnung

        Type str

    available_amount
        Verfügbarer Betrag

        Type fints.formals.Amount1

    balance_booked
        Gebuchter Saldo

        Type fints.formals.Balance2

    balance_pending
        Saldo der vorgemerkten Umsätze

        Type fints.formals.Balance2

    booking_timestamp
        Buchungszeitpunkt

        Type fints.formals.Timestamp1

    currency
        Kontowährung

        Type str

    date_due
        Fälligkeit

        Type datetime.date

    line_of_credit
        Kreditlinie

        Type fints.formals.Amount1

    overdraft
        Überziehung

        Type fints.formals.Amount1

    used_amount
        Bereits verfügbarer Betrag

        Type fints.formals.Amount1
```

```
class fints.segments.saldo.HKSAL5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Saldenabfrage, version 5

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HKSAL'

    VERSION = 5

    account
        Kontoverbindung Auftraggeber

        Type fints.formals.Account2

    all_accounts
        Alle Konten

        Type bool

    max_number_responses
        Maximale Anzahl Einträge

        Type int

    touchdown_point
        Aufsetzpunkt

        Type str

class fints.segments.saldo.HKSAL6(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Saldenabfrage, version 6

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKSAL'

    VERSION = 6

    account
        Kontoverbindung Auftraggeber

        Type fints.formals.Account3

    all_accounts
        Alle Konten

        Type bool

    max_number_responses
        Maximale Anzahl Einträge

        Type int

    touchdown_point
        Aufsetzpunkt

        Type str

class fints.segments.saldo.HKSAL7(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Saldenabfrage, version 7
```

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKSAL'

**VERSION** = 7

**account**

Kontoverbindung international

Type *fints.formals.KTI1*

**all\_accounts**

Alle Konten

Type *bool*

**max\_number\_responses**

Maximale Anzahl Einträge

Type *int*

**touchdown\_point**

Aufsetzpunkt

Type *str*

## fints.segments.statement module

**class** *fints.segments.statement.DIKKU2*(\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Kreditkartenumsätze rückmelden, version 2

Source: Reverse engineered

**TYPE** = 'DIKKU'

**VERSION** = 2

**class** *fints.segments.statement.DIKKUS2*(\*args, \*\*kwargs)

Bases: *fints.segments.base.ParameterSegment*

Kreditkartenumsätze anfordern Parameter, version 2

Source: Reverse engineered

**TYPE** = 'DIKKUS'

**VERSION** = 2

**parameter**

Parameter Kreditkartenumsätze anfordern

Type *fints.formals.QueryCreditCardStatements2*

**class** *fints.segments.statement.DKKKU2*(\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

Kreditkartenumsätze anfordern, version 2

Source: Reverse engineered

**TYPE** = 'DKKKU'

**VERSION** = 2

**account**  
Kontoverbindung Auftraggeber  
Type `fints.formals.Account2`

**credit\_card\_number**  
Kreditkartennummer  
Type `str`

**date\_end**  
Bis Datum  
Type `datetime.date`

**date\_start**  
Von Datum  
Type `datetime.date`

**max\_number\_responses**  
Maximale Anzahl Einträge  
Type `int`

**subaccount**  
Subaccount?  
Type `str`

**touchdown\_point**  
Aufsetzpunkt  
Type `str`

**class** `fints.segments.statement.HICAZ1(*args, **kwargs)`  
Bases: `fints.segments.base.FinTS3Segment`  
Kontoumsätze rückmelden/Zeitraum, version 1  
Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation  
**TYPE** = `'HICAZ'`  
**VERSION** = `1`

**account**  
Kontoverbindung Auftraggeber  
Type `fints.formals.KTI1`

**camt\_descriptor**  
camt-Deskriptor  
Type `str`

**statement\_booked**  
Gebuchte Umsätze  
Type `fints.formals.BookedCamtStatements1`

**statement\_pending**  
Nicht gebuchte Umsätze  
Type `bytes`



```
class fints.segments.statement.HIKAZ5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Kontoumsätze rückmelden/Zeitraum, version 5

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HIKAZ'

    VERSION = 5

    statement_booked
        Gebuchte Umsätze

        Type bytes

    statement_pending
        Nicht gebuchte Umsätze

        Type bytes

class fints.segments.statement.HIKAZ6(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Kontoumsätze rückmelden/Zeitraum, version 6

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIKAZ'

    VERSION = 6

    statement_booked
        Gebuchte Umsätze

        Type bytes

    statement_pending
        Nicht gebuchte Umsätze

        Type bytes

class fints.segments.statement.HIKAZ7(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Kontoumsätze rückmelden/Zeitraum, version 7

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HIKAZ'

    VERSION = 7

    statement_booked
        Gebuchte Umsätze

        Type bytes

    statement_pending
        Nicht gebuchte Umsätze

        Type bytes
```

```
class fints.segments.statement.HKCAZ1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Kontoumsätze anfordern/Zeitraum, version 5

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HKCAZ'

    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    all_accounts
        Alle Konten

        Type bool

    date_end
        Bis Datum

        Type datetime.date

    date_start
        Von Datum

        Type datetime.date

    max_number_responses
        Maximale Anzahl Einträge

        Type int

    supported_camt_messages
        Kontoverbindung international

        Type fints.formals.SupportedMessageTypes

    touchdown_point
        Aufsetzpunkt

        Type str

class fints.segments.statement.HKKAZ5(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    Kontoumsätze anfordern/Zeitraum, version 5

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    TYPE = 'HKKAZ'

    VERSION = 5

    account
        Kontoverbindung Auftraggeber

        Type fints.formals.Account2

    all_accounts
        Alle Konten

        Type bool
```

**date\_end**  
Bis Datum

Type `datetime.date`

**date\_start**  
Von Datum

Type `datetime.date`

**max\_number\_responses**  
Maximale Anzahl Einträge

Type `int`

**touchdown\_point**  
Aufsetzpunkt

Type `str`

**class** `fints.segments.statement.HKKAZ6(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Kontoumsätze anfordern/Zeitraum, version 6

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = `'HKKAZ'`

**VERSION** = `6`

**account**  
Kontoverbindung Auftraggeber

Type `fints.formals.Account3`

**all\_accounts**  
Alle Konten

Type `bool`

**date\_end**  
Bis Datum

Type `datetime.date`

**date\_start**  
Von Datum

Type `datetime.date`

**max\_number\_responses**  
Maximale Anzahl Einträge

Type `int`

**touchdown\_point**  
Aufsetzpunkt

Type `str`

**class** `fints.segments.statement.HKKAZ7(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

Kontoumsätze anfordern/Zeitraum, version 7

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKKAZ'

**VERSION** = 7

**account**

Kontoverbindung international

Type *fints.formals.KTI1*

**all\_accounts**

Alle Konten

Type *bool*

**date\_end**

Bis Datum

Type *datetime.date*

**date\_start**

Von Datum

Type *datetime.date*

**max\_number\_responses**

Maximale Anzahl Einträge

Type *int*

**touchdown\_point**

Aufsetzpunkt

Type *str*

## fints.segments.transfer module

**class** *fints.segments.transfer.HICCMS1*(\*args, \*\*kwargs)

Bases: *fints.segments.base.ParameterSegment*

SEPA-Sammelüberweisung Parameter, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HICCMS'

**VERSION** = 1

**parameter**

Parameter SEPA-Sammelüberweisung

Type *fints.formals.BatchTransferParameter1*

**class** *fints.segments.transfer.HKCCM1*(\*args, \*\*kwargs)

Bases: *fints.segments.base.FinTS3Segment*

SEPA-Sammelüberweisung, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKCCM'

```

VERSION = 1

account
    Kontoverbindung international

    Type fints.formals.KTI1

request_single_booking
    Einzelbuchung gewünscht

    Type bool

sepa_descriptor
    SEPA Descriptor

    Type str

sepa_pain_message
    SEPA pain message

    Type bytes

sum_amount
    Summenfeld

    Type fints.formals.Amount1

class fints.segments.transfer.HKCCS1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    SEPA Einzelüberweisung, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKCCS'

    VERSION = 1

    account
        Kontoverbindung international

        Type fints.formals.KTI1

    sepa_descriptor
        SEPA Descriptor

        Type str

    sepa_pain_message
        SEPA pain message

        Type bytes

class fints.segments.transfer.HKIPM1(*args, **kwargs)
    Bases: fints.segments.base.FinTS3Segment

    SEPA-instant Sammelüberweisung, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    TYPE = 'HKIPM'

    VERSION = 1

```

**account**

Kontoverbindung international

Type `fints.formals.KTI1`

**request\_single\_booking**

Einzelbuchung gewünscht

Type `bool`

**sepa\_descriptor**

SEPA Descriptor

Type `str`

**sepa\_pain\_message**

SEPA pain message

Type `bytes`

**sum\_amount**

Summenfeld

Type `fints.formals.Amount1`

**class** `fints.segments.transfer.HKIPZ1(*args, **kwargs)`

Bases: `fints.segments.base.FinTS3Segment`

SEPA-instant Einzelüberweisung, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**TYPE** = 'HKIPZ'

**VERSION** = 1

**account**

Kontoverbindung international

Type `fints.formals.KTI1`

**sepa\_descriptor**

SEPA Descriptor

Type `str`

**sepa\_pain\_message**

SEPA pain message

Type `bytes`

### 2.1.3 FinTS Segment Sequence

A message is a sequence of segments. The `SegmentSequence` object allows searching for segments by type and version, by default recursing into nested sequences.

**class** `fints.types.SegmentSequence(segments=None)`

A sequence of `FinTS3Segment` objects

**find\_segment\_first** (`*args, **kwargs`)

Finds the first matching segment.

Same parameters as `find_segments()`, but only returns the first match, or `None` if no match is found.

**find\_segment\_highest\_version** (*query=None, version=None, callback=None, recurse=True, default=None*)

Finds the highest matching segment.

Same parameters as `find_segments()`, but returns the match with the highest version, or default if no match is found.

**find\_segments** (*query=None, version=None, callback=None, recurse=True, throw=False*)

Yields an iterable of all matching segments.

#### Parameters

- **query** – Either a str or class specifying a segment type (such as ‘HNHBK’, or [HNHBK3](#)), or a list or tuple of strings or classes. If a list/tuple is specified, segments returning any matching type will be returned.
- **version** – Either an int specifying a segment version, or a list or tuple of ints. If a list/tuple is specified, segments returning any matching version will be returned.
- **callback** – A callable that will be given the segment as its sole argument and must return a boolean indicating whether to return this segment.
- **recurse** – If True (the default), recurse into `SegmentSequenceField` values, otherwise only look at segments in this `SegmentSequence`.
- **throw** – If True, a `FinTSNoResponseError` is thrown if no result is found. Defaults to False.

The match results of all given parameters will be AND-combined.

**print\_nested** (*stream=None, level=0, indent=' ', prefix="", first\_level\_indent=True, trailer="", print\_doc=True, first\_line\_suffix=""*)

**render\_bytes** () → bytes

## 2.1.4 Working with Segments

Objects of `FinTS3Segment` or a subclass can be created by calling their constructor. The constructor takes optional arguments for all fields of the class. Setting and getting fields and subfields works, and consumes and returns Python objects as appropriate:

```
>>> from fints.segments import HNHBK1
>>> s = HNHBK1()
>>> s
fints.segments.HNHBK1(header=fints.formals.SegmentHeader('HNHBK', None, 1), message_
↳number=None)
>>> s.header.number = 3
>>> s.header
fints.formals.SegmentHeader('HNHBK', 3, 1)
```

When setting a value, format and length restrictions will be checked, if possible:

```
>>> s.message_number = 'abc'
ValueError: invalid literal for int() with base 10: 'abc'
>>> s.message_number = 12345
ValueError: Value '12345' cannot be rendered: max_length=4 exceeded
```

The only exception is: Every field can be set to `None` in order to clear the field and make it unset, recursively. No checking is performed whether all fields that are required (or conditionally required) by the specification are set. For

convenience, an unset constructed field will still be filled with an instance of the field's value type, so that subfield accessing will always work, without encountering `None` values on the way.

```
>>> s.header = None
>>> s
fints.segments.HNHBS1(header=fints.formals.SegmentHeader(None, None, None), message_
↳number=None)
```

When calling the constructor with non-keyword arguments, fields are assigned in order, with the exception of `header` in *FintS3Segment* subclasses, which can only be given as a keyword argument. When no `header` argument is present, a *SegmentHeader* is automatically constructed with default values (and no number). It's generally not required to construct the `header` parameter manually.

```
>>> HNHBS1(42)
fints.segments.HNHBS1(header=fints.formals.SegmentHeader('HNHBS', None, 1), message_
↳number=42)
>>> HNHBS1(42, header=SegmentHeader('FOO'))
fints.segments.HNHBS1(header=fints.formals.SegmentHeader('FOO', None, None), message_
↳number=42)
```

Some segment fields have a variable number of values. These are always treated as a list, and minimum/maximum list length is obeyed. Setting a value beyond the end of the list results in an exception. Empty values are added to maintain the correct minimum number of values.

```
>>> from fints.segments import HIRMG2
>>> s = HIRMG2()
>>> s
fints.segments.HIRMG2(header=fints.formals.SegmentHeader('HIRMG', None, 2),
↳responses=[fints.formals.Response(code=None, reference_element=None, text=None)])
>>> s.responses[0].code = '0010'
>>> s.responses[1].code = '0100'
>>> s.print_nested()
fints.segments.HIRMG2(
    header = fints.formals.SegmentHeader('HIRMG', None, 2),
    responses = [
        fints.formals.Response(
            code = '0010',
            reference_element = None,
            text = None,
        ),
        fints.formals.Response(
            code = '0100',
            reference_element = None,
            text = None,
        ),
    ],
)
>>> HIRMG2(responses=[fints.formals.Response('2342')]).print_nested()
fints.segments.HIRMG2(
    header = fints.formals.SegmentHeader('HIRMG', None, 2),
    responses = [
        fints.formals.Response(
            code = '2342',
            reference_element = None,
            text = None,
        ),
    ],
)
```



## 2.1.5 Defining new Segment classes

### Base types

```
class fints.types.Field(length=None, min_length=None, max_length=None, count=None,
                        min_count=None, max_count=None, required=True, _d=None)

    render (value)

class fints.types.TypedField (type=None, *args, **kwargs)
class fints.types.ValueList (parent)
class fints.types.ContainerMeta
class fints.types.Container (*args, **kwargs)

    classmethod naive_parse (data)
    is_unset ()
```

### Field types

```
class fints.fields.DataElementField (*args, **kwargs)
class fints.fields.ContainerField (type=None, *args, **kwargs)
class fints.fields.DataElementGroupField (*args, **kwargs)
class fints.fields.GenericField (*args, **kwargs)

    type = None
class fints.fields.GenericGroupField (*args, **kwargs)

    type = None
class fints.fields.TextField (*args, **kwargs)

    type = 'txt'
class fints.fields.AlphanumericField (*args, **kwargs)

    type = 'an'
class fints.fields.DTAUSField (*args, **kwargs)

    type = 'dta'
class fints.fields.NumericField (*args, **kwargs)

    type = 'num'
class fints.fields.ZeroPaddedNumericField (*args, **kwargs)

    type = ''
```

```
class fints.fields.DigitsField(*args, **kwargs)

    type = 'dig'
class fints.fields.FloatField(*args, **kwargs)

    type = 'float'
class fints.fields.AmountField(*args, **kwargs)

    type = 'wrt'
class fints.fields.BinaryField(*args, **kwargs)

    type = 'bin'
class fints.fields.IDField(*args, **kwargs)

    type = 'id'
class fints.fields.BooleanField(*args, **kwargs)

    type = 'jn'
class fints.fields.CodeFieldMixin(enum=None, *args, **kwargs)
class fints.fields.CodeField(enum=None, *args, **kwargs)

    type = 'code'
class fints.fields.IntCodeField(enum=None, *args, **kwargs)

    type = ''
class fints.fields.CountryField(*args, **kwargs)

    type = 'ctr'
class fints.fields.CurrencyField(*args, **kwargs)

    type = 'cur'
class fints.fields.DateField(*args, **kwargs)

    type = 'dat'
class fints.fields.TimeField(*args, **kwargs)

    type = 'tim'
class fints.fields.PasswordField(*args, **kwargs)

    type = ''
```

```
class fints.fields.SegmentSequenceField(*args, **kwargs)
```

```
    type = 'sf'
```

### Constructed and helper types

```
class fints.formals.DataElementGroup(*args, **kwargs)
```

```
class fints.formals.SegmentHeader(*args, **kwargs)
    Segmentkopf
```

```
    type
        Segmentkennung
        Type str
```

```
    number
        Segmentnummer
        Type int
```

```
    version
        Segmentversion
        Type int
```

```
    reference
        Bezugssegment
        Type int
```

```
class fints.formals.ReferenceMessage(*args, **kwargs)
```

```
    dialog_id
        Type str
```

```
    message_number
        Type int
```

```
class fints.formals.SecurityMethod(*args, **kwargs)
```

```
    An enumeration.
```

```
    DDV = 'DDV'
```

```
    RAH = 'RAH'
```

```
    RDH = 'RDH'
```

```
    PIN = 'PIN'
```

```
class fints.formals.SecurityProfile(*args, **kwargs)
```

```
    Sicherheitsprofil
```

```
    security_method
        Sicherheitsverfahren
        Type fints.formals.SecurityMethod
```

```
    security_method_version
        Version des Sicherheitsverfahrens
        Type int
```

```
class fints.formals.IdentifiedRole(*args, **kwargs)
    An enumeration.

    MS = '1'
        Message Sender

    MR = '2'
        Message Receiver

class fints.formals.SecurityIdentificationDetails(*args, **kwargs)

    identified_role
        Type fints.formals.IdentifiedRole

    cid
        Type bytes

    identifier
        Type str

class fints.formals.DateTimeType(*args, **kwargs)
    An enumeration.

    STS = '1'
        Sicherheitszeitstempel

    CRT = '6'
        Certificate Revocation Time

class fints.formals.SecurityDateTime(*args, **kwargs)

    date_time_type
        Type fints.formals.DateTimeType

    date
        Type datetime.date

    time
        Type datetime.time

class fints.formals.UsageEncryption(*args, **kwargs)
    An enumeration.

    OSY = '2'
        Owner Symmetric

class fints.formals.OperationMode(*args, **kwargs)
    An enumeration.

    CBC = '2'
        Cipher Block Chaining

    ISO_9796_1 = '16'
        ISO 9796-1 (bei RDH)

    ISO_9796_2_RANDOM = '17'
        ISO 9796-2 mit Zufallszahl (bei RDH)
```

```

PKCS1V15 = '18'
    RSASSA-PKCS#1 V1.5 (bei RDH); RSAES-PKCS#1 V1.5 (bei RAH, RDH)

PSS = '19'
    RSASSA-PSS (bei RAH, RDH)

ZZZ = '999'
    Gegenseitig vereinbart (DDV: Retail-MAC)

class fints.formals.EncryptionAlgorithmCoded(*args, **kwargs)
    An enumeration.

TWOKEY3DES = '13'
    2-Key-Triple-DES

AES256 = '14'
    AES-256

class fints.formals.AlgorithmParameterName(*args, **kwargs)
    An enumeration.

KYE = '5'
    Symmetrischer Schlüssel, verschlüsselt mit symmetrischem Schlüssel

KYP = '6'
    Symmetrischer Schlüssel, verschlüsselt mit öffentlichem Schlüssel

class fints.formals.AlgorithmParameterIVName(*args, **kwargs)
    An enumeration.

IVC = '1'
    Initialization value, clear text

class fints.formals.EncryptionAlgorithm(*args, **kwargs)

    usage_encryption
        Type fints.formals.UsageEncryption

    operation_mode
        Type fints.formals.OperationMode

    encryption_algorithm
        Type fints.formals.EncryptionAlgorithmCoded

    algorithm_parameter_value
        Type bytes

    algorithm_parameter_name
        Type fints.formals.AlgorithmParameterName

    algorithm_parameter_iv_name
        Type fints.formals.AlgorithmParameterIVName

    algorithm_parameter_iv_value
        Type bytes

class fints.formals.HashAlgorithm(*args, **kwargs)

```

```
usage_hash
    Type str
hash_algorithm
    Type str
algorithm_parameter_name
    Type str
algorithm_parameter_value
    Type bytes
class fints.formals.SignatureAlgorithm(*args, **kwargs)

usage_signature
    Type str
signature_algorithm
    Type str
operation_mode
    Type str
class fints.formals.BankIdentifier(*args, **kwargs)

COUNTRY_ALPHA_TO_NUMERIC = {'AT': '040', 'BE': '056', 'BG': '100', 'CA': '124', 'CH':
COUNTRY_NUMERIC_TO_ALPHA = {'040': 'AT', '056': 'BE', '100': 'BG', '124': 'CA', '1
country_identifier
    Type str
bank_code
    Type str
class fints.formals.KeyType(*args, **kwargs)
    Schlüsselart
    D = 'D'
        Schlüssel zur Erzeugung digitaler Signaturen
    S = 'S'
        Signierschlüssel
    V = 'V'
        Chiffrierschlüssel
class fints.formals.KeyName(*args, **kwargs)

bank_identifier
    Type fints.formals.BankIdentifier
user_id
    Type str
```

**key\_type**  
Schlüsselart  
  
Type `fints.formals.KeyType`

**key\_number**  
  
Type `int`

**key\_version**  
  
Type `int`

**class** `fints.formals.Certificate(*args, **kwargs)`

**certificate\_type**  
  
Type `str`

**certificate\_content**  
  
Type `bytes`

**class** `fints.formals.UserDefinedSignature(*args, **kwargs)`

**pin**  
  
Type `fints.utils.Password`

**tan**  
  
Type `str`

**class** `fints.formals.Response(*args, **kwargs)`

**code**  
  
Type `str`

**reference\_element**  
  
Type `str`

**text**  
  
Type `str`

**parameters**  
  
Type `str`

**class** `fints.formals.Amount1(*args, **kwargs)`

Betrag

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**amount**  
Wert  
  
Type `decimal.Decimal`

**currency**  
Währung  
  
Type `str`

```
class fints.formals.AccountInformation(*args, **kwargs)
```

```
    account_number
```

```
        Type str
```

```
    subaccount_number
```

```
        Type str
```

```
    bank_identifier
```

```
        Type fints.formals.BankIdentifier
```

```
class fints.formals.AccountLimit(*args, **kwargs)
```

```
    limit_type
```

```
        Type str
```

```
    limit_amount
```

```
        Type fints.formals.Amount1
```

```
    limit_days
```

```
        Type int
```

```
class fints.formals.AllowedTransaction(*args, **kwargs)
```

```
    transaction
```

```
        Type str
```

```
    required_signatures
```

```
        Type int
```

```
    limit_type
```

```
        Type str
```

```
    limit_amount
```

```
        Type fints.formals.Amount1
```

```
    limit_days
```

```
        Type int
```

```
class fints.formals.TANTimeDialogAssociation(*args, **kwargs)
```

```
    An enumeration.
```

```
    NOT_ALLOWED = '1'
```

```
        TAN nicht zeitversetzt / dialogübergreifend erlaubt
```

```
    ALLOWED = '2'
```

```
        TAN zeitversetzt / dialogübergreifend erlaubt
```

```
    BOTH = '3'
```

```
        beide Verfahren unterstützt
```

```
    NOT_APPLICABLE = '4'
```

```
        nicht zutreffend
```



```

class fints.formals.AllowedFormat(*args, **kwargs)
    An enumeration.

    NUMERIC = '1'
        numerisch

    ALPHANUMERIC = '2'
        alfanumerisch

class fints.formals.TANListNumberRequired(*args, **kwargs)
    An enumeration.

    NO = '0'
        Nein

    YES = '2'
        Ja

class fints.formals.InitializationMode(*args, **kwargs)
    An enumeration.

    CLEARTEXT_PIN_NO_TAN = '00'
        Initialisierungsverfahren mit Klartext-PIN und ohne TAN

    ENCRYPTED_PIN_NO_TAN = '01'
        Schablone 01: Verschlüsselte PIN und ohne TAN

    MASK_02 = '02'
        Schablone 02: Reserviert, bei FinTS zur Zeit nicht verwendet

class fints.formals.DescriptionRequired(*args, **kwargs)
    An enumeration.

    MUST_NOT = '0'
        Bezeichnung des TAN-Mediums darf nicht angegeben werden

    MAY = '1'
        Bezeichnung des TAN-Mediums kann angegeben werden

    MUST = '2'
        Bezeichnung des TAN-Mediums muss angegeben werden

class fints.formals.SMSChargeAccountRequired(*args, **kwargs)
    An enumeration.

    MUST_NOT = '0'
        SMS-Abbuchungskonto darf nicht angegeben werden

    MAY = '1'
        SMS-Abbuchungskonto kann angegeben werden

    MUST = '2'
        SMS-Abbuchungskonto muss angegeben werden

class fints.formals.PrincipalAccountRequired(*args, **kwargs)
    An enumeration.

    MUST_NOT = '0'
        Auftraggeberkonto darf nicht angegeben werden

    MUST = '2'
        Auftraggeberkonto muss angegeben werden, wenn im Geschäftsvorfall enthalten

```

```
class fints.formals.TaskHashAlgorithm(*args, **kwargs)
    An enumeration.

    NONE = '0'
        Auftrags-Hashwert nicht unterstützt

    RIPEMD_160 = '1'
        RIPEMD-160

    SHA_1 = '2'
        SHA-1

class fints.formals.TwoStepParametersCommon(*args, **kwargs)

    VERSION
        TAN mechanism version

    security_function
        Sicherheitsfunktion kodiert

        Type str

    tan_process
        TAN-Prozess

        Type str

    tech_id
        Technische Identifikation TAN-Verfahren

        Type str

class fints.formals.TwoStepParameters1(*args, **kwargs)

    name
        Name des Zwei-Schritt-Verfahrens

        Type str

    max_length_input
        Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

        Type int

    allowed_format
        Erlaubtes Format im Zwei-Schritt-Verfahren

        Type fints.formals.AllowedFormat

    text_return_value
        Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

        Type str

    max_length_return_value
        Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

        Type int

    number_of_supported_lists
        Anzahl unterstützter aktiver TAN-Listen

        Type int
```

**multiple\_tans\_allowed**

Mehrfach-TAN erlaubt

Type `bool`

**tan\_time\_delayed\_allowed**

TAN zeitversetzt/dialogübergreifend erlaubt

Type `bool`

**class** `fints.formals.TwoStepParameters2` (\*args, \*\*kwargs)

**name**

Name des Zwei-Schritt-Verfahrens

Type `str`

**max\_length\_input**

Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

Type `int`

**allowed\_format**

Erlaubtes Format im Zwei-Schritt-Verfahren

Type `fints.formals.AllowedFormat`

**text\_return\_value**

Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

Type `str`

**max\_length\_return\_value**

Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

Type `int`

**number\_of\_supported\_lists**

Anzahl unterstützter aktiver TAN-Listen

Type `int`

**multiple\_tans\_allowed**

Mehrfach-TAN erlaubt

Type `bool`

**tan\_time\_dialog\_association**

TAN Zeit- und Dialogbezug

Type `fints.formals.TANTimeDialogAssociation`

**tan\_list\_number\_required**

TAN-Listennummer erforderlich

Type `fints.formals.TANListNumberRequired`

**cancel\_allowed**

Auftragsstorno erlaubt

Type `bool`

**challenge\_class\_required**

Challenge-Klasse erforderlich

Type `bool`

**challenge\_value\_required**

Challenge-Betrag erforderlich

Type `bool`

**class** `fints.formals.TwoStepParameters3(*args, **kwargs)`

**name**

Name des Zwei-Schritt-Verfahrens

Type `str`

**max\_length\_input**

Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

Type `int`

**allowed\_format**

Erlaubtes Format im Zwei-Schritt-Verfahren

Type `fints.formals.AllowedFormat`

**text\_return\_value**

Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

Type `str`

**max\_length\_return\_value**

Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

Type `int`

**number\_of\_supported\_lists**

Anzahl unterstützter aktiver TAN-Listen

Type `int`

**multiple\_tans\_allowed**

Mehrfach-TAN erlaubt

Type `bool`

**tan\_time\_dialog\_association**

TAN Zeit- und Dialogbezug

Type `fints.formals.TANTimeDialogAssociation`

**tan\_list\_number\_required**

TAN-Listennummer erforderlich

Type `fints.formals.TANListNumberRequired`

**cancel\_allowed**

Auftragsstorno erlaubt

Type `bool`

**challenge\_class\_required**

Challenge-Klasse erforderlich

Type `bool`

**challenge\_value\_required**

Challenge-Betrag erforderlich

Type `bool`

**initialization\_mode**

Initialisierungsmodus

Type *fints.formals.InitializationMode*

**description\_required**

Bezeichnung des TAN-Medium erforderlich

Type *fints.formals.DescriptionRequired*

**supported\_media\_number**

Anzahl unterstützter aktiver TAN-Medien

Type *int*

**class** *fints.formals.TwoStepParameters4* (\*args, \*\*kwargs)

**zka\_id**

ZKA TAN-Verfahren

Type *str*

**zka\_version**

Version ZKA TAN-Verfahren

Type *str*

**name**

Name des Zwei-Schritt-Verfahrens

Type *str*

**max\_length\_input**

Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

Type *int*

**allowed\_format**

Erlaubtes Format im Zwei-Schritt-Verfahren

Type *fints.formals.AllowedFormat*

**text\_return\_value**

Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

Type *str*

**max\_length\_return\_value**

Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

Type *int*

**number\_of\_supported\_lists**

Anzahl unterstützter aktiver TAN-Listen

Type *int*

**multiple\_tans\_allowed**

Mehrfach-TAN erlaubt

Type *bool*

**tan\_time\_dialog\_association**

TAN Zeit- und Dialogbezug

Type *fints.formals.TANTimeDialogAssociation*

**tan\_list\_number\_required**  
TAN-Listennummer erforderlich  
  
Type *fints.formals.TANListNumberRequired*

**cancel\_allowed**  
Auftragsstorno erlaubt  
  
Type *bool*

**sms\_charge\_account\_required**  
SMS-Abbuchungskonto erforderlich  
  
Type *bool*

**challenge\_class\_required**  
Challenge-Klasse erforderlich  
  
Type *bool*

**challenge\_value\_required**  
Challenge-Betrag erforderlich  
  
Type *bool*

**challenge\_structured**  
Challenge strukturiert  
  
Type *bool*

**initialization\_mode**  
Initialisierungsmodus  
  
Type *fints.formals.InitializationMode*

**description\_required**  
Bezeichnung des TAN-Medium erforderlich  
  
Type *fints.formals.DescriptionRequired*

**supported\_media\_number**  
Anzahl unterstützter aktiver TAN-Medien  
  
Type *int*

**class** *fints.formals.TwoStepParameters5*(\*args, \*\*kwargs)

**zka\_id**  
ZKA TAN-Verfahren  
  
Type *str*

**zka\_version**  
Version ZKA TAN-Verfahren  
  
Type *str*

**name**  
Name des Zwei-Schritt-Verfahrens  
  
Type *str*

**max\_length\_input**  
Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren  
  
Type *int*

**allowed\_format**  
Erlaubtes Format im Zwei-Schritt-Verfahren  
Type `fints.formals.AllowedFormat`

**text\_return\_value**  
Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren  
Type `str`

**max\_length\_return\_value**  
Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren  
Type `int`

**number\_of\_supported\_lists**  
Anzahl unterstützter aktiver TAN-Listen  
Type `int`

**multiple\_tans\_allowed**  
Mehrfach-TAN erlaubt  
Type `bool`

**tan\_time\_dialog\_association**  
TAN Zeit- und Dialogbezug  
Type `fints.formals.TANTimeDialogAssociation`

**tan\_list\_number\_required**  
TAN-Listennummer erforderlich  
Type `fints.formals.TANListNumberRequired`

**cancel\_allowed**  
Auftragsstorno erlaubt  
Type `bool`

**sms\_charge\_account\_required**  
SMS-Abbuchungskonto erforderlich  
Type `fints.formals.SMSChargeAccountRequired`

**principal\_account\_required**  
Auftraggeberkonto erforderlich  
Type `fints.formals.PrincipalAccountRequired`

**challenge\_class\_required**  
Challenge-Klasse erforderlich  
Type `bool`

**challenge\_structured**  
Challenge strukturiert  
Type `bool`

**initialization\_mode**  
Initialisierungsmodus  
Type `fints.formals.InitializationMode`

**description\_required**  
Bezeichnung des TAN-Medium erforderlich

Type *fints.formals.DescriptionRequired*

**supported\_media\_number**

Anzahl unterstützter aktiver TAN-Medien

Type *int*

**class** *fints.formals.TwoStepParameters6*(\*args, \*\*kwargs)

**zka\_id**

ZKA TAN-Verfahren

Type *str*

**zka\_version**

Version ZKA TAN-Verfahren

Type *str*

**name**

Name des Zwei-Schritt-Verfahrens

Type *str*

**max\_length\_input**

Maximale Länge des Eingabewertes im Zwei-Schritt-Verfahren

Type *int*

**allowed\_format**

Erlaubtes Format im Zwei-Schritt-Verfahren

Type *fints.formals.AllowedFormat*

**text\_return\_value**

Text zur Belegung des Rückgabewertes im Zwei-Schritt-Verfahren

Type *str*

**max\_length\_return\_value**

Maximale Länge des Rückgabewertes im Zwei-Schritt-Verfahren

Type *int*

**multiple\_tans\_allowed**

Mehrfach-TAN erlaubt

Type *bool*

**tan\_time\_dialog\_association**

TAN Zeit- und Dialogbezug

Type *fints.formals.TANTimeDialogAssociation*

**cancel\_allowed**

Auftragsstorno erlaubt

Type *bool*

**sms\_charge\_account\_required**

SMS-Abbuchungskonto erforderlich

Type *fints.formals.SMSChargeAccountRequired*

**principal\_account\_required**

Auftraggeberkonto erforderlich



```

        Type fints.formals.PrincipalAccountRequired

challenge_class_required
    Challenge-Klasse erforderlich

        Type bool

challenge_structured
    Challenge strukturiert

        Type bool

initialization_mode
    Initialisierungsmodus

        Type fints.formals.InitializationMode

description_required
    Bezeichnung des TAN-Medium erforderlich

        Type fints.formals.DescriptionRequired

response_hhd_uc_required
    Antwort HHD_UC erforderlich

        Type bool

supported_media_number
    Anzahl unterstützter aktiver TAN-Medien

        Type int

class fints.formals.ParameterTwostepCommon(*args, **kwargs)

    onestep_method_allowed

        Type bool

    multiple_tasks_allowed

        Type bool

    task_hash_algorithm
    Auftrags-Hashwertverfahren

        Type fints.formals.TaskHashAlgorithm

class fints.formals.ParameterTwostepTAN1(*args, **kwargs)

    security_profile_bank_signature

        Type str

    twostep_parameters

        Type fints.formals.TwoStepParameters1

class fints.formals.ParameterTwostepTAN2(*args, **kwargs)

    twostep_parameters

        Type fints.formals.TwoStepParameters2

class fints.formals.ParameterTwostepTAN3(*args, **kwargs)

```

**twostep\_parameters**

    Type *fints.formals.TwoStepParameters3*

**class** *fints.formals.ParameterTwostepTAN4*(\*args, \*\*kwargs)

**twostep\_parameters**

    Type *fints.formals.TwoStepParameters4*

**class** *fints.formals.ParameterTwostepTAN5*(\*args, \*\*kwargs)

**twostep\_parameters**

    Type *fints.formals.TwoStepParameters5*

**class** *fints.formals.ParameterTwostepTAN6*(\*args, \*\*kwargs)

**twostep\_parameters**

    Type *fints.formals.TwoStepParameters6*

**class** *fints.formals.TransactionTanRequired*(\*args, \*\*kwargs)

**transaction**

    Type *str*

**tan\_required**

    Type *bool*

**class** *fints.formals.ParameterPinTan*(\*args, \*\*kwargs)

**min\_pin\_length**

    Type *int*

**max\_pin\_length**

    Type *int*

**max\_tan\_length**

    Type *int*

**user\_id\_field\_text**

    Type *str*

**customer\_id\_field\_text**

    Type *str*

**transaction\_tans\_required**

    Type *fints.formals.TransactionTanRequired*

**class** *fints.formals.Language2*(\*args, \*\*kwargs)

    Dialogsprache

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

```

DEFAULT = '0'
    Standard

DE = '1'
    Deutsch, 'de', Subset Deutsch, Codeset 1 (Latin 1)

EN = '2'
    Englisch, 'en', Subset Englisch, Codeset 1 (Latin 1)

FR = '3'
    Französisch, 'fr', Subset Französisch, Codeset 1 (Latin 1)

class fints.formals.SupportedLanguages2(*args, **kwargs)

    languages
        Type fints.formals.Language2

class fints.formals.SupportedHBCIVersions2(*args, **kwargs)

    versions
        Type str

class fints.formals.KTZ1(*args, **kwargs)
    Kontoverbindung ZV international, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    is_sepa
        Kontoverwendung SEPA
        Type bool

    iban
        IBAN
        Type str

    bic
        BIC
        Type str

    account_number
        Konto-/Depotnummer
        Type str

    subaccount_number
        Unterkontomerkmal
        Type str

    bank_identifier
        Kreditinstitutskennung
        Type fints.formals.BankIdentifier

    as_sepa_account()

    classmethod from_sepa_account(acc)

```

```
class fints.formals.KT11(*args, **kwargs)
    Kontoverbindung international, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    iban
        IBAN

        Type str

    bic
        BIC

        Type str

    account_number
        Konto-/Depotnummer

        Type str

    subaccount_number
        Unterkontomerkmal

        Type str

    bank_identifier
        Kreditinstitutskennung

        Type fints.formals.BankIdentifier

    classmethod from_sepa_account(acc)

class fints.formals.Account2(*args, **kwargs)
    Kontoverbindung, version 2

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    account_number
        Konto-/Depotnummer

        Type str

    subaccount_number
        Unterkontomerkmal

        Type str

    country_identifier
        Länderkennzeichen

        Type str

    bank_code
        Kreditinstitutscode

        Type str

    classmethod from_sepa_account(acc)

class fints.formals.Account3(*args, **kwargs)
    Kontoverbindung, version 3

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle
```

**account\_number**  
Konto-/Depotnummer

Type `str`

**subaccount\_number**  
Unterkontomerkmal

Type `str`

**bank\_identifier**  
Kreditinstitutskennung

Type `fints.formals.BankIdentifier`

**classmethod from\_sepa\_account** (*acc*)

**class** `fints.formals.SecurityRole` (\*args, \*\*kwargs)

Rolle des Sicherheitslieferanten, kodiert, version 2

Kodierte Information über das Verhältnis desjenigen, der bezüglich der zu sichernden Nachricht die Sicherheit gewährleistet. Die Wahl ist von der bankfachlichen Auslegung der Signatur, respektive vom vertraglichen Zustand zwischen Kunde und Kreditinstitut abhängig.

Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

**ISS** = '1'  
Erfasser, Erstschrift

**CON** = '3'  
Unterstützer, Zweitschrift

**WIT** = '4'  
Zeuge/Übermittler, nicht Erfasser

**class** `fints.formals.CompressionFunction` (\*args, \*\*kwargs)

Komprimierungsfunktion, version 2

Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

**NULL** = '0'  
Keine Kompression

**LZW** = '1'  
Lempel, Ziv, Welch

**COM** = '2'  
Optimized LZW

**LZSS** = '3'  
Lempel, Ziv

**LZHuf** = '4'  
LZ + Huffman Coding

**ZIP** = '5'  
PKZIP

**GZIP** = '6'  
deflate (<http://www.gzip.org/zlib>)

**BZIP2** = '7'  
bzip2 (<http://sourceware.cygnum.com/bzip2/>)

**ZZZ = '999'**

Gegenseitig vereinbart

**class** fintns.formals.**SecurityApplicationArea**(\*args, \*\*kwargs)

Bereich der Sicherheitsapplikation, kodiert, version 2

Informationen darüber, welche Daten vom kryptographischen Prozess verarbeitet werden.

Source: FinTS Financial Transaction Services, Sicherheitsverfahren HBCI

**SHM = '1'**

Signaturkopf und HBCI-Nutzdaten

**SHT = '2'**

Von Signaturkopf bis Signaturabschluss

**class** fintns.formals.**SecurityClass**(\*args, \*\*kwargs)

Sicherheitsklasse, version 1

Die Sicherheitsklasse gibt für jede Signatur den erforderlichen Sicherheitsdienst an.

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**NONE = 0**

Kein Sicherheitsdienst erforderlich

**AUTH = 1**

Sicherheitsdienst 'Authentikation'

**AUTH\_ADV = 2**

Sicherheitsdienst 'Authentikation' mit fortgeschrittener elektronischer Signatur, optionaler Zertifikatsprüfung

**NON\_REPUD = 3**

Sicherheitsdienst 'Non-Repudiation' mit fortgeschrittener elektronischer Signatur, optionaler Zertifikatsprüfung

**NON\_REPUD\_QUAL = 4**

Sicherheitsdienst 'Non-Repudiation' mit fortgeschrittener bzw. qualifizierter elektronischer Signatur, zwingende Zertifikatsprüfung

**class** fintns.formals.**UPDUsage**(\*args, \*\*kwargs)

UPD-Verwendung, version 2

Kennzeichen dafür, wie diejenigen Geschäftsvorfälle zu interpretieren sind, die bei der Beschreibung der Kontoinformationen nicht unter den erlaubten Geschäftsvorfällen aufgeführt sind.

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**UPD\_CONCLUSIVE = '0'**

Die nicht aufgeführten Geschäftsvorfälle sind gesperrt

**UPD\_INCONCLUSIVE = '1'**

Bei nicht aufgeführten Geschäftsvorfällen ist keine Aussage möglich, ob diese erlaubt oder gesperrt sind

**class** fintns.formals.**SystemIDStatus**(\*args, \*\*kwargs)

Kundensystem-Status, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**ID\_UNNECESSARY = '0'**

Kundensystem-ID wird nicht benötigt

**ID\_NECESSARY = '1'**

Kundensystem-ID wird benötigt

```

class fints.formals.SynchronizationMode(*args, **kwargs)
    Synchronisierungsmodus, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

    NEW_SYSTEM_ID = '0'
        Neue Kundensystem-ID zurückmelden

    LAST_MESSAGE = '1'
        Letzte verarbeitete Nachrichtennummer zurückmelden

    SIGNATURE_ID = '2'
        Signatur-ID zurückmelden

class fints.formals.CreditDebit2(*args, **kwargs)
    Soll-Haben-Kennzeichen, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    CREDIT = 'C'
        Haben

    DEBIT = 'D'
        Soll

class fints.formals.Balance1(*args, **kwargs)
    Saldo, version 1

    Source: HBCI Homebanking-Computer-Interface, Schnittstellenspezifikation

    credit_debit
        Soll-Haben-Kennzeichen

        Type fints.formals.CreditDebit2

    amount
        Wert

        Type decimal.Decimal

    currency
        Währung

        Type str

    date
        Datum

        Type datetime.date

    time
        Uhrzeit

        Type datetime.time

    as_mt940_Balance()

class fints.formals.Balance2(*args, **kwargs)
    Saldo, version 2

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    credit_debit
        Soll-Haben-Kennzeichen

```

```
    Type fints.formals.CreditDebit2

amount
    Betrag

    Type fints.formals.Amount1

date
    Datum

    Type datetime.date

time
    Uhrzeit

    Type datetime.time

as_mt940_Balance()

class fints.formals.Timestamp1(*args, **kwargs)
    Zeitstempel

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    date
        Datum

        Type datetime.date

    time
        Uhrzeit

        Type datetime.time

class fints.formals.TANMediaType2(*args, **kwargs)
    TAN-Medium-Art

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

    ALL = '0'
        Alle

    ACTIVE = '1'
        Aktiv

    AVAILABLE = '2'
        Verfügbar

class fints.formals.TANMediaClass3(*args, **kwargs)
    TAN-Medium-Klasse, version 3

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

    ALL = 'A'
        Alle Medien

    LIST = 'L'
        Liste

    GENERATOR = 'G'
        TAN-Generator

    MOBILE = 'M'
        Mobiltelefon mit mobileTAN
```



```

SECODER = 'S'
    Secoder

class fints.formals.TANMediaClass4(*args, **kwargs)
    TAN-Medium-Klasse, version 4

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

    ALL = 'A'
        Alle Medien

    LIST = 'L'
        Liste

    GENERATOR = 'G'
        TAN-Generator

    MOBILE = 'M'
        Mobiltelefon mit mobileTAN

    SECODER = 'S'
        Secoder

    BILATERAL = 'B'
        Bilateral vereinbart

class fints.formals.TANMediumStatus(*args, **kwargs)
    Status

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

    ACTIVE = '1'
        Aktiv

    AVAILABLE = '2'
        Verfügbar

    ACTIVE_SUCCESSOR = '3'
        Aktiv Folgekarte

    AVAILABLE_SUCCESSOR = '4'
        Verfügbar Folgekarte

class fints.formals.TANMedia4(*args, **kwargs)
    TAN-Medium-Liste, version 4

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

    tan_medium_class
        TAN-Medium-Klasse

        Type fints.formals.TANMediaClass3

    status
        Status

        Type fints.formals.TANMediumStatus

    card_number
        Kartennummer

        Type str

    card_sequence
        Kartenfolgennummer

```

```
    Type str
card_type
    Kartenart
    Type int
account
    Kontonummer Auftraggeber
    Type fints.formals.Account3
valid_from
    Gültig ab
    Type datetime.date
valid_until
    Gültig bis
    Type datetime.date
tan_list_number
    TAN-Listennummer
    Type str
tan_medium_name
    Bezeichnung des TAN-Mediums
    Type str
mobile_number_masked
    Mobiltelefonnummer, verschleiert
    Type str
mobile_number
    Mobiltelefonnummer
    Type str
sms_charge_account
    SMS-Abbuchungskonto
    Type fints.formals.KTI1
number_free_tans
    Anzahl freie TANs
    Type int
last_use
    Letzte Benutzung
    Type datetime.date
active_since
    Freigeschaltet am
    Type datetime.date
class fints.formals.TANMedia5(*args, **kwargs)
    TAN-Medium-Liste, version 5
    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN
```

**tan\_medium\_class**  
TAN-Medium-Klasse  
  
Type `fints.formals.TANMediaClass4`

**status**  
Status  
  
Type `fints.formals.TANMediumStatus`

**security\_function**  
Sicherheitsfunktion, kodiert  
  
Type `int`

**card\_number**  
Kartenummer  
  
Type `str`

**card\_sequence**  
Kartenfolgenummer  
  
Type `str`

**card\_type**  
Kartenart  
  
Type `int`

**account**  
Kontonummer Auftraggeber  
  
Type `fints.formals.Account3`

**valid\_from**  
Gültig ab  
  
Type `datetime.date`

**valid\_until**  
Gültig bis  
  
Type `datetime.date`

**tan\_list\_number**  
TAN-Listennummer  
  
Type `str`

**tan\_medium\_name**  
Bezeichnung des TAN-Mediums  
  
Type `str`

**mobile\_number\_masked**  
Mobiltelefonnummer, verschleiert  
  
Type `str`

**mobile\_number**  
Mobiltelefonnummer  
  
Type `str`

**sms\_charge\_account**  
SMS-Abbuchungskonto

**Type** `fints.formals.KTI1`

**number\_free\_tans**  
Anzahl freie TANs

**Type** `int`

**last\_use**  
Letzte Benutzung

**Type** `datetime.date`

**active\_since**  
Freigeschaltet am

**Type** `datetime.date`

**class** `fints.formals.TANUsageOption(*args, **kwargs)`  
TAN-Einsatzoption

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**ALL\_ACTIVE** = '0'  
Kunde kann alle "aktiven" Medien parallel nutzen

**EXACTLY\_ONE** = '1'  
Kunde kann genau ein Medium zu einer Zeit nutzen

**MOBILE\_AND\_GENERATOR** = '2'  
Kunde kann ein Mobiltelefon und einen TAN-Generator parallel nutzen

**class** `fints.formals.ParameterChallengeClass(*args, **kwargs)`  
Parameter Challenge-Klasse

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**parameters**

**Type** `str`

**class** `fints.formals.ResponseHHDUC(*args, **kwargs)`  
Antwort HHD\_UC

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**atc**  
ATC

**Type** `str`

**ac**  
Application Cryptogram AC

**Type** `bytes`

**ef\_id\_data**  
EF\_ID Data

**Type** `bytes`

**cvr**  
CVR

**Type** `bytes`

**version\_info\_chiptan**  
Versionsinfo der chipTAN-Applikation

**Type** `bytes`

**class** `fints.formals.ChallengeValidUntil(*args, **kwargs)`

Gültigkeitsdatum und -uhrzeit für Challenge

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Sicherheitsverfahren PIN/TAN

**date**

Datum

**Type** `datetime.date`

**time**

Uhrzeit

**Type** `datetime.time`

**class** `fints.formals.BatchTransferParameter1(*args, **kwargs)`

Parameter SEPA-Sammelüberweisung, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**max\_transfer\_count**

Maximale Anzahl CreditTransferTransactionInformation

**Type** `int`

**sum\_amount\_required**

Summenfeld benötigt

**Type** `bool`

**single\_booking\_allowed**

Einzelbuchung erlaubt

**Type** `bool`

**class** `fints.formals.ServiceType2(*args, **kwargs)`

An enumeration.

**T\_ONLINE = 1**

T-Online

**TCP\_IP = 2**

TCP/IP (Protokollstack SLIP/PPP)

**HTTPS = 3**

https

**class** `fints.formals.CommunicationParameter2(*args, **kwargs)`

Kommunikationsparameter, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Formals

**service\_type**

Kommunikationsdienst

**Type** `fints.formals.ServiceType2`

**address**

Kommunikationsadresse

**Type** `str`

**address\_adjunct**

Kommunikationsadresszusatz

Type `str`

**filter\_function**

Filterfunktion

Type `str`

**filter\_function\_version**

Version der Filterfunktion

Type `int`

**class** `fints.formals.ScheduledDebitParameter1(*args, **kwargs)`

Parameter terminierte SEPA-Einzellastschrift einreichen, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**min\_advance\_notice\_FNAL\_RCUR**

Minimale Vorlaufzeit FNAL/RCUR

Type `int`

**max\_advance\_notice\_FNAL\_RCUR**

Maximale Vorlaufzeit FNAL/RCUR

Type `int`

**min\_advance\_notice\_FRST\_OOFF**

Minimale Vorlaufzeit FRST/OOFF

Type `int`

**max\_advance\_notice\_FRST\_OOFF**

Maximale Vorlaufzeit FRST/OOFF

Type `int`

**class** `fints.formals.ScheduledDebitParameter2(*args, **kwargs)`

Parameter terminierte SEPA-Einzellastschrift einreichen, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**min\_advance\_notice**

Minimale Vorlaufzeit SEPA-Lastschrift

Type `str`

**max\_advance\_notice**

Maximale Vorlaufzeit SEPA-Lastschrift

Type `str`

**allowed\_purpose\_codes**

Zulässige purpose codes

Type `str`

**supported\_sepa\_formats**

Unterstützte SEPA-Datenformate

Type `str`

```
class fints.formals.ScheduledBatchDebitParameter1(*args, **kwargs)
```

Parameter terminierte SEPA-Sammellastschrift einreichen, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**min\_advance\_notice\_FNAL\_RCUR**

Minimale Vorlaufzeit FNAL/RCUR

Type `int`

**max\_advance\_notice\_FNAL\_RCUR**

Maximale Vorlaufzeit FNAL/RCUR

Type `int`

**min\_advance\_notice\_FRST\_OOFF**

Minimale Vorlaufzeit FRST/OOFF

Type `int`

**max\_advance\_notice\_FRST\_OOFF**

Maximale Vorlaufzeit FRST/OOFF

Type `int`

**max\_debit\_count**

Maximale Anzahl DirectDebitTransfer TransactionInformation

Type `int`

**sum\_amount\_required**

Summenfeld benötigt

Type `bool`

**single\_booking\_allowed**

Einzelbuchung erlaubt

Type `bool`

```
class fints.formals.ScheduledBatchDebitParameter2(*args, **kwargs)
```

Parameter terminierte SEPA-Sammellastschrift einreichen, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**min\_advance\_notice**

Minimale Vorlaufzeit SEPA-Lastschrift

Type `str`

**max\_advance\_notice**

Maximale Vorlaufzeit SEPA-Lastschrift

Type `str`

**max\_debit\_count**

Maximale Anzahl DirectDebitTransfer TransactionInformation

Type `int`

**sum\_amount\_required**

Summenfeld benötigt

Type `bool`

**single\_booking\_allowed**

Einzelbuchung erlaubt

Type `bool`

**allowed\_purpose\_codes**

Zulässige purpose codes

Type `str`

**supported\_sepa\_formats**

Unterstützte SEPA-Datenformate

Type `str`

**class** `fints.formals.ScheduledCOR1DebitParameter1(*args, **kwargs)`

Parameter terminierte SEPA-COR1-Einzellastschrift, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**min\_advance\_notice\_FNAL\_RCUR**

Minimale Vorlaufzeit FNAL/RCUR

Type `int`

**max\_advance\_notice\_FNAL\_RCUR**

Maximale Vorlaufzeit FNAL/RCUR

Type `int`

**min\_advance\_notice\_FRST\_OOFF**

Minimale Vorlaufzeit FRST/OOFF

Type `int`

**max\_advance\_notice\_FRST\_OOFF**

Maximale Vorlaufzeit FRST/OOFF

Type `int`

**allowed\_purpose\_codes**

Zulässige purpose codes

Type `str`

**supported\_sepa\_formats**

Unterstützte SEPA-Datenformate

Type `str`

**class** `fints.formals.ScheduledCOR1BatchDebitParameter1(*args, **kwargs)`

Parameter terminierte SEPA-COR1-Sammellastschrift, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**max\_debit\_count**

Maximale Anzahl DirectDebitTransfer TransactionInformation

Type `int`

**sum\_amount\_required**

Summenfeld benötigt

Type `bool`



**single\_booking\_allowed**

Einzelbuchung erlaubt

Type `bool`

**min\_advance\_notice\_FNAL\_RCUR**

Minimale Vorlaufzeit FNAL/RCUR

Type `int`

**max\_advance\_notice\_FNAL\_RCUR**

Maximale Vorlaufzeit FNAL/RCUR

Type `int`

**min\_advance\_notice\_FRST\_OOFF**

Minimale Vorlaufzeit FRST/OOFF

Type `int`

**max\_advance\_notice\_FRST\_OOFF**

Maximale Vorlaufzeit FRST/OOFF

Type `int`

**allowed\_purpose\_codes**

Zulässige purpose codes

Type `str`

**supported\_sepa\_formats**

Unterstützte SEPA-Datenformate

Type `str`

**class** `fints.formals.SupportedSEPA PainMessages1` (\*args, \*\*kwargs)

Unterstützte SEPA pain messages, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**sepa\_descriptors**

SEPA Descriptor

Type `str`

**class** `fints.formals.QueryScheduledDebitParameter1` (\*args, \*\*kwargs)

Parameter Bestand terminierter SEPA-Einzellastschriften, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**date\_range\_allowed**

Zeitraum möglich

Type `bool`

**max\_number\_responses\_allowed**

Eingabe Anzahl Einträge erlaubt

Type `bool`

**class** `fints.formals.QueryScheduledDebitParameter2` (\*args, \*\*kwargs)

Parameter Bestand terminierter SEPA-Einzellastschriften, version 2

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**max\_number\_responses\_allowed**

Eingabe Anzahl Einträge erlaubt

Type `bool`

**date\_range\_allowed**

Zeitraum möglich

Type `bool`

**supported\_sepa\_formats**

Unterstützte SEPA-Datenformate

Type `str`

**class** `fints.formals.QueryScheduledBatchDebitParameter1(*args, **kwargs)`

Parameter Bestand terminierter SEPA-Sammellastschriften, version 1

Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige Geschäftsvorfälle

**max\_number\_responses\_allowed**

Eingabe Anzahl Einträge erlaubt

Type `bool`

**date\_range\_allowed**

Zeitraum möglich

Type `bool`

**class** `fints.formals.QueryCreditCardStatements2(*args, **kwargs)`

Parameter Kreditkartenumsätze anfordern, version 2

Source: reverse engineered

**cutoff\_days**

Maximale Vorhaltezeit der Umsätze

Type `int`

**max\_number\_responses\_allowed**

Eingabe Anzahl Einträge erlaubt

Type `bool`

**date\_range\_allowed**

Zeitraum möglich

Type `bool`

**class** `fints.formals.SEPACode1(*args, **kwargs)`

An enumeration.

**REVERSAL** = '1'

Reversal

**REVOCATION** = '2'

Revocation

**DELETION** = '3'

Delete

```

class fints.formals.StatusSEPA_Task1(*args, **kwargs)
    An enumeration.

    PENDING = '1'
        In Terminierung

    DECLINED = '2'
        Abgelehnt von erster Inkassostelle

    IN_PROGRESS = '3'
        in Bearbeitung

    PROCESSED = '4'
        Creditoren-seitig verarbeitet, Buchung veranlasst

    REVOKED = '5'
        R-Transaktion wurde veranlasst

class fints.formals.GetSEPAAccountParameter1(*args, **kwargs)
    Parameter SEPA-Kontoverbindung anfordern, version 1

    Source: FinTS Financial Transaction Services, Schnittstellenspezifikation, Messages – Multibankfähige
    Geschäftsvorfälle

    single_account_query_allowed
        Einzelkontenabruf erlaubt

        Type bool

    national_account_allowed
        Nationale Kontoverbindung erlaubt

        Type bool

    structured_purpose_allowed
        Strukturierter Verwendungszweck erlaubt

        Type bool

    supported_sepa_formats
        Unterstützte SEPA-Datenformate

        Type str

class fints.formals.SupportedMessageTypes(*args, **kwargs)
    Unterstützte camt-Messages

    Source: Messages - Multibankfähige Geschäftsvorfälle (SEPA) - C.2.3.1.1.1

    expected_type
        Unterstützte camt-messages

        Type str

class fints.formals.BookedCamtStatements1(*args, **kwargs)
    Gebuchte camt-Umsätze

    Source: Messages - Multibankfähige Geschäftsvorfälle (SEPA)

    camt_statements
        camt-Umsätze gebucht

        Type bytes

```



### f

- `fints.fields`, 77
- `fints.formals`, 79
- `fints.segments.accounts`, 29
- `fints.segments.auth`, 30
- `fints.segments.bank`, 40
- `fints.segments.base`, 44
- `fints.segments.debit`, 44
- `fints.segments.depot`, 54
- `fints.segments.dialog`, 56
- `fints.segments.journal`, 57
- `fints.segments.message`, 59
- `fints.segments.saldo`, 63
- `fints.segments.statement`, 67
- `fints.segments.transfer`, 72
- `fints.types`, 77



## A

- ac* (*fints.formals.ResponseHHDUC* attribute), 104
- account* (*fints.formals.TANMedia4* attribute), 102
- account* (*fints.formals.TANMedia5* attribute), 103
- account* (*fints.segments.auth.HKTAN5* attribute), 38
- account* (*fints.segments.auth.HKTAN6* attribute), 39
- account* (*fints.segments.debit.BatchDebitBase* attribute), 44
- account* (*fints.segments.debit.HIDBS1* attribute), 45
- account* (*fints.segments.debit.HIDBS2* attribute), 46
- account* (*fints.segments.debit.HIDMB1* attribute), 47
- account* (*fints.segments.debit.HKDBS1* attribute), 51
- account* (*fints.segments.debit.HKDBS2* attribute), 51
- account* (*fints.segments.debit.HKDMB1* attribute), 52
- account* (*fints.segments.debit.HKDSC1* attribute), 53
- account* (*fints.segments.debit.HKDSE1* attribute), 53
- account* (*fints.segments.debit.HKDSE2* attribute), 54
- account* (*fints.segments.depot.HKWPD5* attribute), 55
- account* (*fints.segments.depot.HKWPD6* attribute), 55
- account* (*fints.segments.saldo.HISAL5* attribute), 63
- account* (*fints.segments.saldo.HISAL6* attribute), 64
- account* (*fints.segments.saldo.HISAL7* attribute), 65
- account* (*fints.segments.saldo.HKSAL5* attribute), 66
- account* (*fints.segments.saldo.HKSAL6* attribute), 66
- account* (*fints.segments.saldo.HKSAL7* attribute), 67
- account* (*fints.segments.statement.DKKKU2* attribute), 67
- account* (*fints.segments.statement.HICAZ1* attribute), 68
- account* (*fints.segments.statement.HKCAZ1* attribute), 70
- account* (*fints.segments.statement.HKKAZ5* attribute), 70
- account* (*fints.segments.statement.HKKAZ6* attribute), 71
- account* (*fints.segments.statement.HKKAZ7* attribute), 72
- account* (*fints.segments.transfer.HKCCM1* attribute), 73
- account* (*fints.segments.transfer.HKCCS1* attribute), 73
- account* (*fints.segments.transfer.HKIPM1* attribute), 73
- account* (*fints.segments.transfer.HKIPZ1* attribute), 74
- Account2* (class in *fints.formals*), 96
- Account3* (class in *fints.formals*), 96
- account\_currency* (*fints.segments.bank.HIUPD6* attribute), 42
- account\_information* (*fints.segments.bank.HIUPD6* attribute), 42
- account\_limit* (*fints.segments.bank.HIUPD6* attribute), 42
- account\_number* (*fints.formals.Account2* attribute), 96
- account\_number* (*fints.formals.Account3* attribute), 96
- account\_number* (*fints.formals.AccountInformation* attribute), 84
- account\_number* (*fints.formals.KTII* attribute), 96
- account\_number* (*fints.formals.KTZI* attribute), 95
- account\_product* (*fints.segments.saldo.HISAL5* attribute), 63
- account\_product* (*fints.segments.saldo.HISAL6* attribute), 64
- account\_product* (*fints.segments.saldo.HISAL7* attribute), 65
- account\_product\_name* (*fints.segments.bank.HIUPD6* attribute), 42
- account\_type* (*fints.segments.bank.HIUPD6* attribute), 43
- AccountInformation* (class in *fints.formals*), 83
- AccountLimit* (class in *fints.formals*), 84
- accounts* (*fints.segments.accounts.HISPA1* attribute), 30
- accounts* (*fints.segments.accounts.HKSPA1* attribute), 30
- ACTIVE* (*fints.formals.TANMediaType2* attribute), 100
- ACTIVE* (*fints.formals.TANMediumStatus* attribute), 101
- active\_since* (*fints.formals.TANMedia4* attribute),

102  
active\_since (*fints.formals.TANMedia5* attribute), 104  
ACTIVE\_SUCCESOR (*fints.formals.TANMediumStatus* attribute), 101  
address (*fints.formals.CommunicationParameter2* attribute), 105  
address\_adjunct (*fints.formals.CommunicationParameter2* attribute), 105  
AES256 (*fints.formals.EncryptionAlgorithmCoded* attribute), 81  
algorithm\_parameter\_iv\_name (*fints.formals.EncryptionAlgorithm* attribute), 81  
algorithm\_parameter\_iv\_value (*fints.formals.EncryptionAlgorithm* attribute), 81  
algorithm\_parameter\_name (*fints.formals.EncryptionAlgorithm* attribute), 81  
algorithm\_parameter\_name (*fints.formals.HashAlgorithm* attribute), 82  
algorithm\_parameter\_value (*fints.formals.EncryptionAlgorithm* attribute), 81  
algorithm\_parameter\_value (*fints.formals.HashAlgorithm* attribute), 82  
AlgorithmParameterIVName (class in *fints.formals*), 81  
AlgorithmParameterName (class in *fints.formals*), 81  
ALL (*fints.formals.TANMediaClass3* attribute), 100  
ALL (*fints.formals.TANMediaClass4* attribute), 101  
ALL (*fints.formals.TANMediaType2* attribute), 100  
all\_accounts (*fints.segments.saldo.HKSAL5* attribute), 66  
all\_accounts (*fints.segments.saldo.HKSAL6* attribute), 66  
all\_accounts (*fints.segments.saldo.HKSAL7* attribute), 67  
all\_accounts (*fints.segments.statement.HKCAZ1* attribute), 70  
all\_accounts (*fints.segments.statement.HKKAZ5* attribute), 70  
all\_accounts (*fints.segments.statement.HKKAZ6* attribute), 71  
all\_accounts (*fints.segments.statement.HKKAZ7* attribute), 72  
ALL\_ACTIVE (*fints.formals.TANUsageOption* attribute), 104  
ALLOWED (*fints.formals.TANTimeDialogAssociation* attribute), 84  
allowed\_format (*fints.formals.TwoStepParameters1* attribute), 86  
allowed\_format (*fints.formals.TwoStepParameters2* attribute), 87  
allowed\_format (*fints.formals.TwoStepParameters3* attribute), 88  
allowed\_format (*fints.formals.TwoStepParameters4* attribute), 89  
allowed\_format (*fints.formals.TwoStepParameters5* attribute), 90  
allowed\_format (*fints.formals.TwoStepParameters6* attribute), 92  
allowed\_purpose\_codes (*fints.formals.ScheduledBatchDebitParameter2* attribute), 108  
allowed\_purpose\_codes (*fints.formals.ScheduledCOR1BatchDebitParameter1* attribute), 109  
allowed\_purpose\_codes (*fints.formals.ScheduledCOR1DebitParameter1* attribute), 108  
allowed\_purpose\_codes (*fints.formals.ScheduledDebitParameter2* attribute), 106  
allowed\_transactions (*fints.segments.bank.HIUPD6* attribute), 43  
AllowedFormat (class in *fints.formals*), 84  
AllowedTransaction (class in *fints.formals*), 84  
ALPHANUMERIC (*fints.formals.AllowedFormat* attribute), 85  
AlphanumericField (class in *fints.fields*), 77  
amount (*fints.formals.Amount1* attribute), 83  
amount (*fints.formals.Balance1* attribute), 99  
amount (*fints.formals.Balance2* attribute), 100  
Amount1 (class in *fints.formals*), 83  
AmountField (class in *fints.fields*), 78  
as\_mt940\_Balance() (*fints.formals.Balance1* method), 99  
as\_mt940\_Balance() (*fints.formals.Balance2* method), 100  
as\_sepa\_account() (*fints.formals.KTZ1* method), 95  
atc (*fints.formals.ResponseHHDUC* attribute), 104  
AUTH (*fints.formals.SecurityClass* attribute), 98  
AUTH\_ADV (*fints.formals.SecurityClass* attribute), 98  
AVAILABLE (*fints.formals.TANMediaType2* attribute), 100  
AVAILABLE (*fints.formals.TANMediumStatus* attribute), 101  
available\_amount (*fints.segments.saldo.HISAL5* attribute), 63  
available\_amount (*fints.segments.saldo.HISAL6* attribute), 64



available\_amount (*fints.segments.saldo.HISAL7 attribute*), 65  
 AVAILABLE\_SUCCESSOR (*fints.formals.TANMediumStatus attribute*), 101

## B

Balance1 (*class in fints.formals*), 99  
 Balance2 (*class in fints.formals*), 99  
 balance\_booked (*fints.segments.saldo.HISAL5 attribute*), 63  
 balance\_booked (*fints.segments.saldo.HISAL6 attribute*), 64  
 balance\_booked (*fints.segments.saldo.HISAL7 attribute*), 65  
 balance\_pending (*fints.segments.saldo.HISAL5 attribute*), 63  
 balance\_pending (*fints.segments.saldo.HISAL6 attribute*), 64  
 balance\_pending (*fints.segments.saldo.HISAL7 attribute*), 65  
 bank\_code (*fints.formals.Account2 attribute*), 96  
 bank\_code (*fints.formals.BankIdentifier attribute*), 82  
 bank\_identifier (*fints.formals.Account3 attribute*), 97  
 bank\_identifier (*fints.formals.AccountInformation attribute*), 84  
 bank\_identifier (*fints.formals.KeyName attribute*), 82  
 bank\_identifier (*fints.formals.KTI1 attribute*), 96  
 bank\_identifier (*fints.formals.KTZ1 attribute*), 95  
 bank\_identifier (*fints.segments.auth.HKIDN2 attribute*), 35  
 bank\_identifier (*fints.segments.bank.HIBPA3 attribute*), 41  
 bank\_identifier (*fints.segments.bank.HIKOM4 attribute*), 41  
 bank\_name (*fints.segments.bank.HIBPA3 attribute*), 41  
 BankIdentifier (*class in fints.formals*), 82  
 BatchDebitBase (*class in fints.segments.debit*), 44  
 BatchTransferParameter1 (*class in fints.formals*), 105  
 ben (*fints.segments.auth.HITAN2 attribute*), 31  
 ben (*fints.segments.auth.HITAN3 attribute*), 32  
 ben (*fints.segments.auth.HITAN5 attribute*), 33  
 bic (*fints.formals.KTI1 attribute*), 96  
 bic (*fints.formals.KTZ1 attribute*), 95  
 BILATERAL (*fints.formals.TANMediaClass4 attribute*), 101  
 BinaryField (*class in fints.fields*), 78  
 BookedCamtStatements1 (*class in fints.formals*), 111  
 booking\_date (*fints.segments.saldo.HISAL5 attribute*), 63

booking\_time (*fints.segments.saldo.HISAL5 attribute*), 63  
 booking\_timestamp (*fints.segments.saldo.HISAL6 attribute*), 64  
 booking\_timestamp (*fints.segments.saldo.HISAL7 attribute*), 65  
 BooleanField (*class in fints.fields*), 78  
 BOTH (*fints.formals.TANTimeDialogAssociation attribute*), 84  
 bpd\_version (*fints.segments.auth.HKVVB3 attribute*), 40  
 bpd\_version (*fints.segments.bank.HIBPA3 attribute*), 41  
 BZIP2 (*fints.formals.CompressionFunction attribute*), 97

## C

camt\_descriptor (*fints.segments.statement.HICAZ1 attribute*), 68  
 camt\_statements (*fints.formals.BookedCamtStatements1 attribute*), 111  
 cancel\_allowed (*fints.formals.TwoStepParameters2 attribute*), 87  
 cancel\_allowed (*fints.formals.TwoStepParameters3 attribute*), 88  
 cancel\_allowed (*fints.formals.TwoStepParameters4 attribute*), 90  
 cancel\_allowed (*fints.formals.TwoStepParameters5 attribute*), 91  
 cancel\_allowed (*fints.formals.TwoStepParameters6 attribute*), 92  
 cancel\_task (*fints.segments.auth.HKTAN2 attribute*), 36  
 cancel\_task (*fints.segments.auth.HKTAN3 attribute*), 37  
 cancel\_task (*fints.segments.auth.HKTAN5 attribute*), 38  
 cancel\_task (*fints.segments.auth.HKTAN6 attribute*), 39  
 card\_number (*fints.formals.TANMedia4 attribute*), 101  
 card\_number (*fints.formals.TANMedia5 attribute*), 103  
 card\_sequence (*fints.formals.TANMedia4 attribute*), 101  
 card\_sequence (*fints.formals.TANMedia5 attribute*), 103  
 card\_type (*fints.formals.TANMedia4 attribute*), 102  
 card\_type (*fints.formals.TANMedia5 attribute*), 103  
 CBC (*fints.formals.OperationMode attribute*), 80  
 Certificate (*class in fints.formals*), 83  
 certificate (*fints.segments.message.HNSHK4 attribute*), 61

`certificate` (*fints.segments.message.HNVSK3 attribute*), 62

`certificate_content` (*fints.formals.Certificate attribute*), 83

`certificate_type` (*fints.formals.Certificate attribute*), 83

`challenge` (*fints.client.NeedTANResponse attribute*), 10

`challenge` (*fints.segments.auth.HITAN2 attribute*), 31

`challenge` (*fints.segments.auth.HITAN3 attribute*), 32

`challenge` (*fints.segments.auth.HITAN5 attribute*), 33

`challenge` (*fints.segments.auth.HITAN6 attribute*), 33

`challenge_class` (*fints.segments.auth.HKTAN2 attribute*), 36

`challenge_class` (*fints.segments.auth.HKTAN3 attribute*), 37

`challenge_class` (*fints.segments.auth.HKTAN5 attribute*), 38

`challenge_class` (*fints.segments.auth.HKTAN6 attribute*), 39

`challenge_class_required` (*fints.formals.TwoStepParameters2 attribute*), 87

`challenge_class_required` (*fints.formals.TwoStepParameters3 attribute*), 88

`challenge_class_required` (*fints.formals.TwoStepParameters4 attribute*), 90

`challenge_class_required` (*fints.formals.TwoStepParameters5 attribute*), 91

`challenge_class_required` (*fints.formals.TwoStepParameters6 attribute*), 93

`challenge_hhduc` (*fints.client.NeedTANResponse attribute*), 10

`challenge_hhduc` (*fints.segments.auth.HITAN5 attribute*), 33

`challenge_hhduc` (*fints.segments.auth.HITAN6 attribute*), 34

`challenge_html` (*fints.client.NeedTANResponse attribute*), 10

`challenge_matrix` (*fints.client.NeedTANResponse attribute*), 10

`challenge_raw` (*fints.client.NeedTANResponse attribute*), 10

`challenge_structured` (*fints.formals.TwoStepParameters4 attribute*), 90

`challenge_structured` (*fints.formals.TwoStepParameters5 attribute*), 91

`challenge_structured` (*fints.formals.TwoStepParameters6 attribute*), 93

`challenge_valid_until` (*fints.segments.auth.HITAN2 attribute*), 31

`challenge_valid_until` (*fints.segments.auth.HITAN3 attribute*), 32

`challenge_valid_until` (*fints.segments.auth.HITAN5 attribute*), 33

`challenge_valid_until` (*fints.segments.auth.HITAN6 attribute*), 34

`challenge_value_required` (*fints.formals.TwoStepParameters2 attribute*), 87

`challenge_value_required` (*fints.formals.TwoStepParameters3 attribute*), 88

`challenge_value_required` (*fints.formals.TwoStepParameters4 attribute*), 90

`ChallengeValidUntil` (*class in fints.formals*), 105

`cid` (*fints.formals.SecurityIdentificationDetails attribute*), 80

`CLEARTEXT_PIN_NO_TAN` (*fints.formals.InitializationMode attribute*), 85

`code` (*fints.formals.Response attribute*), 83

`CodeField` (*class in fints.fields*), 78

`CodeFieldMixin` (*class in fints.fields*), 78

`COM` (*fints.formals.CompressionFunction attribute*), 97

`communication_parameters` (*fints.segments.bank.HIKOM4 attribute*), 41

`CommunicationParameter2` (*class in fints.formals*), 105

`compression_function` (*fints.segments.message.HNVSK3 attribute*), 62

`CompressionFunction` (*class in fints.formals*), 97

`CON` (*fints.formals.SecurityRole attribute*), 97

`Container` (*class in fints.types*), 77

`ContainerField` (*class in fints.fields*), 77

`ContainerMeta` (*class in fints.types*), 77

`COUNTRY_ALPHA_TO_NUMERIC` (*fints.formals.BankIdentifier attribute*), 82

`country_identifier` (*fints.formals.Account2 attribute*), 96

`country_identifier` (*fints.formals.BankIdentifier attribute*), 82

`COUNTRY_NUMERIC_TO_ALPHA` (*fints.formals.BankIdentifier attribute*), 82

`CountryField` (*class in fints.fields*), 78

`CREDIT` (*fints.formals.CreditDebit2 attribute*), 99

`credit_card_number` (*fints.segments.statement.DKKKU2 attribute*), 68

- credit\_debit (*fints.formals.Balance1* attribute), 99
  - credit\_debit (*fints.formals.Balance2* attribute), 99
  - CreditDebit2 (class in *fints.formals*), 99
  - CRT (*fints.formals.DateTimeType* attribute), 80
  - currency (*fints.formals.Amount1* attribute), 83
  - currency (*fints.formals.Balance1* attribute), 99
  - currency (*fints.segments.depot.HKWPD5* attribute), 55
  - currency (*fints.segments.depot.HKWPD6* attribute), 55
  - currency (*fints.segments.saldo.HISAL5* attribute), 63
  - currency (*fints.segments.saldo.HISAL6* attribute), 64
  - currency (*fints.segments.saldo.HISAL7* attribute), 65
  - CurrencyField (class in *fints.fields*), 78
  - customer\_id (*fints.segments.auth.HKIDN2* attribute), 35
  - customer\_id (*fints.segments.bank.HIUPD6* attribute), 43
  - customer\_id\_field\_text (*fints.formals.ParameterPinTan* attribute), 94
  - cutoff\_days (*fints.formals.QueryCreditCardStatements2* attribute), 110
  - cvr (*fints.formals.ResponseHHDUC* attribute), 104
- ## D
- D (*fints.formals.KeyType* attribute), 82
  - data (*fints.segments.message.HNVSD1* attribute), 62
  - DataElementField (class in *fints.fields*), 77
  - DataElementGroup (class in *fints.formals*), 79
  - DataElementGroupField (class in *fints.fields*), 77
  - date (*fints.formals.Balance1* attribute), 99
  - date (*fints.formals.Balance2* attribute), 100
  - date (*fints.formals.ChallengeValidUntil* attribute), 105
  - date (*fints.formals.SecurityDateTime* attribute), 80
  - date (*fints.formals.Timestamp1* attribute), 100
  - date (*fints.segments.journal.HIPRO3* attribute), 57
  - date (*fints.segments.journal.HIPRO4* attribute), 58
  - date\_booked (*fints.segments.debit.HIDMB1* attribute), 47
  - date\_due (*fints.segments.saldo.HISAL5* attribute), 63
  - date\_due (*fints.segments.saldo.HISAL6* attribute), 64
  - date\_due (*fints.segments.saldo.HISAL7* attribute), 65
  - date\_end (*fints.segments.debit.HKDBS1* attribute), 51
  - date\_end (*fints.segments.debit.HKDBS2* attribute), 51
  - date\_end (*fints.segments.debit.HKDMB1* attribute), 52
  - date\_end (*fints.segments.journal.HKPRO3* attribute), 58
  - date\_end (*fints.segments.journal.HKPRO4* attribute), 59
  - date\_end (*fints.segments.statement.DKKKU2* attribute), 68
  - date\_end (*fints.segments.statement.HKCAZ1* attribute), 70
  - date\_end (*fints.segments.statement.HKKAZ5* attribute), 70
  - date\_end (*fints.segments.statement.HKKAZ6* attribute), 71
  - date\_end (*fints.segments.statement.HKKAZ7* attribute), 72
  - date\_entered (*fints.segments.debit.HIDMB1* attribute), 47
  - date\_range\_allowed (*fints.formals.QueryCreditCardStatements2* attribute), 110
  - date\_range\_allowed (*fints.formals.QueryScheduledBatchDebitParameter1* attribute), 110
  - date\_range\_allowed (*fints.formals.QueryScheduledDebitParameter1* attribute), 109
  - date\_range\_allowed (*fints.formals.QueryScheduledDebitParameter2* attribute), 110
  - date\_start (*fints.segments.debit.HKDBS1* attribute), 51
  - date\_start (*fints.segments.debit.HKDBS2* attribute), 51
  - date\_start (*fints.segments.debit.HKDMB1* attribute), 52
  - date\_start (*fints.segments.journal.HKPRO3* attribute), 59
  - date\_start (*fints.segments.journal.HKPRO4* attribute), 59
  - date\_start (*fints.segments.statement.DKKKU2* attribute), 68
  - date\_start (*fints.segments.statement.HKCAZ1* attribute), 70
  - date\_start (*fints.segments.statement.HKKAZ5* attribute), 71
  - date\_start (*fints.segments.statement.HKKAZ6* attribute), 71
  - date\_start (*fints.segments.statement.HKKAZ7* attribute), 72
  - date\_time\_type (*fints.formals.SecurityDateTime* attribute), 80
  - DateField (class in *fints.fields*), 78
  - DateTimeType (class in *fints.formals*), 80
  - DDV (*fints.formals.SecurityMethod* attribute), 79
  - DE (*fints.formals.Language2* attribute), 95
  - DEBIT (*fints.formals.CreditDebit2* attribute), 99
  - debit\_count (*fints.segments.debit.HIDMB1* attribute), 47
  - DebitResponseBase (class in *fints.segments.debit*), 45
  - DECLINED (*fints.formals.StatusSEPATask1* attribute), 111
  - DEFAULT (*fints.formals.Language2* attribute), 94

default\_language (*fints.segments.bank.HIKOM4 attribute*), 42

DELETION (*fints.formals.SEPACCode1 attribute*), 110

description\_required (*fints.formals.TwoStepParameters3 attribute*), 89

description\_required (*fints.formals.TwoStepParameters4 attribute*), 90

description\_required (*fints.formals.TwoStepParameters5 attribute*), 91

description\_required (*fints.formals.TwoStepParameters6 attribute*), 93

DescriptionRequired (*class in fints.formals*), 85

dialog\_id (*fints.formals.ReferenceMessage attribute*), 79

dialog\_id (*fints.segments.dialog.HKEND1 attribute*), 57

dialog\_id (*fints.segments.message.HNHBK3 attribute*), 59

DigitsField (*class in fints.fields*), 77

DIKKU2 (*class in fints.segments.statement*), 67

DIKKUS2 (*class in fints.segments.statement*), 67

DKKKU2 (*class in fints.segments.statement*), 67

DTAUSField (*class in fints.fields*), 77

## E

ef\_id\_data (*fints.formals.ResponseHHDUC attribute*), 104

EN (*fints.formals.Language2 attribute*), 95

ENCRYPTED\_PIN\_NO\_TAN (*fints.formals.InitializationMode attribute*), 85

encryption\_algorithm (*fints.formals.EncryptionAlgorithm attribute*), 81

encryption\_algorithm (*fints.segments.message.HNVSK3 attribute*), 62

EncryptionAlgorithm (*class in fints.formals*), 81

EncryptionAlgorithmCoded (*class in fints.formals*), 81

end\_bank\_identifier (*fints.segments.bank.HKKOM4 attribute*), 43

EXACTLY\_ONE (*fints.formals.TANUsageOption attribute*), 104

expected\_type (*fints.formals.SupportedMessageTypes attribute*), 111

extension (*fints.segments.bank.HIUPA4 attribute*), 42

extension (*fints.segments.bank.HIUPD6 attribute*), 43

## F

Field (*class in fints.types*), 77

filter\_function (*fints.formals.CommunicationParameter2 attribute*), 106

filter\_function\_version (*fints.formals.CommunicationParameter2 attribute*), 106

find\_segment\_first () (*fints.types.SegmentSequence method*), 74

find\_segment\_highest\_version () (*fints.types.SegmentSequence method*), 74

find\_segments () (*fints.types.SegmentSequence method*), 75

find\_subclass () (*fints.segments.base.FinTS3Segment class method*), 29

fints.fields (*module*), 77

fints.formals (*module*), 79

fints.segments.accounts (*module*), 29

fints.segments.auth (*module*), 30

fints.segments.bank (*module*), 40

fints.segments.base (*module*), 44

fints.segments.debit (*module*), 44

fints.segments.depot (*module*), 54

fints.segments.dialog (*module*), 56

fints.segments.journal (*module*), 57

fints.segments.message (*module*), 59

fints.segments.saldo (*module*), 63

fints.segments.statement (*module*), 67

fints.segments.transfer (*module*), 72

fints.types (*module*), 77

FinTS3Parser (*class in fints.parser*), 27

FinTS3Segment (*class in fints.segments.base*), 29

FinTS3SegmentMeta (*class in fints.segments.base*), 44

FinTS3Serializer (*class in fints.parser*), 27

FloatField (*class in fints.fields*), 78

FR (*fints.formals.Language2 attribute*), 95

from\_data () (*fints.client.NeedRetryResponse class method*), 11

from\_sepa\_account () (*fints.formals.Account2 class method*), 96

from\_sepa\_account () (*fints.formals.Account3 class method*), 97

from\_sepa\_account () (*fints.formals.KTII class method*), 96

from\_sepa\_account () (*fints.formals.KTZI class method*), 95

further\_tan\_follows (*fints.segments.auth.HKTAN2 attribute*), 36

further\_tan\_follows (*fints.segments.auth.HKTAN3 attribute*), 37

further\_tan\_follows



(*fints.segments.auth.HKTAN5* attribute), 38  
 further\_tan\_follows  
 (*fints.segments.auth.HKTAN6* attribute), 39

## G

GENERATOR (*fints.formals.TANMediaClass3* attribute), 100  
 GENERATOR (*fints.formals.TANMediaClass4* attribute), 101  
 GenericField (*class in fints.fields*), 77  
 GenericGroupField (*class in fints.fields*), 77  
 get\_data() (*fints.client.NeedTANResponse* method), 10  
 GetSEPAAccountParameter1 (*class in fints.formals*), 111  
 GZIP (*fints.formals.CompressionFunction* attribute), 97

## H

hash\_algorithm (*fints.formals.HashAlgorithm* attribute), 82  
 hash\_algorithm (*fints.segments.message.HNSHK4* attribute), 61  
 HashAlgorithm (*class in fints.formals*), 81  
 hbci\_version (*fints.segments.message.HNHBK3* attribute), 60  
 header (*fints.segments.base.FinTS3Segment* attribute), 29  
 HIBPA3 (*class in fints.segments.bank*), 40  
 HICAZ1 (*class in fints.segments.statement*), 68  
 HICCMS1 (*class in fints.segments.transfer*), 72  
 HIDBS1 (*class in fints.segments.debit*), 45  
 HIDBS2 (*class in fints.segments.debit*), 46  
 HIDBSS1 (*class in fints.segments.debit*), 46  
 HIDBSS2 (*class in fints.segments.debit*), 47  
 HIDMB1 (*class in fints.segments.debit*), 47  
 HIDMBS1 (*class in fints.segments.debit*), 47  
 HIDMC1 (*class in fints.segments.debit*), 48  
 HIDMCS1 (*class in fints.segments.debit*), 48  
 HIDME1 (*class in fints.segments.debit*), 48  
 HIDME2 (*class in fints.segments.debit*), 48  
 HIDMES1 (*class in fints.segments.debit*), 49  
 HIDMES2 (*class in fints.segments.debit*), 49  
 HIDSC1 (*class in fints.segments.debit*), 49  
 HIDSCS1 (*class in fints.segments.debit*), 49  
 HIDSE1 (*class in fints.segments.debit*), 49  
 HIDSE2 (*class in fints.segments.debit*), 50  
 HIDSES1 (*class in fints.segments.debit*), 50  
 HIDSES2 (*class in fints.segments.debit*), 50  
 HIKAZ5 (*class in fints.segments.statement*), 68  
 HIKAZ6 (*class in fints.segments.statement*), 69  
 HIKAZ7 (*class in fints.segments.statement*), 69  
 HIKOM4 (*class in fints.segments.bank*), 41

HIPINS1 (*class in fints.segments.auth*), 30  
 HIPRO3 (*class in fints.segments.journal*), 57  
 HIPRO4 (*class in fints.segments.journal*), 57  
 HIPROS3 (*class in fints.segments.journal*), 58  
 HIPROS4 (*class in fints.segments.journal*), 58  
 HIRMG2 (*class in fints.segments.dialog*), 56  
 HIRMS2 (*class in fints.segments.dialog*), 56  
 HISAL5 (*class in fints.segments.saldo*), 63  
 HISAL6 (*class in fints.segments.saldo*), 64  
 HISAL7 (*class in fints.segments.saldo*), 64  
 HISPA1 (*class in fints.segments.accounts*), 29  
 HISPAS1 (*class in fints.segments.accounts*), 30  
 HISYN4 (*class in fints.segments.dialog*), 56  
 HITAB4 (*class in fints.segments.auth*), 30  
 HITAB5 (*class in fints.segments.auth*), 31  
 HITAN2 (*class in fints.segments.auth*), 31  
 HITAN3 (*class in fints.segments.auth*), 32  
 HITAN5 (*class in fints.segments.auth*), 32  
 HITAN6 (*class in fints.segments.auth*), 33  
 HITANS1 (*class in fints.segments.auth*), 34  
 HITANS2 (*class in fints.segments.auth*), 34  
 HITANS3 (*class in fints.segments.auth*), 34  
 HITANS4 (*class in fints.segments.auth*), 34  
 HITANS5 (*class in fints.segments.auth*), 35  
 HITANS6 (*class in fints.segments.auth*), 35  
 HITANSBase (*class in fints.segments.auth*), 35  
 HIUPA4 (*class in fints.segments.bank*), 42  
 HIUPD6 (*class in fints.segments.bank*), 42  
 HIWPD5 (*class in fints.segments.depot*), 54  
 HIWPD6 (*class in fints.segments.depot*), 54  
 HKCAZ1 (*class in fints.segments.statement*), 69  
 HKCCM1 (*class in fints.segments.transfer*), 72  
 HKCCS1 (*class in fints.segments.transfer*), 73  
 HKDBS1 (*class in fints.segments.debit*), 50  
 HKDBS2 (*class in fints.segments.debit*), 51  
 HKDMB1 (*class in fints.segments.debit*), 52  
 HKDMC1 (*class in fints.segments.debit*), 52  
 HKDME1 (*class in fints.segments.debit*), 52  
 HKDME2 (*class in fints.segments.debit*), 52  
 HKDSC1 (*class in fints.segments.debit*), 53  
 HKDSE1 (*class in fints.segments.debit*), 53  
 HKDSE2 (*class in fints.segments.debit*), 53  
 HKEND1 (*class in fints.segments.dialog*), 56  
 HKIDN2 (*class in fints.segments.auth*), 35  
 HKIPM1 (*class in fints.segments.transfer*), 73  
 HKIPZ1 (*class in fints.segments.transfer*), 74  
 HKKAZ5 (*class in fints.segments.statement*), 70  
 HKKAZ6 (*class in fints.segments.statement*), 71  
 HKKAZ7 (*class in fints.segments.statement*), 71  
 HKKOM4 (*class in fints.segments.bank*), 43  
 HKPRO3 (*class in fints.segments.journal*), 58  
 HKPRO4 (*class in fints.segments.journal*), 59  
 HKSAL5 (*class in fints.segments.saldo*), 65  
 HKSAL6 (*class in fints.segments.saldo*), 66

HKSAL7 (class in *fints.segments.saldo*), 66  
HKSPA1 (class in *fints.segments.accounts*), 30  
HKSYN3 (class in *fints.segments.dialog*), 57  
HKTAB4 (class in *fints.segments.auth*), 36  
HKTAB5 (class in *fints.segments.auth*), 36  
HKTAN2 (class in *fints.segments.auth*), 36  
HKTAN3 (class in *fints.segments.auth*), 37  
HKTAN5 (class in *fints.segments.auth*), 38  
HKTAN6 (class in *fints.segments.auth*), 39  
HKVVB3 (class in *fints.segments.auth*), 40  
HKWPD5 (class in *fints.segments.depot*), 54  
HKWPD6 (class in *fints.segments.depot*), 55  
HNHBK3 (class in *fints.segments.message*), 59  
HNHBS1 (class in *fints.segments.message*), 60  
HNSHA2 (class in *fints.segments.message*), 60  
HNSHK4 (class in *fints.segments.message*), 60  
HNVSD1 (class in *fints.segments.message*), 61  
HNVSK3 (class in *fints.segments.message*), 62  
Holding (class in *fints.models*), 6  
holdings (*fints.segments.depot.HIWPD5* attribute), 54  
holdings (*fints.segments.depot.HIWPD6* attribute), 54  
HTTPS (*fints.formals.ServiceType2* attribute), 105

## I

iban (*fints.formals.KT11* attribute), 96  
iban (*fints.formals.KTZ1* attribute), 95  
iban (*fints.segments.bank.HIUPD6* attribute), 43  
ID\_NECESSARY (*fints.formals.SystemIDStatus* attribute), 98  
ID\_UNNECESSARY (*fints.formals.SystemIDStatus* attribute), 98  
identified\_role (*fints.formals.SecurityIdentificationDetails* attribute), 80  
IdentifiedRole (class in *fints.formals*), 79  
identifier (*fints.formals.SecurityIdentificationDetails* attribute), 80  
IDField (class in *fints.fields*), 78  
IN\_PROGRESS (*fints.formals.StatusSEPATask1* attribute), 111  
initialization\_mode  
    (*fints.formals.TwoStepParameters3* attribute), 88  
initialization\_mode  
    (*fints.formals.TwoStepParameters4* attribute), 90  
initialization\_mode  
    (*fints.formals.TwoStepParameters5* attribute), 91  
initialization\_mode  
    (*fints.formals.TwoStepParameters6* attribute), 93  
InitializationMode (class in *fints.formals*), 85  
IntCodeField (class in *fints.fields*), 78  
is\_sepa (*fints.formals.KTZ1* attribute), 95

is\_unset () (*fints.types.Container* method), 77  
ISO\_9796\_1 (*fints.formals.OperationMode* attribute), 80  
ISO\_9796\_2\_RANDOM (*fints.formals.OperationMode* attribute), 80  
ISS (*fints.formals.SecurityRole* attribute), 97  
IVC (*fints.formals.AlgorithmParameterIVName* attribute), 81

## K

key\_name (*fints.segments.message.HNSHK4* attribute), 61  
key\_name (*fints.segments.message.HNVSK3* attribute), 62  
key\_number (*fints.formals.KeyName* attribute), 83  
key\_type (*fints.formals.KeyName* attribute), 82  
key\_version (*fints.formals.KeyName* attribute), 83  
KeyName (class in *fints.formals*), 82  
KeyType (class in *fints.formals*), 82  
KT11 (class in *fints.formals*), 95  
KTZ1 (class in *fints.formals*), 95  
KYE (*fints.formals.AlgorithmParameterName* attribute), 81  
KYP (*fints.formals.AlgorithmParameterName* attribute), 81

## L

language (*fints.segments.auth.HKVVB3* attribute), 40  
Language2 (class in *fints.formals*), 94  
languages (*fints.formals.SupportedLanguages2* attribute), 95  
LAST\_MESSAGE (*fints.formals.SynchronizationMode* attribute), 99  
last\_use (*fints.formals.TANMedia4* attribute), 102  
last\_use (*fints.formals.TANMedia5* attribute), 104  
limit\_amount (*fints.formals.AccountLimit* attribute), 84  
limit\_amount (*fints.formals.AllowedTransaction* attribute), 84  
limit\_days (*fints.formals.AccountLimit* attribute), 84  
limit\_days (*fints.formals.AllowedTransaction* attribute), 84  
limit\_type (*fints.formals.AccountLimit* attribute), 84  
limit\_type (*fints.formals.AllowedTransaction* attribute), 84  
line\_of\_credit (*fints.segments.saldo.HISAL5* attribute), 63  
line\_of\_credit (*fints.segments.saldo.HISAL6* attribute), 64  
line\_of\_credit (*fints.segments.saldo.HISAL7* attribute), 65  
LIST (*fints.formals.TANMediaClass3* attribute), 100  
LIST (*fints.formals.TANMediaClass4* attribute), 101

LZHuf ( <i>fints.formals.CompressionFunction</i> attribute), 97	max_length_input ( <i>fints.formals.TwoStepParameters6</i> attribute), 92
LZSS ( <i>fints.formals.CompressionFunction</i> attribute), 97	max_length_return_value
LZW ( <i>fints.formals.CompressionFunction</i> attribute), 97	( <i>fints.formals.TwoStepParameters1</i> attribute), 86
<b>M</b>	max_length_return_value
MASK_02 ( <i>fints.formals.InitializationMode</i> attribute), 85	( <i>fints.formals.TwoStepParameters2</i> attribute), 87
max_advance_notice	max_length_return_value
( <i>fints.formals.ScheduledBatchDebitParameter2</i> attribute), 107	( <i>fints.formals.TwoStepParameters3</i> attribute), 88
max_advance_notice	max_length_return_value
( <i>fints.formals.ScheduledDebitParameter2</i> attribute), 106	( <i>fints.formals.TwoStepParameters4</i> attribute), 89
max_advance_notice_FINAL_RCUR	max_length_return_value
( <i>fints.formals.ScheduledBatchDebitParameter1</i> attribute), 107	( <i>fints.formals.TwoStepParameters5</i> attribute), 91
max_advance_notice_FINAL_RCUR	max_length_return_value
( <i>fints.formals.ScheduledCORIBatchDebitParameter1</i> attribute), 109	( <i>fints.formals.TwoStepParameters6</i> attribute), 92
max_advance_notice_FINAL_RCUR	max_message_length ( <i>fints.segments.bank.HIBPA3</i> attribute), 41
( <i>fints.formals.ScheduledCORIBatchDebitParameter1</i> attribute), 108	max_number_responses
max_advance_notice_FINAL_RCUR	( <i>fints.segments.bank.HKKOM4</i> attribute), 43
( <i>fints.formals.ScheduledDebitParameter1</i> attribute), 106	max_number_responses
max_advance_notice_FRST_OOFF	( <i>fints.segments.debit.HKDBS1</i> attribute), 51
( <i>fints.formals.ScheduledBatchDebitParameter1</i> attribute), 107	max_number_responses
max_advance_notice_FRST_OOFF	( <i>fints.segments.debit.HKDBS2</i> attribute), 51
( <i>fints.formals.ScheduledCORIBatchDebitParameter1</i> attribute), 109	max_number_responses
max_advance_notice_FRST_OOFF	( <i>fints.segments.debit.HKDDB1</i> attribute), 52
( <i>fints.formals.ScheduledCORIBatchDebitParameter1</i> attribute), 108	max_number_responses
max_advance_notice_FRST_OOFF	( <i>fints.segments.depot.HKWPD5</i> attribute), 55
( <i>fints.formals.ScheduledDebitParameter1</i> attribute), 106	max_number_responses
max_debit_count ( <i>fints.formals.ScheduledBatchDebitParameter1</i> attribute), 107	( <i>fints.segments.depot.HKWPD6</i> attribute), 55
max_debit_count ( <i>fints.formals.ScheduledBatchDebitParameter2</i> attribute), 107	max_number_responses
max_debit_count ( <i>fints.formals.ScheduledCORIBatchDebitParameter1</i> attribute), 108	( <i>fints.segments.journal.HKPRO3</i> attribute), 59
max_length_input ( <i>fints.formals.TwoStepParameters1</i> attribute), 86	max_number_responses
max_length_input ( <i>fints.formals.TwoStepParameters2</i> attribute), 87	( <i>fints.segments.journal.HKPRO4</i> attribute), 59
max_length_input ( <i>fints.formals.TwoStepParameters3</i> attribute), 88	max_number_responses
max_length_input ( <i>fints.formals.TwoStepParameters4</i> attribute), 89	( <i>fints.segments.saldo.HKSAL5</i> attribute), 66
max_length_input ( <i>fints.formals.TwoStepParameters5</i> attribute), 90	max_number_responses
	( <i>fints.segments.saldo.HKSAL6</i> attribute), 66
	max_number_responses
	( <i>fints.segments.saldo.HKSAL7</i> attribute),

67  
 max\_number\_responses  
     (*fints.segments.statement.DKKKU2* attribute),  
 68  
 max\_number\_responses  
     (*fints.segments.statement.HKCAZ1* attribute),  
 70  
 max\_number\_responses  
     (*fints.segments.statement.HKKAZ5* attribute),  
 71  
 max\_number\_responses  
     (*fints.segments.statement.HKKAZ6* attribute),  
 71  
 max\_number\_responses  
     (*fints.segments.statement.HKKAZ7* attribute),  
 72  
 max\_number\_responses\_allowed  
     (*fints.formals.QueryCreditCardStatements2*  
     attribute), 110  
 max\_number\_responses\_allowed  
     (*fints.formals.QueryScheduledBatchDebitParameter1*  
     attribute), 110  
 max\_number\_responses\_allowed  
     (*fints.formals.QueryScheduledDebitParameter1*  
     attribute), 109  
 max\_number\_responses\_allowed  
     (*fints.formals.QueryScheduledDebitParameter2*  
     attribute), 110  
 max\_number\_tasks (*fints.segments.base.ParameterSegment*  
     attribute), 44  
 max\_number\_tasks (*fints.segments.base.ParameterSegment*  
     attribute), 44  
 max\_pin\_length (*fints.formals.ParameterPinTan* at-  
     tribute), 94  
 max\_tan\_length (*fints.formals.ParameterPinTan* at-  
     tribute), 94  
 max\_timeout (*fints.segments.bank.HIBPA3* attribute),  
 41  
 max\_transfer\_count  
     (*fints.formals.BatchTransferParameter1* at-  
     tribute), 105  
 MAY (*fints.formals.DescriptionRequired* attribute), 85  
 MAY (*fints.formals.SMSSChargeAccountRequired* at-  
     tribute), 85  
 message\_number (*fints.formals.ReferenceMessage* at-  
     tribute), 79  
 message\_number (*fints.segments.dialog.HISYN4* at-  
     tribute), 56  
 message\_number (*fints.segments.message.HNHBK3*  
     attribute), 60  
 message\_number (*fints.segments.message.HNHBS1*  
     attribute), 60  
 message\_size (*fints.segments.message.HNHBK3* at-  
     tribute), 60  
 min\_advance\_notice  
     (*fints.formals.ScheduledBatchDebitParameter2*  
     attribute), 107  
 min\_advance\_notice  
     (*fints.formals.ScheduledDebitParameter2*  
     attribute), 106  
 min\_advance\_notice\_FINAL\_RCUR  
     (*fints.formals.ScheduledBatchDebitParameter1*  
     attribute), 107  
 min\_advance\_notice\_FINAL\_RCUR  
     (*fints.formals.ScheduledCOR1BatchDebitParameter1*  
     attribute), 109  
 min\_advance\_notice\_FINAL\_RCUR  
     (*fints.formals.ScheduledCOR1DebitParameter1*  
     attribute), 108  
 min\_advance\_notice\_FINAL\_RCUR  
     (*fints.formals.ScheduledDebitParameter1*  
     attribute), 106  
 min\_advance\_notice\_FRST\_OOFF  
     (*fints.formals.ScheduledBatchDebitParameter1*  
     attribute), 107  
 min\_advance\_notice\_FRST\_OOFF  
     (*fints.formals.ScheduledCOR1BatchDebitParameter1*  
     attribute), 109  
 min\_advance\_notice\_FRST\_OOFF  
     (*fints.formals.ScheduledCOR1DebitParameter1*  
     attribute), 108  
 min\_advance\_notice\_FRST\_OOFF  
     (*fints.formals.ScheduledDebitParameter1*  
     attribute), 106  
 min\_number\_signatures  
     (*fints.segments.base.ParameterSegment* at-  
     tribute), 44  
 min\_number\_signatures  
     (*fints.segments.base.ParameterSegment\_22*  
     attribute), 44  
 min\_pin\_length (*fints.formals.ParameterPinTan* at-  
     tribute), 94  
 min\_timeout (*fints.segments.bank.HIBPA3* attribute),  
 41  
 MOBILE (*fints.formals.TANMediaClass3* attribute), 100  
 MOBILE (*fints.formals.TANMediaClass4* attribute), 101  
 MOBILE\_AND\_GENERATOR  
     (*fints.formals.TANUsageOption* attribute),  
 104  
 mobile\_number (*fints.formals.TANMedia4* attribute),  
 102  
 mobile\_number (*fints.formals.TANMedia5* attribute),  
 103  
 mobile\_number\_masked (*fints.formals.TANMedia4*  
     attribute), 102  
 mobile\_number\_masked (*fints.formals.TANMedia5*  
     attribute), 103  
 MR (*fints.formals.IdentifiedRole* attribute), 80



`mro()` (*fints.segments.base.FinTS3SegmentMeta* method), 44  
`MS` (*fints.formals.IdentifiedRole* attribute), 80  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters1* attribute), 86  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters2* attribute), 87  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters3* attribute), 88  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters4* attribute), 89  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters5* attribute), 91  
`multiple_tans_allowed` (*fints.formals.TwoStepParameters6* attribute), 92  
`multiple_tasks_allowed` (*fints.formals.ParameterTwostepCommon* attribute), 93  
`MUST` (*fints.formals.DescriptionRequired* attribute), 85  
`MUST` (*fints.formals.PrincipalAccountRequired* attribute), 85  
`MUST` (*fints.formals.SMSChargeAccountRequired* attribute), 85  
`MUST_NOT` (*fints.formals.DescriptionRequired* attribute), 85  
`MUST_NOT` (*fints.formals.PrincipalAccountRequired* attribute), 85  
`MUST_NOT` (*fints.formals.SMSChargeAccountRequired* attribute), 85

## N

`naive_parse()` (*fints.types.Container* class method), 77  
`name` (*fints.formals.TwoStepParameters1* attribute), 86  
`name` (*fints.formals.TwoStepParameters2* attribute), 87  
`name` (*fints.formals.TwoStepParameters3* attribute), 88  
`name` (*fints.formals.TwoStepParameters4* attribute), 89  
`name` (*fints.formals.TwoStepParameters5* attribute), 90  
`name` (*fints.formals.TwoStepParameters6* attribute), 92  
`name_account_owner_1` (*fints.segments.bank.HIUPD6* attribute), 43  
`name_account_owner_2` (*fints.segments.bank.HIUPD6* attribute), 43  
`national_account_allowed` (*fints.formals.GetSEPAAccountParameter1* attribute), 111  
`NeedRetryResponse` (class in *fints.client*), 11  
`NeedTANResponse` (class in *fints.client*), 10  
`NEW_SYSTEM_ID` (*fints.formals.SynchronizationMode* attribute), 99  
`NO` (*fints.formals.TANListNumberRequired* attribute), 85  
`NON_REPUD` (*fints.formals.SecurityClass* attribute), 98  
`NON_REPUD_QUAL` (*fints.formals.SecurityClass* attribute), 98  
`NONE` (*fints.formals.SecurityClass* attribute), 98  
`NONE` (*fints.formals.TaskHashAlgorithm* attribute), 86  
`NOT_ALLOWED` (*fints.formals.TANTimeDialogAssociation* attribute), 84  
`NOT_APPLICABLE` (*fints.formals.TANTimeDialogAssociation* attribute), 84  
`NULL` (*fints.formals.CompressionFunction* attribute), 97  
`number` (*fints.formals.SegmentHeader* attribute), 79  
`number_free_tans` (*fints.formals.TANMedia4* attribute), 102  
`number_free_tans` (*fints.formals.TANMedia5* attribute), 104  
`number_of_supported_lists` (*fints.formals.TwoStepParameters1* attribute), 86  
`number_of_supported_lists` (*fints.formals.TwoStepParameters2* attribute), 87  
`number_of_supported_lists` (*fints.formals.TwoStepParameters3* attribute), 88  
`number_of_supported_lists` (*fints.formals.TwoStepParameters4* attribute), 89  
`number_of_supported_lists` (*fints.formals.TwoStepParameters5* attribute), 91  
`number_tasks` (*fints.segments.bank.HIBPA3* attribute), 41  
`NUMERIC` (*fints.formals.AllowedFormat* attribute), 85  
`NumericField` (class in *fints.fields*), 77

## O

`onestep_method_allowed` (*fints.formals.ParameterTwostepCommon* attribute), 93  
`operation_mode` (*fints.formals.EncryptionAlgorithm* attribute), 81  
`operation_mode` (*fints.formals.SignatureAlgorithm* attribute), 82  
`OperationMode` (class in *fints.formals*), 80  
`OSY` (*fints.formals.UsageEncryption* attribute), 80  
`overdraft` (*fints.segments.saldo.HISAL6* attribute), 64  
`overdraft` (*fints.segments.saldo.HISAL7* attribute), 65

## P

parameter (*fints.segments.accounts.HISPAS1 attribute*), 30

parameter (*fints.segments.auth.HIPINS1 attribute*), 30

parameter (*fints.segments.auth.HITANS1 attribute*), 34

parameter (*fints.segments.auth.HITANS2 attribute*), 34

parameter (*fints.segments.auth.HITANS3 attribute*), 34

parameter (*fints.segments.auth.HITANS4 attribute*), 35

parameter (*fints.segments.auth.HITANS5 attribute*), 35

parameter (*fints.segments.auth.HITANS6 attribute*), 35

parameter (*fints.segments.debit.HIDBSS1 attribute*), 47

parameter (*fints.segments.debit.HIDBSS2 attribute*), 47

parameter (*fints.segments.debit.HIDMBS1 attribute*), 48

parameter (*fints.segments.debit.HIDMCS1 attribute*), 48

parameter (*fints.segments.debit.HIDMES1 attribute*), 49

parameter (*fints.segments.debit.HIDMES2 attribute*), 49

parameter (*fints.segments.debit.HIDSCS1 attribute*), 49

parameter (*fints.segments.debit.HIDSES1 attribute*), 50

parameter (*fints.segments.debit.HIDSES2 attribute*), 50

parameter (*fints.segments.statement.DIKKUS2 attribute*), 67

parameter (*fints.segments.transfer.HICCMS1 attribute*), 72

parameter\_challenge\_class (*fints.segments.auth.HKTAN2 attribute*), 37

parameter\_challenge\_class (*fints.segments.auth.HKTAN3 attribute*), 37

parameter\_challenge\_class (*fints.segments.auth.HKTAN5 attribute*), 38

parameter\_challenge\_class (*fints.segments.auth.HKTAN6 attribute*), 39

ParameterChallengeClass (class in *fints.formals*), 104

ParameterPinTan (class in *fints.formals*), 94

parameters (*fints.formals.ParameterChallengeClass attribute*), 104

parameters (*fints.formals.Response attribute*), 83

ParameterSegment (class in *fints.segments.base*), 44

ParameterSegment\_22 (class in *fints.segments.base*), 44

ParameterTwostepCommon (class in *fints.formals*),

93

ParameterTwostepTAN1 (class in *fints.formals*), 93

ParameterTwostepTAN2 (class in *fints.formals*), 93

ParameterTwostepTAN3 (class in *fints.formals*), 93

ParameterTwostepTAN4 (class in *fints.formals*), 94

ParameterTwostepTAN5 (class in *fints.formals*), 94

ParameterTwostepTAN6 (class in *fints.formals*), 94

parse\_message() (*fints.parser.FinTS3Parser method*), 27

PasswordField (class in *fints.fields*), 78

PENDING (*fints.formals.StatusSEPATask1 attribute*), 111

PIN (*fints.formals.SecurityMethod attribute*), 79

pin (*fints.formals.UserDefinedSignature attribute*), 83

PKCS1V15 (*fints.formals.OperationMode attribute*), 80

principal\_account\_required (*fints.formals.TwoStepParameters5 attribute*), 91

principal\_account\_required (*fints.formals.TwoStepParameters6 attribute*), 92

PrincipalAccountRequired (class in *fints.formals*), 85

print\_nested() (*fints.segments.base.FinTS3Segment method*), 29

print\_nested() (*fints.types.SegmentSequence method*), 75

PROCESSED (*fints.formals.StatusSEPATask1 attribute*), 111

product\_name (*fints.segments.auth.HKVVB3 attribute*), 40

product\_version (*fints.segments.auth.HKVVB3 attribute*), 40

PSS (*fints.formals.OperationMode attribute*), 81

## Q

quality (*fints.segments.depot.HKWPD5 attribute*), 55

quality (*fints.segments.depot.HKWPD6 attribute*), 55

QueryCreditCardStatements2 (class in *fints.formals*), 110

QueryScheduledBatchDebitParameter1 (class in *fints.formals*), 110

QueryScheduledDebitParameter1 (class in *fints.formals*), 109

QueryScheduledDebitParameter2 (class in *fints.formals*), 109

## R

RAH (*fints.formals.SecurityMethod attribute*), 79

RDH (*fints.formals.SecurityMethod attribute*), 79

reference (*fints.formals.SegmentHeader attribute*), 79

reference (*fints.segments.journal.HIPRO3 attribute*), 57

reference (*fints.segments.journal.HIPRO4 attribute*), 58

reference\_element (*fints.formals.Response* attribute), 83  
 reference\_message (*fints.segments.journal.HIPRO3* attribute), 57  
 reference\_message (*fints.segments.journal.HIPRO4* attribute), 58  
 reference\_message (*fints.segments.message.HNHBK3* attribute), 60  
 ReferenceMessage (class in *fints.formals*), 79  
 render() (*fints.types.Field* method), 77  
 render\_bytes() (*fints.types.SegmentSequence* method), 75  
 request\_single\_booking (*fints.segments.debit.BatchDebitBase* attribute), 45  
 request\_single\_booking (*fints.segments.transfer.HKCCM1* attribute), 73  
 request\_single\_booking (*fints.segments.transfer.HKIPM1* attribute), 74  
 required\_signatures (*fints.formals.AllowedTransaction* attribute), 84  
 Response (class in *fints.formals*), 83  
 response\_hhd\_uc (*fints.segments.auth.HKTAN6* attribute), 39  
 response\_hhd\_uc\_required (*fints.formals.TwoStepParameters6* attribute), 93  
 ResponseHHDUC (class in *fints.formals*), 104  
 responses (*fints.segments.dialog.HIRMG2* attribute), 56  
 responses (*fints.segments.dialog.HIRMS2* attribute), 56  
 responses (*fints.segments.journal.HIPRO3* attribute), 57  
 responses (*fints.segments.journal.HIPRO4* attribute), 58  
 REVERSAL (*fints.formals.SEPACCode1* attribute), 110  
 REVOCATION (*fints.formals.SEPACCode1* attribute), 110  
 REVOKED (*fints.formals.StatusSEPATask1* attribute), 111  
 RIPEMD\_160 (*fints.formals.TaskHashAlgorithm* attribute), 86

## S

S (*fints.formals.KeyType* attribute), 82  
 ScheduledBatchDebitParameter1 (class in *fints.formals*), 106  
 ScheduledBatchDebitParameter2 (class in *fints.formals*), 107  
 ScheduledCOR1BatchDebitParameter1 (class in *fints.formals*), 108  
 ScheduledCOR1DebitParameter1 (class in *fints.formals*), 108  
 ScheduledDebitParameter1 (class in *fints.formals*), 106  
 ScheduledDebitParameter2 (class in *fints.formals*), 106  
 SECODER (*fints.formals.TANMediaClass3* attribute), 100  
 SECODER (*fints.formals.TANMediaClass4* attribute), 101  
 security\_application\_area (*fints.segments.message.HNSHK4* attribute), 61  
 security\_class (*fints.segments.base.ParameterSegment* attribute), 44  
 security\_datetime (*fints.segments.message.HNSHK4* attribute), 61  
 security\_datetime (*fints.segments.message.HNVSK3* attribute), 62  
 security\_function (*fints.formals.TANMedia5* attribute), 103  
 security\_function (*fints.formals.TwoStepParametersCommon* attribute), 86  
 security\_function (*fints.segments.message.HNSHK4* attribute), 61  
 security\_function (*fints.segments.message.HNVSK3* attribute), 62  
 security\_identification\_details (*fints.segments.message.HNSHK4* attribute), 61  
 security\_identification\_details (*fints.segments.message.HNVSK3* attribute), 62  
 security\_method (*fints.formals.SecurityProfile* attribute), 79  
 security\_method\_version (*fints.formals.SecurityProfile* attribute), 79  
 security\_profile (*fints.segments.message.HNSHK4* attribute), 61  
 security\_profile (*fints.segments.message.HNVSK3* attribute), 62  
 security\_profile\_bank\_signature (*fints.formals.ParameterTwostepTAN1* attribute), 93  
 security\_reference (*fints.segments.message.HNSHA2* attribute), 60  
 security\_reference (*fints.segments.message.HNSHK4* attribute), 61  
 security\_reference\_digital\_signature (*fints.segments.dialog.HISYN4* attribute), 56  
 security\_reference\_number (*fints.segments.message.HNSHK4* attribute), 61  
 security\_reference\_signature\_key (*fints.segments.dialog.HISYN4* attribute), 56  
 security\_role (*fints.segments.message.HNSHK4* at-

tribute), 61

security\_role (*fints.segments.message.HNVSK3 attribute*), 62

SecurityApplicationArea (class in *fints.formals*), 98

SecurityClass (class in *fints.formals*), 98

SecurityDateTime (class in *fints.formals*), 80

SecurityIdentificationDetails (class in *fints.formals*), 80

SecurityMethod (class in *fints.formals*), 79

SecurityProfile (class in *fints.formals*), 79

SecurityRole (class in *fints.formals*), 97

segment\_type (*fints.segments.auth.HKTAN5 attribute*), 38

segment\_type (*fints.segments.auth.HKTAN6 attribute*), 39

SegmentHeader (class in *fints.formals*), 79

SegmentSequence (class in *fints.types*), 74

SegmentSequenceField (class in *fints.fields*), 78

sepa\_c\_code (*fints.segments.debit.HIDBS2 attribute*), 46

sepa\_descriptor (*fints.segments.debit.BatchDebitBase attribute*), 45

sepa\_descriptor (*fints.segments.debit.HIDBS1 attribute*), 45

sepa\_descriptor (*fints.segments.debit.HIDBS2 attribute*), 46

sepa\_descriptor (*fints.segments.debit.HKDSC1 attribute*), 53

sepa\_descriptor (*fints.segments.debit.HKDSE1 attribute*), 53

sepa\_descriptor (*fints.segments.debit.HKDSE2 attribute*), 54

sepa\_descriptor (*fints.segments.transfer.HKCCM1 attribute*), 73

sepa\_descriptor (*fints.segments.transfer.HKCCS1 attribute*), 73

sepa\_descriptor (*fints.segments.transfer.HKIPM1 attribute*), 74

sepa\_descriptor (*fints.segments.transfer.HKIPZ1 attribute*), 74

sepa\_descriptors (*fints.formals.SupportedSEPAPainMessages1 attribute*), 109

sepa\_pain\_message (*fints.segments.debit.BatchDebitBase attribute*), 45

sepa\_pain\_message (*fints.segments.debit.HIDBS1 attribute*), 45

sepa\_pain\_message (*fints.segments.debit.HIDBS2 attribute*), 46

sepa\_pain\_message (*fints.segments.debit.HKDSC1 attribute*), 53

sepa\_pain\_message (*fints.segments.debit.HKDSE1 attribute*), 53

sepa\_pain\_message (*fints.segments.debit.HKDSE2 attribute*), 54

sepa\_pain\_message (*fints.segments.transfer.HKCCM1 attribute*), 73

sepa\_pain\_message (*fints.segments.transfer.HKCCS1 attribute*), 73

sepa\_pain\_message (*fints.segments.transfer.HKIPM1 attribute*), 74

sepa\_pain\_message (*fints.segments.transfer.HKIPZ1 attribute*), 74

SEPAAccount (class in *fints.models*), 4

SEPACode1 (class in *fints.formals*), 110

serialize\_message () (*fints.parser.FinTS3Serializer method*), 27

service\_type (*fints.formals.CommunicationParameter2 attribute*), 105

ServiceType2 (class in *fints.formals*), 105

SHA\_1 (*fints.formals.TaskHashAlgorithm attribute*), 86

SHM (*fints.formals.SecurityApplicationArea attribute*), 98

SHT (*fints.formals.SecurityApplicationArea attribute*), 98

signature\_algorithm (*fints.formals.SignatureAlgorithm attribute*), 82

signature\_algorithm (*fints.segments.message.HNSHK4 attribute*), 61

SIGNATURE\_ID (*fints.formals.SynchronizationMode attribute*), 99

SignatureAlgorithm (class in *fints.formals*), 82

single\_account\_query\_allowed (*fints.formals.GetSEPAAccountParameter1 attribute*), 111

single\_booking\_allowed (*fints.formals.BatchTransferParameter1 attribute*), 105

single\_booking\_allowed (*fints.formals.ScheduledBatchDebitParameter1 attribute*), 107

single\_booking\_allowed (*fints.formals.ScheduledBatchDebitParameter2 attribute*), 107

single\_booking\_allowed (*fints.formals.ScheduledCOR1BatchDebitParameter1 attribute*), 108

sms\_charge\_account (*fints.formals.TANMedia4 attribute*), 102

sms\_charge\_account (*fints.formals.TANMedia5 attribute*), 103

sms\_charge\_account (*fints.segments.auth.HKTAN5 attribute*), 38

sms\_charge\_account (*fints.segments.auth.HKTAN6 attribute*), 39

sms\_charge\_account\_required



(*fints.formals.TwoStepParameters4* attribute), 90  
 sms\_charge\_account\_required (*fints.formals.TwoStepParameters5* attribute), 91  
 sms\_charge\_account\_required (*fints.formals.TwoStepParameters6* attribute), 92  
 SMSChargeAccountRequired (class in *fints.formals*), 85  
 start\_bank\_identifier (*fints.segments.bank.HKKOM4* attribute), 43  
 statement\_booked (*fints.segments.statement.HICAZ1* attribute), 68  
 statement\_booked (*fints.segments.statement.HIKAZ5* attribute), 69  
 statement\_booked (*fints.segments.statement.HIKAZ6* attribute), 69  
 statement\_booked (*fints.segments.statement.HIKAZ7* attribute), 69  
 statement\_pending (*fints.segments.statement.HICAZ1* attribute), 68  
 statement\_pending (*fints.segments.statement.HIKAZ5* attribute), 69  
 statement\_pending (*fints.segments.statement.HIKAZ6* attribute), 69  
 statement\_pending (*fints.segments.statement.HIKAZ7* attribute), 69  
 status (*fints.formals.TANMedia4* attribute), 101  
 status (*fints.formals.TANMedia5* attribute), 103  
 status\_sepa\_task (*fints.segments.debit.HIDBS2* attribute), 46  
 StatusSEPA\_Task1 (class in *fints.formals*), 110  
 structured\_purpose\_allowed (*fints.formals.GetSEPAAccountParameter1* attribute), 111  
 STS (*fints.formals.DateTimeType* attribute), 80  
 subaccount (*fints.segments.statement.DKKKU2* attribute), 68  
 subaccount\_number (*fints.formals.Account2* attribute), 96  
 subaccount\_number (*fints.formals.Account3* attribute), 97  
 subaccount\_number (*fints.formals.AccountInformation* attribute), 84  
 subaccount\_number (*fints.formals.KTI1* attribute), 96  
 subaccount\_number (*fints.formals.KTZ1* attribute), 95  
 sum\_amount (*fints.segments.debit.BatchDebitBase* attribute), 45  
 sum\_amount (*fints.segments.debit.HIDMB1* attribute), 47  
 sum\_amount (*fints.segments.transfer.HKCCM1* attribute), 73  
 sum\_amount (*fints.segments.transfer.HKIPM1* attribute), 74  
 sum\_amount\_required (*fints.formals.BatchTransferParameter1* attribute), 105  
 sum\_amount\_required (*fints.formals.ScheduledBatchDebitParameter1* attribute), 107  
 sum\_amount\_required (*fints.formals.ScheduledBatchDebitParameter2* attribute), 107  
 sum\_amount\_required (*fints.formals.ScheduledCOR1BatchDebitParameter1* attribute), 108  
 supported\_camt\_messages (*fints.segments.statement.HKCAZ1* attribute), 70  
 supported\_hbci\_version (*fints.segments.bank.HIBPA3* attribute), 41  
 supported\_languages (*fints.segments.bank.HIBPA3* attribute), 41  
 supported\_media\_number (*fints.formals.TwoStepParameters3* attribute), 89  
 supported\_media\_number (*fints.formals.TwoStepParameters4* attribute), 90  
 supported\_media\_number (*fints.formals.TwoStepParameters5* attribute), 92  
 supported\_media\_number (*fints.formals.TwoStepParameters6* attribute), 93  
 supported\_sepa\_formats (*fints.formals.GetSEPAAccountParameter1* attribute), 111  
 supported\_sepa\_formats (*fints.formals.QueryScheduledDebitParameter2* attribute), 110  
 supported\_sepa\_formats (*fints.formals.ScheduledBatchDebitParameter2* attribute), 108  
 supported\_sepa\_formats (*fints.formals.ScheduledCOR1BatchDebitParameter1* attribute), 109  
 supported\_sepa\_formats (*fints.formals.ScheduledCOR1DebitParameter1*

*attribute*), 108  
supported\_sepa\_formats  
    (*fints.formals.ScheduledDebitParameter2*  
    *attribute*), 106  
supported\_sepa\_pain\_messages  
    (*fints.segments.debit.HKDBS1* *attribute*),  
    51  
supported\_sepa\_pain\_messages  
    (*fints.segments.debit.HKDBS2* *attribute*),  
    51  
SupportedHBCIVersions2 (*class in fints.formals*),  
    95  
SupportedLanguages2 (*class in fints.formals*), 95  
SupportedMessageTypes (*class in fints.formals*),  
    111  
SupportedSEPAPainMessages1 (*class in*  
    *fints.formals*), 109  
synchronization\_mode  
    (*fints.segments.dialog.HKSYN3* *attribute*),  
    57  
SynchronizationMode (*class in fints.formals*), 98  
system\_id (*fints.segments.auth.HKIDN2* *attribute*), 35  
system\_id (*fints.segments.dialog.HISYN4* *attribute*),  
    56  
system\_id\_status (*fints.segments.auth.HKIDN2* *at-*  
    *tribute*), 35  
SystemIDStatus (*class in fints.formals*), 98

## T

T\_ONLINE (*fints.formals.ServiceType2* *attribute*), 105  
tan (*fints.formals.UserDefinedSignature* *attribute*), 83  
tan\_list\_number (*fints.formals.TANMedia4* *at-*  
    *tribute*), 102  
tan\_list\_number (*fints.formals.TANMedia5* *at-*  
    *tribute*), 103  
tan\_list\_number (*fints.segments.auth.HITAN2* *at-*  
    *tribute*), 31  
tan\_list\_number (*fints.segments.auth.HITAN3* *at-*  
    *tribute*), 32  
tan\_list\_number (*fints.segments.auth.HITAN5* *at-*  
    *tribute*), 33  
tan\_list\_number (*fints.segments.auth.HKTAN2* *at-*  
    *tribute*), 37  
tan\_list\_number (*fints.segments.auth.HKTAN3* *at-*  
    *tribute*), 37  
tan\_list\_number (*fints.segments.auth.HKTAN5* *at-*  
    *tribute*), 38  
tan\_list\_number\_required  
    (*fints.formals.TwoStepParameters2* *attribute*),  
    87  
tan\_list\_number\_required  
    (*fints.formals.TwoStepParameters3* *attribute*),  
    88  
tan\_list\_number\_required  
    (*fints.formals.TwoStepParameters4* *attribute*),  
    89  
tan\_list\_number\_required  
    (*fints.formals.TwoStepParameters5* *attribute*),  
    91  
tan\_media\_class (*fints.segments.auth.HKTAB4* *at-*  
    *tribute*), 36  
tan\_media\_class (*fints.segments.auth.HKTAB5* *at-*  
    *tribute*), 36  
tan\_media\_list (*fints.segments.auth.HITAB4*  
    *attribute*), 31  
tan\_media\_list (*fints.segments.auth.HITAB5*  
    *attribute*), 31  
tan\_media\_type (*fints.segments.auth.HKTAB4* *at-*  
    *tribute*), 36  
tan\_media\_type (*fints.segments.auth.HKTAB5* *at-*  
    *tribute*), 36  
tan\_medium\_class (*fints.formals.TANMedia4*  
    *attribute*), 101  
tan\_medium\_class (*fints.formals.TANMedia5*  
    *attribute*), 102  
tan\_medium\_name (*fints.formals.TANMedia4* *at-*  
    *tribute*), 102  
tan\_medium\_name (*fints.formals.TANMedia5* *at-*  
    *tribute*), 103  
tan\_medium\_name (*fints.segments.auth.HITAN3* *at-*  
    *tribute*), 32  
tan\_medium\_name (*fints.segments.auth.HITAN5* *at-*  
    *tribute*), 33  
tan\_medium\_name (*fints.segments.auth.HITAN6* *at-*  
    *tribute*), 34  
tan\_medium\_name (*fints.segments.auth.HKTAN3* *at-*  
    *tribute*), 37  
tan\_medium\_name (*fints.segments.auth.HKTAN5* *at-*  
    *tribute*), 38  
tan\_medium\_name (*fints.segments.auth.HKTAN6* *at-*  
    *tribute*), 40  
tan\_process (*fints.formals.TwoStepParametersCommon*  
    *attribute*), 86  
tan\_process (*fints.segments.auth.HITAN2* *attribute*),  
    31  
tan\_process (*fints.segments.auth.HITAN3* *attribute*),  
    32  
tan\_process (*fints.segments.auth.HITAN5* *attribute*),  
    33  
tan\_process (*fints.segments.auth.HITAN6* *attribute*),  
    34  
tan\_process (*fints.segments.auth.HKTAN2* *attribute*),  
    37  
tan\_process (*fints.segments.auth.HKTAN3* *attribute*),  
    37  
tan\_process (*fints.segments.auth.HKTAN5* *attribute*),  
    39

tan\_process (*fints.segments.auth.HKTAN6* attribute), 40  
 tan\_required (*fints.formals.TransactionTanRequired* attribute), 94  
 tan\_time\_delayed\_allowed (*fints.formals.TwoStepParameters1* attribute), 87  
 tan\_time\_dialog\_association (*fints.formals.TwoStepParameters2* attribute), 87  
 tan\_time\_dialog\_association (*fints.formals.TwoStepParameters3* attribute), 88  
 tan\_time\_dialog\_association (*fints.formals.TwoStepParameters4* attribute), 89  
 tan\_time\_dialog\_association (*fints.formals.TwoStepParameters5* attribute), 91  
 tan\_time\_dialog\_association (*fints.formals.TwoStepParameters6* attribute), 92  
 tan\_usage\_option (*fints.segments.auth.HITAB4* attribute), 31  
 tan\_usage\_option (*fints.segments.auth.HITAB5* attribute), 31  
 TANListNumberRequired (class in *fints.formals*), 85  
 TANMedia4 (class in *fints.formals*), 101  
 TANMedia5 (class in *fints.formals*), 102  
 TANMediaClass3 (class in *fints.formals*), 100  
 TANMediaClass4 (class in *fints.formals*), 101  
 TANMediaType2 (class in *fints.formals*), 100  
 TANMediumStatus (class in *fints.formals*), 101  
 TANTimeDialogAssociation (class in *fints.formals*), 84  
 TANUsageOption (class in *fints.formals*), 104  
 task\_cancelable (*fints.segments.debit.HIDBS1* attribute), 45  
 task\_changeable (*fints.segments.debit.HIDBS1* attribute), 46  
 task\_changeable (*fints.segments.debit.HIDBS2* attribute), 46  
 task\_hash\_algorithm (*fints.formals.ParameterTwostepCommon* attribute), 93  
 task\_hash\_value (*fints.segments.auth.HITAN2* attribute), 32  
 task\_hash\_value (*fints.segments.auth.HITAN3* attribute), 32  
 task\_hash\_value (*fints.segments.auth.HITAN5* attribute), 33  
 task\_hash\_value (*fints.segments.auth.HITAN6* attribute), 34  
 task\_hash\_value (*fints.segments.auth.HKTAN2* attribute), 37  
 task\_hash\_value (*fints.segments.auth.HKTAN3* attribute), 38  
 task\_hash\_value (*fints.segments.auth.HKTAN5* attribute), 39  
 task\_hash\_value (*fints.segments.auth.HKTAN6* attribute), 40  
 task\_id (*fints.segments.debit.DebitResponseBase* attribute), 45  
 task\_id (*fints.segments.debit.HIDBS1* attribute), 46  
 task\_id (*fints.segments.debit.HIDBS2* attribute), 46  
 task\_id (*fints.segments.debit.HIDMB1* attribute), 47  
 task\_reference (*fints.segments.auth.HITAN2* attribute), 32  
 task\_reference (*fints.segments.auth.HITAN3* attribute), 32  
 task\_reference (*fints.segments.auth.HITAN5* attribute), 33  
 task\_reference (*fints.segments.auth.HITAN6* attribute), 34  
 task\_reference (*fints.segments.auth.HKTAN2* attribute), 37  
 task\_reference (*fints.segments.auth.HKTAN3* attribute), 38  
 task\_reference (*fints.segments.auth.HKTAN5* attribute), 39  
 task\_reference (*fints.segments.auth.HKTAN6* attribute), 40  
 TaskHashAlgorithm (class in *fints.formals*), 85  
 TCP\_IP (*fints.formals.ServiceType2* attribute), 105  
 tech\_id (*fints.formals.TwoStepParametersCommon* attribute), 86  
 terminal\_flicker\_unix() (in module *fints.hhd.flicker*), 10  
 text (*fints.formals.Response* attribute), 83  
 text\_return\_value (*fints.formals.TwoStepParameters1* attribute), 86  
 text\_return\_value (*fints.formals.TwoStepParameters2* attribute), 87  
 text\_return\_value (*fints.formals.TwoStepParameters3* attribute), 88  
 text\_return\_value (*fints.formals.TwoStepParameters4* attribute), 89  
 text\_return\_value (*fints.formals.TwoStepParameters5* attribute), 91  
 text\_return\_value (*fints.formals.TwoStepParameters6* attribute), 92

TextField (class in *fints.fields*), 77  
 time (*fints.formals.Balance1* attribute), 99  
 time (*fints.formals.Balance2* attribute), 100  
 time (*fints.formals.ChallengeValidUntil* attribute), 105  
 time (*fints.formals.SecurityDateTime* attribute), 80  
 time (*fints.formals.Timestamp1* attribute), 100  
 time (*fints.segments.journal.HIPRO3* attribute), 57  
 time (*fints.segments.journal.HIPRO4* attribute), 58  
 TimeField (class in *fints.fields*), 78  
 Timestamp1 (class in *fints.formals*), 100  
 touchdown\_point (*fints.segments.bank.HKKOM4* attribute), 43  
 touchdown\_point (*fints.segments.debit.HKDBS1* attribute), 51  
 touchdown\_point (*fints.segments.debit.HKDBS2* attribute), 52  
 touchdown\_point (*fints.segments.debit.HKDMB1* attribute), 52  
 touchdown\_point (*fints.segments.depot.HKWPD5* attribute), 55  
 touchdown\_point (*fints.segments.depot.HKWPD6* attribute), 55  
 touchdown\_point (*fints.segments.journal.HKPRO3* attribute), 59  
 touchdown\_point (*fints.segments.journal.HKPRO4* attribute), 59  
 touchdown\_point (*fints.segments.saldo.HKSAL5* attribute), 66  
 touchdown\_point (*fints.segments.saldo.HKSAL6* attribute), 66  
 touchdown\_point (*fints.segments.saldo.HKSAL7* attribute), 67  
 touchdown\_point (*fints.segments.statement.DKKKU2* attribute), 68  
 touchdown\_point (*fints.segments.statement.HKCAZ1* attribute), 70  
 touchdown\_point (*fints.segments.statement.HKKAZ5* attribute), 71  
 touchdown\_point (*fints.segments.statement.HKKAZ6* attribute), 71  
 touchdown\_point (*fints.segments.statement.HKKAZ7* attribute), 72  
 transaction (*fints.formals.AllowedTransaction* attribute), 84  
 transaction (*fints.formals.TransactionTanRequired* attribute), 94  
 transaction\_tans\_required  
     (*fints.formals.ParameterPinTan* attribute), 94  
 TransactionTanRequired (class in *fints.formals*), 94  
 TWOKEY3DES (*fints.formals.EncryptionAlgorithmCoded* attribute), 81  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN1* attribute), 93  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN2* attribute), 93  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN3* attribute), 93  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN4* attribute), 94  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN5* attribute), 94  
 twostep\_parameters  
     (*fints.formals.ParameterTwostepTAN6* attribute), 94  
 TwoStepParameters1 (class in *fints.formals*), 86  
 TwoStepParameters2 (class in *fints.formals*), 87  
 TwoStepParameters3 (class in *fints.formals*), 88  
 TwoStepParameters4 (class in *fints.formals*), 89  
 TwoStepParameters5 (class in *fints.formals*), 90  
 TwoStepParameters6 (class in *fints.formals*), 92  
 TwoStepParametersCommon (class in *fints.formals*), 86  
 type (*fints.fields.AlphanumericField* attribute), 77  
 type (*fints.fields.AmountField* attribute), 78  
 type (*fints.fields.BinaryField* attribute), 78  
 type (*fints.fields.BooleanField* attribute), 78  
 type (*fints.fields.CodeField* attribute), 78  
 type (*fints.fields.CountryField* attribute), 78  
 type (*fints.fields.CurrencyField* attribute), 78  
 type (*fints.fields.DateField* attribute), 78  
 type (*fints.fields.DigitsField* attribute), 78  
 type (*fints.fields.DTAUSField* attribute), 77  
 type (*fints.fields.FloatField* attribute), 78  
 type (*fints.fields.GenericField* attribute), 77  
 type (*fints.fields.GenericGroupField* attribute), 77  
 type (*fints.fields.IDField* attribute), 78  
 type (*fints.fields.IntCodeField* attribute), 78  
 type (*fints.fields.NumericField* attribute), 77  
 type (*fints.fields.PasswordField* attribute), 78  
 type (*fints.fields.SegmentSequenceField* attribute), 79  
 type (*fints.fields.TextField* attribute), 77  
 type (*fints.fields.TimeField* attribute), 78  
 type (*fints.fields.ZeroPaddedNumericField* attribute), 77  
 type (*fints.formals.SegmentHeader* attribute), 79  
 TYPE (*fints.segments.accounts.HISPA1* attribute), 29  
 TYPE (*fints.segments.accounts.HISPAS1* attribute), 30  
 TYPE (*fints.segments.accounts.HKSPA1* attribute), 30  
 TYPE (*fints.segments.auth.HIPINS1* attribute), 30  
 TYPE (*fints.segments.auth.HITAB4* attribute), 31  
 TYPE (*fints.segments.auth.HITAB5* attribute), 31  
 TYPE (*fints.segments.auth.HITAN2* attribute), 31



- TYPE (*fints.segments.auth.HITAN3* attribute), 32
- TYPE (*fints.segments.auth.HITAN5* attribute), 33
- TYPE (*fints.segments.auth.HITAN6* attribute), 33
- TYPE (*fints.segments.auth.HITANS1* attribute), 34
- TYPE (*fints.segments.auth.HITANS2* attribute), 34
- TYPE (*fints.segments.auth.HITANS3* attribute), 34
- TYPE (*fints.segments.auth.HITANS4* attribute), 35
- TYPE (*fints.segments.auth.HITANS5* attribute), 35
- TYPE (*fints.segments.auth.HITANS6* attribute), 35
- TYPE (*fints.segments.auth.HITANSBase* attribute), 35
- TYPE (*fints.segments.auth.HKIDN2* attribute), 35
- TYPE (*fints.segments.auth.HKTAB4* attribute), 36
- TYPE (*fints.segments.auth.HKTAB5* attribute), 36
- TYPE (*fints.segments.auth.HKTAN2* attribute), 36
- TYPE (*fints.segments.auth.HKTAN3* attribute), 37
- TYPE (*fints.segments.auth.HKTAN5* attribute), 38
- TYPE (*fints.segments.auth.HKTAN6* attribute), 39
- TYPE (*fints.segments.auth.HKVVB3* attribute), 40
- TYPE (*fints.segments.bank.HIBPA3* attribute), 41
- TYPE (*fints.segments.bank.HIKOM4* attribute), 41
- TYPE (*fints.segments.bank.HIUPA4* attribute), 42
- TYPE (*fints.segments.bank.HIUPD6* attribute), 42
- TYPE (*fints.segments.bank.HKKOM4* attribute), 43
- TYPE (*fints.segments.base.FinTS3Segment* attribute), 29
- TYPE (*fints.segments.base.ParameterSegment* attribute), 44
- TYPE (*fints.segments.base.ParameterSegment\_22* attribute), 44
- TYPE (*fints.segments.debit.BatchDebitBase* attribute), 44
- TYPE (*fints.segments.debit.DebitResponseBase* attribute), 45
- TYPE (*fints.segments.debit.HIDBS1* attribute), 45
- TYPE (*fints.segments.debit.HIDBS2* attribute), 46
- TYPE (*fints.segments.debit.HIDBSS1* attribute), 46
- TYPE (*fints.segments.debit.HIDBSS2* attribute), 47
- TYPE (*fints.segments.debit.HIDMB1* attribute), 47
- TYPE (*fints.segments.debit.HIDMBS1* attribute), 48
- TYPE (*fints.segments.debit.HIDMC1* attribute), 48
- TYPE (*fints.segments.debit.HIDMCS1* attribute), 48
- TYPE (*fints.segments.debit.HIDME1* attribute), 48
- TYPE (*fints.segments.debit.HIDME2* attribute), 48
- TYPE (*fints.segments.debit.HIDMES1* attribute), 49
- TYPE (*fints.segments.debit.HIDMES2* attribute), 49
- TYPE (*fints.segments.debit.HIDSC1* attribute), 49
- TYPE (*fints.segments.debit.HIDSCS1* attribute), 49
- TYPE (*fints.segments.debit.HIDSE1* attribute), 50
- TYPE (*fints.segments.debit.HIDSE2* attribute), 50
- TYPE (*fints.segments.debit.HIDSES1* attribute), 50
- TYPE (*fints.segments.debit.HIDSES2* attribute), 50
- TYPE (*fints.segments.debit.HKDBS1* attribute), 50
- TYPE (*fints.segments.debit.HKDBS2* attribute), 51
- TYPE (*fints.segments.debit.HKDMB1* attribute), 52
- TYPE (*fints.segments.debit.HKDMC1* attribute), 52
- TYPE (*fints.segments.debit.HKDME1* attribute), 52
- TYPE (*fints.segments.debit.HKDME2* attribute), 53
- TYPE (*fints.segments.debit.HKDSC1* attribute), 53
- TYPE (*fints.segments.debit.HKDSE1* attribute), 53
- TYPE (*fints.segments.debit.HKDSE2* attribute), 54
- TYPE (*fints.segments.depot.HIWPD5* attribute), 54
- TYPE (*fints.segments.depot.HIWPD6* attribute), 54
- TYPE (*fints.segments.depot.HKWPD5* attribute), 55
- TYPE (*fints.segments.depot.HKWPD6* attribute), 55
- TYPE (*fints.segments.dialog.HIRMG2* attribute), 56
- TYPE (*fints.segments.dialog.HIRMS2* attribute), 56
- TYPE (*fints.segments.dialog.HISYN4* attribute), 56
- TYPE (*fints.segments.dialog.HKEND1* attribute), 57
- TYPE (*fints.segments.dialog.HKSYN3* attribute), 57
- TYPE (*fints.segments.journal.HIPRO3* attribute), 57
- TYPE (*fints.segments.journal.HIPRO4* attribute), 58
- TYPE (*fints.segments.journal.HIPROS3* attribute), 58
- TYPE (*fints.segments.journal.HIPROS4* attribute), 58
- TYPE (*fints.segments.journal.HKPRO3* attribute), 58
- TYPE (*fints.segments.journal.HKPRO4* attribute), 59
- TYPE (*fints.segments.message.HNHBK3* attribute), 59
- TYPE (*fints.segments.message.HNHBS1* attribute), 60
- TYPE (*fints.segments.message.HNSHA2* attribute), 60
- TYPE (*fints.segments.message.HNSHK4* attribute), 60
- TYPE (*fints.segments.message.HNVSD1* attribute), 62
- TYPE (*fints.segments.message.HNVSK3* attribute), 62
- TYPE (*fints.segments.saldo.HISAL5* attribute), 63
- TYPE (*fints.segments.saldo.HISAL6* attribute), 64
- TYPE (*fints.segments.saldo.HISAL7* attribute), 65
- TYPE (*fints.segments.saldo.HKSAL5* attribute), 66
- TYPE (*fints.segments.saldo.HKSAL6* attribute), 66
- TYPE (*fints.segments.saldo.HKSAL7* attribute), 67
- TYPE (*fints.segments.statement.DIKKU2* attribute), 67
- TYPE (*fints.segments.statement.DIKKUS2* attribute), 67
- TYPE (*fints.segments.statement.DKKKU2* attribute), 67
- TYPE (*fints.segments.statement.HICAZ1* attribute), 68
- TYPE (*fints.segments.statement.HIKAZ5* attribute), 69
- TYPE (*fints.segments.statement.HIKAZ6* attribute), 69
- TYPE (*fints.segments.statement.HIKAZ7* attribute), 69
- TYPE (*fints.segments.statement.HKCAZ1* attribute), 70
- TYPE (*fints.segments.statement.HKKAZ5* attribute), 70
- TYPE (*fints.segments.statement.HKKAZ6* attribute), 71
- TYPE (*fints.segments.statement.HKKAZ7* attribute), 72
- TYPE (*fints.segments.transfer.HICCMS1* attribute), 72
- TYPE (*fints.segments.transfer.HKCCM1* attribute), 72
- TYPE (*fints.segments.transfer.HKCCS1* attribute), 73
- TYPE (*fints.segments.transfer.HKIPM1* attribute), 73
- TYPE (*fints.segments.transfer.HKIPZ1* attribute), 74
- TypedField (class in *fints.types*), 77
- ## U
- UPD\_CONCLUSIVE (*fints.formals.UPDUsage* attribute), 98
- UPD\_INCONCLUSIVE (*fints.formals.UPDUsage* attribute), 98

- `upd_usage` (*fints.segments.bank.HIUPA4 attribute*), 42
  - `upd_version` (*fints.segments.auth.HKVVB3 attribute*), 40
  - `upd_version` (*fints.segments.bank.HIUPA4 attribute*), 42
  - `UPDUsage` (class in *fints.formals*), 98
  - `usage_encryption` (*fints.formals.EncryptionAlgorithm attribute*), 81
  - `usage_hash` (*fints.formals.HashAlgorithm attribute*), 81
  - `usage_signature` (*fints.formals.SignatureAlgorithm attribute*), 82
  - `UsageEncryption` (class in *fints.formals*), 80
  - `used_amount` (*fints.segments.saldo.HISAL5 attribute*), 63
  - `used_amount` (*fints.segments.saldo.HISAL6 attribute*), 64
  - `used_amount` (*fints.segments.saldo.HISAL7 attribute*), 65
  - `user_defined_signature` (*fints.segments.message.HNSHA2 attribute*), 60
  - `user_id` (*fints.formals.KeyName attribute*), 82
  - `user_id_field_text` (*fints.formals.ParameterPinTan attribute*), 94
  - `user_identifier` (*fints.segments.bank.HIUPA4 attribute*), 42
  - `UserDefinedSignature` (class in *fints.formals*), 83
  - `username` (*fints.segments.bank.HIUPA4 attribute*), 42
- ## V
- `V` (*fints.formals.KeyType attribute*), 82
  - `valid_from` (*fints.formals.TANMedia4 attribute*), 102
  - `valid_from` (*fints.formals.TANMedia5 attribute*), 103
  - `valid_until` (*fints.formals.TANMedia4 attribute*), 102
  - `valid_until` (*fints.formals.TANMedia5 attribute*), 103
  - `validation_result` (*fints.segments.message.HNSHA2 attribute*), 60
  - `ValueList` (class in *fints.types*), 77
  - `version` (*fints.formals.SegmentHeader attribute*), 79
  - `VERSION` (*fints.formals.TwoStepParametersCommon attribute*), 86
  - `VERSION` (*fints.segments.accounts.HISPA1 attribute*), 30
  - `VERSION` (*fints.segments.accounts.HISPAS1 attribute*), 30
  - `VERSION` (*fints.segments.accounts.HKSPA1 attribute*), 30
  - `VERSION` (*fints.segments.auth.HIPINS1 attribute*), 30
  - `VERSION` (*fints.segments.auth.HITAB4 attribute*), 31
  - `VERSION` (*fints.segments.auth.HITAB5 attribute*), 31
  - `VERSION` (*fints.segments.auth.HITAN2 attribute*), 31
  - `VERSION` (*fints.segments.auth.HITAN3 attribute*), 32
  - `VERSION` (*fints.segments.auth.HITAN5 attribute*), 33
  - `VERSION` (*fints.segments.auth.HITAN6 attribute*), 33
  - `VERSION` (*fints.segments.auth.HITANS1 attribute*), 34
  - `VERSION` (*fints.segments.auth.HITANS2 attribute*), 34
  - `VERSION` (*fints.segments.auth.HITANS3 attribute*), 34
  - `VERSION` (*fints.segments.auth.HITANS4 attribute*), 35
  - `VERSION` (*fints.segments.auth.HITANS5 attribute*), 35
  - `VERSION` (*fints.segments.auth.HITANS6 attribute*), 35
  - `VERSION` (*fints.segments.auth.HITANSBase attribute*), 35
  - `VERSION` (*fints.segments.auth.HKIDN2 attribute*), 35
  - `VERSION` (*fints.segments.auth.HKTAB4 attribute*), 36
  - `VERSION` (*fints.segments.auth.HKTAB5 attribute*), 36
  - `VERSION` (*fints.segments.auth.HKTAN2 attribute*), 36
  - `VERSION` (*fints.segments.auth.HKTAN3 attribute*), 37
  - `VERSION` (*fints.segments.auth.HKTAN5 attribute*), 38
  - `VERSION` (*fints.segments.auth.HKTAN6 attribute*), 39
  - `VERSION` (*fints.segments.auth.HKVVB3 attribute*), 40
  - `VERSION` (*fints.segments.bank.HIBPA3 attribute*), 41
  - `VERSION` (*fints.segments.bank.HIKOM4 attribute*), 41
  - `VERSION` (*fints.segments.bank.HIUPA4 attribute*), 42
  - `VERSION` (*fints.segments.bank.HIUPD6 attribute*), 42
  - `VERSION` (*fints.segments.bank.HKKOM4 attribute*), 43
  - `VERSION` (*fints.segments.base.FinTS3Segment attribute*), 29
  - `VERSION` (*fints.segments.base.ParameterSegment attribute*), 44
  - `VERSION` (*fints.segments.base.ParameterSegment\_22 attribute*), 44
  - `VERSION` (*fints.segments.debit.BatchDebitBase attribute*), 44
  - `VERSION` (*fints.segments.debit.DebitResponseBase attribute*), 45
  - `VERSION` (*fints.segments.debit.HIDBS1 attribute*), 45
  - `VERSION` (*fints.segments.debit.HIDBS2 attribute*), 46
  - `VERSION` (*fints.segments.debit.HIDBSS1 attribute*), 47
  - `VERSION` (*fints.segments.debit.HIDBSS2 attribute*), 47
  - `VERSION` (*fints.segments.debit.HIDMB1 attribute*), 47
  - `VERSION` (*fints.segments.debit.HIDMBS1 attribute*), 48
  - `VERSION` (*fints.segments.debit.HIDMC1 attribute*), 48
  - `VERSION` (*fints.segments.debit.HIDMCS1 attribute*), 48
  - `VERSION` (*fints.segments.debit.HIDME1 attribute*), 48
  - `VERSION` (*fints.segments.debit.HIDME2 attribute*), 48
  - `VERSION` (*fints.segments.debit.HIDMES1 attribute*), 49
  - `VERSION` (*fints.segments.debit.HIDMES2 attribute*), 49
  - `VERSION` (*fints.segments.debit.HIDSC1 attribute*), 49
  - `VERSION` (*fints.segments.debit.HIDSCS1 attribute*), 49
  - `VERSION` (*fints.segments.debit.HIDSE1 attribute*), 50
  - `VERSION` (*fints.segments.debit.HIDSE2 attribute*), 50
  - `VERSION` (*fints.segments.debit.HIDSES1 attribute*), 50
  - `VERSION` (*fints.segments.debit.HIDSES2 attribute*), 50
  - `VERSION` (*fints.segments.debit.HKDBS1 attribute*), 50
  - `VERSION` (*fints.segments.debit.HKDBS2 attribute*), 51
  - `VERSION` (*fints.segments.debit.HKDMB1 attribute*), 52

- VERSION (*fints.segments.debit.HKDMC1* attribute), 52  
 VERSION (*fints.segments.debit.HKDME1* attribute), 52  
 VERSION (*fints.segments.debit.HKDME2* attribute), 53  
 VERSION (*fints.segments.debit.HKDSC1* attribute), 53  
 VERSION (*fints.segments.debit.HKDSE1* attribute), 53  
 VERSION (*fints.segments.debit.HKDSE2* attribute), 54  
 VERSION (*fints.segments.depot.HIWPD5* attribute), 54  
 VERSION (*fints.segments.depot.HIWPD6* attribute), 54  
 VERSION (*fints.segments.depot.HKWPD5* attribute), 55  
 VERSION (*fints.segments.depot.HKWPD6* attribute), 55  
 VERSION (*fints.segments.dialog.HIRMG2* attribute), 56  
 VERSION (*fints.segments.dialog.HIRMS2* attribute), 56  
 VERSION (*fints.segments.dialog.HISYN4* attribute), 56  
 VERSION (*fints.segments.dialog.HKEND1* attribute), 57  
 VERSION (*fints.segments.dialog.HKSYN3* attribute), 57  
 VERSION (*fints.segments.journal.HIPRO3* attribute), 57  
 VERSION (*fints.segments.journal.HIPRO4* attribute), 58  
 VERSION (*fints.segments.journal.HIPROS3* attribute), 58  
 VERSION (*fints.segments.journal.HIPROS4* attribute), 58  
 VERSION (*fints.segments.journal.HKPRO3* attribute), 58  
 VERSION (*fints.segments.journal.HKPRO4* attribute), 59  
 VERSION (*fints.segments.message.HNHBK3* attribute), 59  
 VERSION (*fints.segments.message.HNHBS1* attribute), 60  
 VERSION (*fints.segments.message.HNSHA2* attribute), 60  
 VERSION (*fints.segments.message.HNSHK4* attribute), 61  
 VERSION (*fints.segments.message.HNVSD1* attribute), 62  
 VERSION (*fints.segments.message.HNVSK3* attribute), 62  
 VERSION (*fints.segments.saldo.HISAL5* attribute), 63  
 VERSION (*fints.segments.saldo.HISAL6* attribute), 64  
 VERSION (*fints.segments.saldo.HISAL7* attribute), 65  
 VERSION (*fints.segments.saldo.HKSAL5* attribute), 66  
 VERSION (*fints.segments.saldo.HKSAL6* attribute), 66  
 VERSION (*fints.segments.saldo.HKSAL7* attribute), 67  
 VERSION (*fints.segments.statement.DIKKU2* attribute), 67  
 VERSION (*fints.segments.statement.DIKKUS2* attribute), 67  
 VERSION (*fints.segments.statement.DKKKU2* attribute), 67  
 VERSION (*fints.segments.statement.HICAZ1* attribute), 68  
 VERSION (*fints.segments.statement.HIKAZ5* attribute), 69  
 VERSION (*fints.segments.statement.HIKAZ6* attribute), 69  
 VERSION (*fints.segments.statement.HIKAZ7* attribute), 69  
 VERSION (*fints.segments.statement.HKCAZ1* attribute), 70  
 VERSION (*fints.segments.statement.HKKAZ5* attribute), 70  
 VERSION (*fints.segments.statement.HKKAZ6* attribute), 71  
 VERSION (*fints.segments.statement.HKKAZ7* attribute), 72  
 VERSION (*fints.segments.transfer.HICCMS1* attribute), 72  
 VERSION (*fints.segments.transfer.HKCCM1* attribute), 72  
 VERSION (*fints.segments.transfer.HKCCS1* attribute), 73  
 VERSION (*fints.segments.transfer.HKIPM1* attribute), 73  
 VERSION (*fints.segments.transfer.HKIPZ1* attribute), 74  
 version\_info\_chiptan (*fints.formals.ResponseHHDUC* attribute), 104  
 versions (*fints.formals.SupportedHBCIVersions2* attribute), 95
- ## W
- WIT (*fints.formals.SecurityRole* attribute), 97
- ## Y
- YES (*fints.formals.TANListNumberRequired* attribute), 85
- ## Z
- ZeroPaddedNumericField (class in *fints.fields*), 77  
 ZIP (*fints.formals.CompressionFunction* attribute), 97  
 zka\_id (*fints.formals.TwoStepParameters4* attribute), 89  
 zka\_id (*fints.formals.TwoStepParameters5* attribute), 90  
 zka\_id (*fints.formals.TwoStepParameters6* attribute), 92  
 zka\_version (*fints.formals.TwoStepParameters4* attribute), 89  
 zka\_version (*fints.formals.TwoStepParameters5* attribute), 90  
 zka\_version (*fints.formals.TwoStepParameters6* attribute), 92  
 ZZZ (*fints.formals.CompressionFunction* attribute), 97  
 ZZZ (*fints.formals.OperationMode* attribute), 81