

Tesch mediafinanz GmbH

Weiße Breite 5
49084 Osnabrück
Germany

TESCH mediafinanz 
PART OF LOWELL GROUP

SOAP web service

Description of the
Encashment SOAP interface
WSDL version 210

Contents

1	Definition	4
2	Target group	4
3	Introduction	4
4	Prerequisites	4
4.1	Client account	4
4.2	Conditions of use	4
4.3	Linking	4
4.4	Application license key	5
4.5	Client license key	5
5	WSDL	5
6	Character encoding	5
7	Authentication	5
8	Sandbox mode/production mode	6
9	Typographical conventions	6
10	Functions	7
10.1	bookDirectPayment()	7
10.2	closeClaim()	7
10.3	commitTransaction()	8
10.4	enableLongTermObservation()	8
10.5	getClaimAccountingChanges()	8
10.6	getClaimAccountingSummary()	9
10.7	getClaimHistory()	9
10.8	getClaimOptions()	9
10.9	getClaimStatus()	9
10.10	getClaimStatusChanges()	10
10.11	getMessageHistory()	10
10.12	getNewMessages()	10
10.13	getPayoutClaimList()	10
10.14	getPayoutList()	11
10.15	newClaim()	11
10.16	sendMessage()	12
10.17	updateAnnotation()	12
11	Data types	12
11.1	AdditionalAddress	12
11.2	ArrayOfClaimHistoryChange	13
11.3	ArrayOfClaimHistoryEntry	13
11.4	ArrayOfClaimMessage	13
11.5	ArrayOfClaimOption	13

11.6	ArrayOfClaimStatusChange	13
11.7	ArrayOfError	13
11.8	ArrayOfFileNumber	13
11.9	ArrayOfNewClaimMessage	13
11.10	ArrayOfPayoutSummary	13
11.11	ArrayOfPayoutClaim	14
11.12	Auth	14
11.13	Claim	15
11.14	ClaimAccountingChanges	16
11.15	ClaimAccountingSummary	17
11.16	ClaimHistoryChange	17
11.17	ClaimHistoryEntry	17
11.18	ClaimIdentifier	18
11.19	ClaimMessage	18
11.20	ClaimStatus	18
11.21	ClaimStatusChange	19
11.22	ClaimStatusChanges	19
11.23	Debtor	19
11.24	DirectPayment	20
11.25	Money	20
11.26	NewClaimMessage	20
11.27	NewClaimMessages	21
11.28	NewClaimResult	21
11.29	PayoutClaim	21
11.30	PayoutSummary	22
12	Error codes	23
12.1	General errors	23
12.2	Authentication errors	24
12.3	Function errors	26
13	Status list	28
14	Contact details for questions	29

1 Definition

SOAP (*initially: Simple Object Access Protocol*) is a network protocol which allows data to be exchanged between systems, and remote procedure calls to be performed. SOAP is based on the services of other standards: XML to represent the data, and Internet protocols of the transport and application layer (see TCP/IP reference model) to transmit messages. The latest combination is SOAP via HTTP and TCP. Officially, the abbreviation SOAP has not been used as an acronym since version 1.2, as it is now a name in its own right.

2 Target group

This documentation is intended for software developers who want to implement a connection to the Tesch mediafinanz GmbH – hereinafter ‘mediafinanz’ – web service. To understand the WSDL document being used, which describes the available functions and data types, knowledge of WSDL and XML schemas is advantageous.

3 Introduction

The mediafinanz SOAP interface offers automated access to the functions and services which are also available via the mediafinanz client online system. Above all, this includes easy management of debt collection orders and the performing of creditworthiness checks. By connecting our interfaces, many of the functions offered by mediafinanz can be used directly from your application. This allows your business processes to be extended cost-effectively and easily to include the areas of debt collection and risk management.

This document describes the mediafinanz **debt collection/receivables management** interface. You will find additional information in our client online system on **performing creditworthiness checks**.

4 Prerequisites

4.1 Client account

Your customer needs an activated mediafinanz client account to use the interface.

4.2 Conditions of use

To integrate the mediafinanz SOAP interface into your software product, you must agree to these conditions of use. These primarily concern your willingness to adapt your application promptly to interface modifications and extensions. You can view the conditions of use in the client online system (click [Settings > Interfaces](#), [SOAP](#) tab).

4.3 Linking

Interface requests triggered by an application always relate to exactly one mediafinanz client account, which must be activated beforehand for use by the application being used (linking). If you merely want to integrate the interface into your application for your own purposes, and you do not intend additional mediafinanz clients to use your application, we can perform the linking for you. However, if you intend to make your application accessible to other clients, you can perform the linking as follows:

1. (*General case*): mediafinanz will provide you with a list of registration keys. You can forward each registration key to your application users. Linking is performed by entering a valid registration key in our client online system (menu [Options > Settings > Interfaces](#), [SOAP](#) tab).

2. (*Special case*): If you have entered into a separate collaboration agreement with mediafinanz, the accounts of the clients assigned to you are already linked automatically to your application. To use the interface via your application, all customers have to do is grant their approval once in the client online system. (Menu [Options > Settings > Interfaces, Partner](#) tab).

4.4 Application license key

An application license key is required for each application communicating using the mediafinanz interface. This authorises the application to direct requests to the mediafinanz interface. Exactly one application licence is required per application, irrespective of how many clients use the application. Please contact our IT department (itsupport@mediafinanz.de) to obtain the application license key for your software.

4.5 Client license key

Not applicable to collaborative partners: The client license key identifies the requesting client. Once linked, this license key can also be viewed in the mediafinanz client online system (menu [Options> Settings> Interfaces, SOAP](#) tab), and is typically transferred to the application being used via copy & paste.

5 WSDL

The encashment web service is fully specified by a WSDL (Web Service Description Language) document, which is the focus of this documentation. To understand how the web service operates, it is worthwhile taking a close look at the WSDL document.

It can be called via the following link using the style *Rpc/Encoded*:

<https://soap.mediafinanz.de/encashment210.wsdl>

A compatible WSDL file in the style *Document/Literal* can be found here:

<https://soap.mediafinanz.de/encashmentLiteral210.wsdl>

6 Character encoding

Data submitted in the SOAP request must be UTF-8 encoded. Likewise, responses are returned encoded in UTF-8.

7 Authentication

Each request is authenticated, whereby the permission of the client and the application used is checked. Authentication is performed based on the Auth parameter, which may comprise two methods:

- ✎ **Client authentication** (*general case*): The request is authorised by stating the client ID (`clientId`) and a request license key (`licenceKey`). The request license key is formed from the application license and client license (see 11.12 Auth data type).
- ✎ **Partner authentication** (*special case*): If a separate collaboration agreement has been entered into with mediafinanz, a mediafinanz collaborative partner may authorise a SOAP request even without knowing the client ID and the client license key. To do this, instead of the client ID, the client's unique customer number, under which the client is listed in the collaborative partner's database (hereinafter: partner customer ID), is specified as the `clientId`. The request licence key (`licenceKey`) is formed from the partner customer ID and the partner licence key (see 11.12 Auth data type).

8 Sandbox mode/production mode

For each request, it can be set whether it is to be executed in test or production mode using the element `sandbox` (type `boolean`) of the data type `Auth`.

During the development and test period, the `sandbox` element should always be set to `TRUE` to execute requests exclusively in test mode. Once implemented successfully, you can then set `sandbox` to `FALSE`. From then on, all requests will be executed in production mode. If you receive the error message 'function call not yet permitted in non-sandbox-mode' when in production mode, please contact mediafinanz technical support as mediafinanz needs to approve the function.

If a number of mediafinanz clients use your application, it is best to allow the user to decide whether test or production mode is used. Please check the extent to which you want to make this setting available to your users, such as via a configuration screen, INI file or registry key, etc.

Please note: The client online system uses the production database exclusively. Interface requests executed in test mode can therefore not be checked in the client online system.

9 Typographical conventions

The following typographical conventions are used in this documentation:

- ⌘ Complex data types and their elements: `Auth` `sandbox`
- ⌘ Primitive data types: `integer`
- ⌘ Optional parameters and elements: `[Auth]` `[integer]`
- ⌘ Functions: `sendMessage()`
- ⌘ Nested elements: `L date`

10 Functions

Generally, you will receive the specified return value from each function call. However, in the event of a fault, a SOAP fault will be triggered containing an error code and a short description. As standard, SOAP libraries may report SOAP faults as exceptions.

10.1 *bookDirectPayment()*

Description	Informs mediafinanz about a (partial) payment made directly by the debtor to the creditor. The amount received is booked as a credit, and reminder proceedings are continued for any remaining amount still open. In the DirectPayment parameter, please state the date payment was received together with the exact amount that has been received. If the direct payment includes full or partial settlement of the debt collection fees charged by mediafinanz, you will receive an invoice for these from mediafinanz. If the direct payment is made exclusive of full settlement of the debt collection fees, mediafinanz will continue to claim for the remaining amount against your debtor.			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim
	3	DirectPayment	directPayment	Details on the direct payment received

10.2 *closeClaim()*

Description	This function can be called to prematurely close a claim already submitted to mediafinanz without specifying the reasons or to prematurely close debt collection proceedings. If mediafinanz has not yet started debt collection proceedings, they can be closed free of charge. Otherwise, mediafinanz will charge a fixed fee of €6 plus VAT. If you want to close proceedings as your debtor has paid you directly, please use the bookDirectPayment() function instead.			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.3 *commitTransaction()*

Description	Several functions include a transaction ID in the return value (<code>transactionId</code>). Once these functions' return values been processed successfully, the transaction needs to be confirmed using the <code>commitTransaction()</code> function.			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	<code>string</code>	transactionId	Transaction ID

10.4 *enableLongTermObservation()*

Description	Transfers a debt collection claim to long-term observation, if possible. The <code>getClaimOptions()</code> function specifies whether a claim can be transferred to long-term observation.			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.5 *getClaimAccountingChanges()*

Description	Returns a list with file numbers where changes have been made to the booking account (e.g. payment receipts, return debit notes, etc.) since the function was last called. The changes themselves are not reported using this function – for this purpose, the <code>getClaimAccountingSummary()</code> function (see below) is available, which can be called in a loop for all file numbers concerned. Be aware that no file numbers will be returned when this function is first called.			
Return type	ClaimAccountingChanges			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter

10.6 *getClaimAccountingSummary()*

Description	Returns detailed information relating to the accounting ledger of a debt collection claim, including the total debt, amount already paid, remaining amount still open, the current payout amount and the payout history.			
Return type	ClaimAccountingSummary			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.7 *getClaimHistory()*

Description	Returns the complete history of the debt collection claim.			
Return type	ArrayOfClaimHistoryEntry			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.8 *getClaimOptions()*

Description	Returns the options currently available relating to a claim. The return value is an array which may currently include the following strings: <ul style="list-style-type: none"> ⌘ addressIdentification: An address inquiry may be performed. ⌘ longTermObservation: The claim can be transferred to long-term observation. ⌘ factoring: The claim can be offered for sale. ⌘ lawyer: The claim can be transferred to the contract firm to conduct statutory reminder proceedings. ⌘ close: The claim can be closed. 			
Return type	ArrayOfClaimOption			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.9 *getClaimStatus()*

Description	Returns detailed information about the current status of a debt collection claim. To keep track of the status of all your debt collection cases currently in the system, instead of calling this function periodically for all claims, use the function 10.10 <i>getClaimStatusChanges()</i> , which only returns the status changes since the last function call.			
Return type	ClaimStatus			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.10 *getClaimStatusChanges()*

Description	Returns all status changes since the last function call. Be aware that no status changes are returned when this function is first called.			
Return type	ClaimStatusChanges			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter

10.11 *getMessageHistory()*

Description	Returns all messages (to and from mediafinanz) relating to a debt collection claim.			
Return type	ArrayOfClaimMessage			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim

10.12 *getNewMessages()*

Description	Returns new messages relating to debt collection claims.			
Return type	NewClaimMessages			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter

10.13 *getPayoutClaimList()*

Description	Returns debt collection claims included in a payout. Besides the debtor information, file number and your invoice number, current accounting details are also returned. In other words, this function returns exactly the same data which you are also provided with in each emailed/posted debt collection payout.			
Return type	ArrayOfPayoutClaim			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	string	payoutNumber	Payout number

10.14 *getPayoutList()*

Description	Returns a list of all available debt collection payouts. The debt collection claims included in a payout can be called using the above-mentioned <code>getPayoutClaimList()</code> function.			
Return type	ArrayOfPayoutSummary			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter

10.15 *newClaim()*

Description	Use this function to submit a new debt collection claim to mediafinanz. If successful, you will receive the file number assigned by mediafinanz as the return value. If validation errors occur, these will be returned in the form of the ArrayOfError data type. Approval by mediafinanz is required before this function is used for the first time in production mode. To obtain this, please contact mediafinanz technical support.			
Return type	NewClaimResult			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	Claim	Claim	Claim information
	3	Debtor	debtor	Debtor information

10.16 *sendMessage()*

Description	Sends a message to mediafinanz regarding a debt collection claim. Messages from mediafinanz can be called using the above-mentioned <code>getNewMessages()</code> function.			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim
3	<code>string</code>	message	Message text	

10.17 *updateAnnotation()*

Description	<p>Updates the short note saved relating to a debt collection claim.</p> <p>To process claims, it can be useful to provide the mediafinanz agent with additional information. The <code>note</code> element in the Claim data type is provided for this purpose. It can be transferred when submitting a debt collection claim using <code>newClaim()</code>. Information submitted using the <code>note</code> element should, however, be kept as general as possible as it cannot be changed afterwards.</p> <p>This function is available for notes which may need to be updated once a claim has been submitted. If you submit a debt collection claim immediately after a bounced invoice, but you initially allow the contract to continue to run with your customer, you can use this function such as to inform mediafinanz about the current contract status ('active contract'). If you then terminate the contract with your customer during the ongoing debt collection proceedings, you can change the short note such as to 'contract terminated'.</p> <p>This function should only be used after consultation with mediafinanz.</p>			
Return type	<code>boolean</code>			
Parameter	No.	Type	Name	Note
	1	Auth	auth	Authentication parameter
	2	ClaimIdentifier	claimIdentifier	Identifies the claim
3	<code>string</code>	annotation	Note text	

11 Data types

11.1 *AdditionalAddress*

Debtor's additional address. If you have an additional delivery address for your claim, you

can provide this when submitting your claim using the newClaim() function in the Debtor parameter.

Type	Name	Note
string	co	Optional c/o address of the debtor (e.g. 'c/o Smith')
string	street	Number and street
string	postcode	Postcode
string	City	Location
string	country	ISO 3166 country code (e.g. DE for Germany, AT for Austria, CH for Switzerland)
[integer]	addressStatus	Optional. Status of debtor's postal address. Use only after consultation with mediafinanz.

11.2 ArrayOfClaimHistoryChange

Array with elements of ClaimHistoryChange type.

11.3 ArrayOfClaimHistoryEntry

Array with elements of ClaimHistoryEntry type.

11.4 ArrayOfClaimMessage

Array with elements of ClaimMessage type.

11.5 ArrayOfClaimOption

Array with elements of string type.

11.6 ArrayOfClaimStatusChange

Array with elements of ClaimStatusChange type.

11.7 ArrayOfError

This data type is used if multiple errors need to be reported in one go, such as because several mandatory fields were not populated. The data type can include as many error strings as required. Typically, the SOAP library used by you converts this element into an array.

11.8 ArrayOfFileNumber

Array with elements of string type.

11.9 ArrayOfNewClaimMessage

Array with elements of NewClaimMessage type.

11.10 ArrayOfPayoutSummary

Array with elements of PayoutSummary type.

11.11 *ArrayOfPayoutClaim*

Array with elements of PayoutClaim type.

11.12 *Auth*

Authentication parameter. This data type typically comprises three elements:

Type	Name	Note
<code>integer</code>	<code>clientId</code>	mediafinanz client ID (customer no.)
<code>string</code>	<code>licenceKey</code>	<p>The request licence key used to authorise the request. It is formed as follows from the application licence key of the used application and the client license key:</p> <ul style="list-style-type: none">⌘ The client license key (CL) is appended to the application license key (AL).⌘ An MD5 hash is produced from the resulting 64-character string. The result of the hash function is the 32-character <code>licenceKey</code>, which can be used to authorise the request. <p><code>licenceKey := MD5(AL + CL)</code></p>
<code>boolean</code>	<code>sandbox</code>	Specifies whether the request is to be executed in sandbox or production mode (see 8, Sandbox mode/production mode)

If a separate collaboration agreement has been entered into with mediafinanz, the `Auth` parameter comprises four elements:

Type	Name	Note
<code>integer</code>	<code>partnerId</code>	The partner ID assigned by mediafinanz for the collaborative partner
<code>string</code>	<code>clientId</code>	The customer number under which the client is listed in the collaborative partner's database (partner customer ID)
<code>string</code>	<code>licenceKey</code>	<p>The request licence key used to authorise the request. It is formed as follows from the partner customer ID and the partner license key:</p> <ul style="list-style-type: none">⌘ The partner license key (PL) is appended to the partner customer ID (PCID).⌘ An MD5 hash is produced from the resulting character string. The result of the hash function is the 32-character <code>licenceKey</code>, which can be used to authorise the request. <p><code>licenceKey := MD5(PCID + PL)</code></p>
<code>boolean</code>	<code>sandbox</code>	Specifies whether the request is to be executed in sandbox or production mode (see 8, Sandbox

mode/production mode)

11.13 Claim

Debt collection claim.

Type	Name	Note
[string]	invoice	Optional. The number of the unpaid invoice. This should be unique. The claim can then be clearly identified later on using the invoice number (see 11.18 ClaimIdentifier).
integer	type	Indicates the type of claim concerned: <ul style="list-style-type: none"> ⌘ 1: Goods were sold ⌘ 2: Goods were sold on a prepayment basis ⌘ 3: A service was provided
string	reason	Description of service provided/goods sold relating to the claim (e.g. '1 x Siemens S55 mobile phone'). Important: The actual service provided/goods sold should be listed here using key words. The text submitted will be copied word for word to the reminder notice, so it is limited to 250 characters.
Money	originalValue	Original claim value (excluding reminder fees)
Money	overdueFees	Claim reminder fees
[Money]	returnDebitNoteFees	Optional. Debtor's return debit note fees.
date	dateOfOrigin	Date of the provision of service/sale of goods relating to the claim (e.g. delivery date) in the format YYYY-MM-DD
date	dateOfLastReminder	Date of most recent reminder in the format YYYY-MM-DD
string	note	Note about the claim. Is displayed to the mediafinanz agent.
[integer]	profile	Optional. Controls the type of dunning run to be used. Use only after consultation with mediafinanz.
[string]	annotation	Optional. Short note (e.g. 'Active customer') that can be updated by the client at any time. See also function 10.17 updateAnnotation().
[date]	contractDate	Optional. Contract date for ongoing contracts/repeat services (e.g. Mobile phone contract, rental contract); DD.MM.YYYY or YYYY-MM-DD).
[string]	originalCreator	Optional. Original creditor (company in whose name the claim arose (in the event of assignment, purchase of receivables), company name and, if applicable, web address).
[integer]	catalogReason	Optional. Indicates the main reason for the claim from the following catalogue:

		<ul style="list-style-type: none"> ⌘ 40100: Newspaper notice(s) ⌘ 40101: Medical service ⌘ 40102: Consulting ⌘ 40103: Service ⌘ 40104: IT service/web design ⌘ 40105: Increased transport fee ⌘ 40106: Freight costs ⌘ 40107: Business activity provided by self-employed person ⌘ 40108: Tradesperson service ⌘ 40109: Hotel costs ⌘ 40110: Daycare centre contribution/dinner money ⌘ 40111: Patient transport costs ⌘ 40112: Course/teaching costs ⌘ 40113: Broker's commission ⌘ 40114: Rent ⌘ 40115: Membership ⌘ 40117: Lawyer's fee ⌘ 40118: Travel service ⌘ 40119: Repair service ⌘ 40120: Server rental ⌘ 40121: Tax advisor's fee ⌘ 40122: Transport costs ⌘ 40123: Telecommunications services ⌘ 40124: Veterinary service ⌘ 40125: Teaching costs ⌘ 40126: Agency activities ⌘ 40128: Goods delivery(ies) ⌘ 40129: Goods delivery on prepayment basis ⌘ 40130: Web hosting ⌘ 40131: Other claim ⌘ 40132: Dentistry service
[string]	catalogText	Optional. Other main claim catalogue reason (catalogReason is 40131).

11.14 **ClaimAccountingChanges**

Claims with account changes.

Type	Name	Note
string	transactionId	Transaction ID. Is required for the commitTransaction() function.
ArrayOfFileNumber	fileNumbers	Array with file numbers of claims with account changes

11.15 *ClaimAccountingSummary*

Summary of a claim account.

Type	Name	Note
Money	totalDebts	Total debts (original claim + reminder fees)
Money	paid	Amount already paid
Money	outstanding	Amount still outstanding
Money	currentPayout	Current payout amount
Money	sumPayout	Sum of all payouts to the client
[struct]	payoutHistory	Optional. Details of a payout to the client (can occur 0 to n times)
date	└ date	Date of payout
Money	└ total	Payout amount
string	└ payoutNumber	Payout number

11.16 *ClaimHistoryChange*

New entry added to the history of a debt collection claim.

Type	Name	Note
string	fileNumber	mediafinanz file number
dateTime	time	Time created
string	subject	Entry subject line
[string]	details	Optional. Entry details.

11.17 *ClaimHistoryEntry*

History entry for a debt collection claim.

Type	Name	Note
dateTime	time	Time created
string	subject	Entry subject line
[string]	details	Optional. Entry details.

11.18 *ClaimIdentifier*

Claim identifier. This data type is used when identifying a claim that has already been submitted, such as to query the status or cancel it. Either `invoice` or `fileNumber` must be specified.

Type	Name	Note
<code>string</code>	<code>invoice</code>	Invoice number specified when submitting the debt collection claim. The invoice number can only be used to identify a claim if it is unique; in other words, it has not been assigned to multiple claims.
<code>string</code>	<code>fileNumber</code>	The file number assigned by mediafinanz after a debt collection claim has been submitted

11.19 *ClaimMessage*

Message relating to a debt collection claim.

Type	Name	Note
<code>dateTime</code>	<code>time</code>	Time created
<code>string</code>	<code>subject</code>	Message subject line
<code>string</code>	<code>sender</code>	Message sender
<code>string</code>	<code>text</code>	Message text

11.20 *ClaimStatus*

Current status of a debt collection claim.

Type	Name	Note
<code>integer</code>	<code>statusCode</code>	Numerical status code
<code>string</code>	<code>statusText</code>	Status description
<code>[string]</code>	<code>statusDetails</code>	Optional. Contains text with additional details about the current status, such as details about payment agreements which have been entered into.

11.21 *ClaimStatusChange*

Debt collection claim status change.

Type	Name	Note
<code>string</code>	fileNumber	mediafinanz file number
<code>string</code>	invoiceNumber	Invoice number
<code>dateTime</code>	time	Time of status change
<code>integer</code>	statusCode	Internal status code
<code>string</code>	statusName	Status name
<code>[integer]</code>	closingReasonCode	Optional. Closing code.
<code>[string]</code>	closingReasonExplanation	Optional. Closing reason.
<code>[string]</code>	closingReasonNote	Optional. Note about the closing reason, such as file numbers in the case of consolidated claims.

11.22 *ClaimStatusChanges*

Debt collection claim status changes.

Type	Name	Note
<code>string</code>	transactionId	Transaction ID. Is required for the <code>commitTransaction()</code> function.
<code>ArrayOfClaimStatusChange</code>	changes	Array with status changes

11.23 *Debtor*

Debtor details.

Type	Name	Note
<code>[string]</code>	id	Optional. Debtor's unique customer number under which the debtor is listed in the client's database.
<code>string</code>	address	Debtor's title. Permitted values are: ⌘ m : Mr ⌘ f : Ms ⌘ c : Company ⌘ @ : Unknown (in this case an attempt will be made to determine a suitable title automatically based on the first name)
<code>string</code>	firstname	Debtor's first name
<code>string</code>	lastname	Debtor's last name
<code>string</code>	company	Debtor's company name
<code>string</code>	co	c/o address of the debtor (e.g. 'c/o Smith')
<code>string</code>	street	Number and street
<code>string</code>	postcode	Postcode
<code>string</code>	City	Location

<code>string</code>	country	ISO 3166 country code (e.g. DE for Germany, AT for Austria, CH for Switzerland)
<code>[integer]</code>	addressStatus	Optional. Status of debtor's postal address. Use only after consultation with mediafinanz.
<code>string</code>	telephone1	Main telephone number
<code>string</code>	telephone2	Additional telephone number
<code>[string]</code>	fax	Fax number
<code>string</code>	email	Email address
<code>[date]</code>	dateOfBirth	Optional. Debtor's date of birth in the format YYYY-MM-DD.
<code>[AdditionalAddress]</code>	deliveryAddress	Optional. Alternative delivery address, if provided.

11.24 **DirectPayment**

Direct payment by the debtor to the client.

Type	Name	Note
<code>date</code>	dateOfPayment	Date payment is received in the client's account in the format YYYY-MM-DD
Money	paidAmount	Exact amount received. This information is required to determine whether the debtor has paid the debt collection fees to the client.

11.25 **Money**

This data type represents money amounts. It is based on the elementary `decimal` data type, limited to two decimal places.

11.26 **NewClaimMessage**

New message from mediafinanz relating to a debt collection claim.

Type	Name	Note
<code>DateTime</code>	time	Time message created
<code>string</code>	fileNumber	mediafinanz file number
<code>string</code>	invoiceNumber	Invoice number
<code>string</code>	text	Message text

11.27 *NewClaimMessages*

New messages from mediafinanz relating to debt collection claims.

Type	Name	Note
<code>string</code>	transactionId	Transaction ID. Is required for the <code>commitTransaction()</code> function.
<code>ArrayOfNewClaimMessage</code>	messages	Array with messages from mediafinanz relating to debt collection claims

11.28 *NewClaimResult*

Value returned following submission of a debt collection claim. Besides the optional `info` element, either the `errorList` or the `fileNumber` element will be returned.

Type	Name	Note
<code>[string]</code>	info	Optional. May include a note for the client.
<code>ArrayOfError</code>	errorList	If the submitted claim could not be saved, this element contains an array with error messages (<code>ArrayOfError</code> type). This may occur, such as when mandatory fields have not been completed or improper entries have been identified.
<code>string</code>	fileNumber	If the claim was saved successfully, this element contains the mediafinanz file number under which the proceedings will now be conducted. This file number can be used to identify a claim (see 11.18 <code>ClaimIdentifier</code>).

11.29 *PayoutClaim*

Details about a debt collection claim which has been included in a payout.

Type	Name	Note
<code>string</code>	fileNumber	mediafinanz file number
<code>string</code>	invoiceNumber	Invoice number
<code>string</code>	debtorFirstname	Debtor's first name
<code>string</code>	debtorLastname	Debtor's last name
<code>string</code>	debtorCompany	Debtor's company name
Money	currentDebtorPayment	Current payment of the debtor
Money	totalDebts	Total amount demanded
Money	outstanding	Amount still outstanding
Money	previouslySettled	Amount previously paid out
Money	currentlySettled	Amount currently paid out
Money	currentlySettledOverdueFees	Of which reminder fees
Money	currentlySettledOther	Of which others

	s	
Money	currentlySettledBase Claim	Of which original claim
Money	currentlySettledEncashment ↵ Costs	Debt collection costs currently paid out

11.30 **PayoutSummary**

Summary of a debt collection payout.

Type	Name	Note
<code>string</code>	payoutNumber	Payout number
<code>date</code>	payoutDate	Payout date
Money	totalPayout	Payout amount

12 Error codes

The following list contains a description of the error messages which may arise when calling the documented functions. Errors are returned as SOAP faults, and can be treated as exceptions in the majority of programming languages. As further error codes may be introduced in the future, you should programme your application in such a way that errors which are not listed here can also be handled.

12.1 General errors

Code	Message
-100	server error
	An error has occurred in the SOAP service for which no further details are given.

Code	Message
-100	incomplete or invalid parameters
	Not all required parameters have been specified for the function call or specified parameters contain invalid values.

Code	Message
-101	call to unknown function
	The called function is not part of the mediafinanz SOAP service.

Code	Message
-102	You are using a deprecated version of this service! Please update your application!
	You are using an outdated version of the SOAP interface, which has since been deactivated by mediafinanz.

Code	Message
-702	internal error
	An error occurred with an external service provider.

Code	Message
-999	<i>Variable maintenance message</i>
	The function is unavailable owing to maintenance work by mediafinanz or a third-party service provider. The maintenance message includes further details about the maintenance work.

Code	Message
0	unknown operation
	The called function is not part of the mediafinanz SOAP service (only applies to WSDL document/literal).

Code	Message
0	invalid parameter
	The specified parameters are invalid (only applies to WSDL document/literal).

12.2 Authentication errors

Code	Message
-201	Authentication failed! Error -201
	The specified licenceKey parameter does not contain 32 characters or the client could not be identified.

Code	Message
-202	Authentication failed! Error-Code -202
	The licenceKey parameter was generated without a valid partner licence key (only applies to partner authentication).

Code	Message
-203	Authentication failed! Error-Code -203
	The partner license key used to generate the licenceKey parameter is invalid (only applies to partner authentication).

Code	Message
-204	Authentication failed! Error-Code -204
	Error when authenticating the partner (only applies to partner authentication).

Code	Message
-205	Authentication failed! Error-Code -205
	The application key used by you is no longer valid. Please contact mediafinanz.

Code	Message
-206	Authentication failed! Error-Code -206
	12.3 The client account has not yet been set up for use via the API. (See 4.3, Linking)

Code	Message
-207	Authentication failed! Error-Code -207
	The client license key used to generate the licenceKey parameter is invalid.

Code	Message
-208	Authentication failed! Error-Code -208
	The application key used by you is no longer valid. Please contact mediafinanz.

Code	Message
-209	Authentication failed! Error-Code -209
	The client account has been deactivated.

Code	Message
-210	forbidden function call
	The called function has been blocked for your application.

Code	Message
-211	function call not yet permitted in non-sandbox-mode. Please contact mediafinanz technical support!
	The called function has not yet been approved by mediafinanz. This function can only be used in production mode once it has been approved by mediafinanz. (See 8, Fehler! Verweisquelle konnte nicht gefunden werden.)

Code	Message
-212	administration through partner not yet permitted by client!
	The client has not (yet) agreed to administration by the partner. Approval can be granted in the client online system. https://mandos.mediafinanz.de/api [link in German] (only applies to partner authentication).

Code	Message
-213	test-account is restricted to sandbox-mode
	The client specified in the clientId parameter is a test account. This client cannot execute any functions in production mode.

12.4 Function errors

Code	Message
-300	<i>Variable error message</i>
	General errors occurred during the submission of a new debt collection order. Further details are displayed in the error text (e.g. 'This account has been temporarily suspended for the submission of new claims.').

Code	Message
-400	claim could not be identified
	No claim could be identified using the claimIdentifier specified in the function call.

Code	Message
-501	invalid payout number
	The specified payout number is invalid.

Code	Message
-502	invalid payout number
	The specified payout number is invalid.

Code	Message
-601	option not available for current claim-status
	The option is not available for the current claim status.
Code	Message

-700	<i>Variable error message</i>
	The called function has been blocked for the client. Refer to the error message for further details.

Code	Message
-801	<i>wrong address index, must be one of (null, 0, 1, 2)</i>
	An incorrect index was used when accessing an address. If no index or 0 is specified, the current main address is used. For 1 and 2, the corresponding addresses listed in Mandos will be used.

Code	Message
-802	<i>new address and first address are the same</i>
	The newly submitted address is the same as the one stored as the main address.

Code	Message
-803	<i>new address and second address are the same</i>
	The newly submitted address is the same as the one stored as the second address.

Code	Message
-804	<i>wrong address index (1), must be one of (null, 0) (only one address is given)</i>
	An incorrect index was used when accessing an address. If no index or 0 is specified, the current main address is used. In this case, only one address exists.

Code	Message
-805	<i>wrong address index (2), must be one of (null, 0, 1) (only two addresses are given)</i>
	An incorrect index was used when accessing an address. If no index or 0 is specified, the current main address is used; if 1 is specified, the only other address present will be used.

13 Status list

The following list provides a description of each state a claim can go through at mediafinanz. A status comprises a code, a description and, potentially, further details. The returned status codes are internal values. This means the same description is used for several codes. As further statuses may be introduced in the future, you should programme your application in such a way that statuses which are not listed here can also be handled.

Code	Description
10201	Newly submitted
10202	In progress
10203	Closed unsuccessfully
10204	In progress
10205	In progress
10206	Payment agreed
10207	Payment agreement not complied with
10208	Closed after part payment
10209	Full payment made
10210	Cancelled
10211	With lawyer
10212	Cancelled by mediafinanz
10213	In progress
10214	Newly submitted
10215	In progress
10216	In progress
10217	Direct payment made to you
10218	In progress
10219	Payment agreed
10220	In progress
10221	In progress
10222	Judicial reminder proceedings closed unsuccessfully
10223	Awaiting your decision

10224	In progress
10225	Long-term observation
10226	Offered for sale
10227	In progress
10228	In progress
10229	In progress
10230	In progress
10231	In progress
10234	Long-term observation
10235	Debt payment collection closed unsuccessfully
10238	In progress
10239	Long-term observation unsuccessful
10240	Unsuccessful sale of receivables
10241	Settlement reached
10242	In progress
10243	In progress
10244	Judicial reminder proceedings
10245	Incomplete direct payment made to you
10246	Newly submitted

14 Contact details for questions

If you have any questions about the interface, please contact:

Tesch mediafinanz GmbH

IT department

Tel: +49 (0)541 2029-110

Fax: +49 (0)541 2029-101

Email: technik@mediafinanz.de