



CRM JSON API

Table of Contents

Document Version	3
	3
Introduction	4
	4
	4
Http Post Parameters	5
Client:	6
Add	6
Client Get	10
	10
(Client) List	11
Portal User:	12
Get	12
Contact Person:	13
Add	13
Parameter Listing:	15
Payment Method:	17
Validate	17
Parameter Listing:	18
Add<List of Bank Accounts>	19
Add	21
Parameter Listing:	24
Mark for Debit Order	26
DebiCheck	27
Send DebiCheck Authentication Request	27
Notes for Netcash users	28
Add contract to Netcash masterfile	28
Send DebiCheck Cancellation Request	29
Get Contract (DebiCheck Response)	30
Contract:	32
Add	32
	32
Response Parameter Listing:	33
Parameter Listing:	34
Adding a contract for sending DebiCheck mandates	35
Sample DebiCheck request JSON for Capitec	35
Sample DebiCheck request JSON for Netcash	36
Mandate Parameter Listing for DebiCheck:	37

Contract Update	38
Parameter Listing:	39
Sample Java code:	40

Document Version

Version	Date	Change
1	06/05/2015	Initial draft
2	13/11/2017	Validate Payment method
5	10/06/2021	Update contract
6	21/04/2022	Add Payment Method changes
7	15/05/2022	Add Payment Method list
8	20/05/2022	Changes to addPayments
9	28/05/2022	Changes to addPayments
2.10	17/10/2024	Client List document search fields
2.11	23/06/2025	Add contract - fields for DebCheck, new getContract method
2.12	16/07/2026	Send DebiCheck mandate for Netcash
2.13	17/07/2026	Update Payment Provider Master file (Netcash)

Introduction

- RESTful JSON over https.
- UTF-8 encoding.
- Date format: ISO8601 UTC, yyyy-MM-dd'T'HH:mm:ssz e.g. 2015-05-23T00:00:00UTC.
- Amounts must be in the format 0.00 e.g. 1000.00
- API user is created under the General Settings tab -> Staff Management -> Create User. Select API User checkbox.
 - Api Key will automatically be generated.
 - Must have the Client role selected.
 - Enter a password and save.

Http Post Parameters

Http Request	Description
apikey	Created in staff management in the General Settings tab. This must be set in the http header request for each call.
password	Created in staff management in the General Settings tab. This must be set in the http header request for each call.
Http content	Mime Type: application/json The JSON message in UTF-8 text format.

Client:

Add

Post URL:

<https://secure.billdozer.com/api/rest/client/add>

Sample JSON request:

```
{ "legalEntity":false,
  "accountNumber":"ROB001",
  "title":"Mr",
  "initials":"RJ",
  "firstName":"Robert",
  "lastName":"Pretorius",
  "name":"Robert Pretorius",
  "dob":"1975-03-13T00:00:00UTC",
  "occupation":"",
  "gender":"M",
  "identityNumber":"7503135046080",
  "emailAddress":"robert@billdozer.com",
  "companyName":null,
  "tradingAsName":null,
  "vatNumber":null,
  "flag":"",
  "flag2":"",
  "registrationNumber":null,
  "notifyEmail":true,
  "clientTelephones":[
    { "type":"MOBILE",
      "number":"0741146632",
      "description":"" }
  ],
  "clientAddresses":[
    { "type":"STREET",
      "line1":"11 Vondel Road",
      "line2":"",
      "line3":"Edgemead",
      "line4":"",
      "city":"Cape Town",
      "postCode":"7441",
      "country":"ZA" }
  ],
  "additionalProperties":{
    "updateIfExistingClient":true
  },
  "testMode":false
}
```

Client Fields (for add):

*Flag: O = Optional, R = Required

Name	Flag	Availability	Description
id		get	Long. Unique id if this is for an existing client
active		update get	Boolean: Is the client active or not.
legalEntity	R	add/ update get	Boolean. Is this client a legal entity or a natural person.
accountNumber	O	add/ update get	String. Unique account number. If no account number is supplied then an automatic account number concatenating the first 3 digits of the clients name and a sequential counter. E.g. clients name Robert Pretorius. Very first account number will be ROB1.
title	O	add/ update get	String. Title
initials	O	add/ update get	String. Initials
firstName	O/R	add/ update get	String. Required if a natural person.
lastName	O/R	add/ update get	String. Required if a natural person.
name	O/R	get	String. If a natural person then it is the concatenated first and last name. If a legal entity then the companyName.
dob	O	add/ update get	Date. Date of birth. E.g. 1975-03-13T00:00:00UTC
occupation	O	add/ update get	String
gender	O	add/ update get	String. M – Male, F – Female.
identityNumber	O	add/ update get	String. ID number. Will automatically be validated if provided.
emailAddress	O/R	add/ update get	Required if notifyEmail is true and is a natural person.
companyName	R	add/ update get	String. Required if legal entity.
tradingAsName	O	add/ update get	String. Trading as name.
vatNumber	O	add/ update get	String. Vat registration number.
flag	O	add/ update get	String. Category grouping.
flag2	O	add/ update get	String. Category2 grouping.
registrationNumber	O	add/ update get	String. Company registration number.
notifyEmail	O	add/ update get	Boolean. If a natural person and want this client to automatically receive invoices, receipts, debit notes and statement notifications via email.
testMode	O	add/ update get	Boolean. Default false. If set to true then additions and updates are not committed to the database. The transaction id and reference will also not automatically created. "resultJson" property will also be returned in the Response which contains the full json of the transaction object for debugging.
clientTelephones	O	add/ update get	An array of telephones
↳ id		get	The id of the telephone line

Name	Flag	Availability		Description
↳ type	R	add/ update	get	String. The type of telephone object. Either "LANDLINE", "MOBILE", "FACSIMILE", "ALT"
↳ number	R	add/ update	get	String. Telephone number.
↳ description	O	add/ update	get	String. Description.
clientAddresses	O	add/ update	get	An array of addresses
↳ id	R	add/ update	get	Long. The unique id of the contract to perform a lookup.
↳ type	R	add/ update	get	String. Either: "STREET", "POSTAL", "BILLING", "WORK"
↳ line1	R	add/ update	get	String. Line 1, Required.
↳ line2	O	add/ update	get	String. Line 2
↳ line3	O	add/ update	get	String. Suburb.
↳ line4	O	add/ update	get	String Line 4.
↳ city	O	add/ update	get	String. City.
↳ postCode	O	add/ update	get	String. Zip, Postal Code.
↳ country	O	add/ update	get	String ISO 3166 2 digit country code. http://27.org/isocountrylist/ Example. ZA for South Africa.
↳ gps	O	add/ update	get	String. Gps coordinates.
additionalProperties	O	add/ update	get	Name value pair list. Used for future requirements,
↳ updateIfExistingClient	O	add		Updates the client based on the provided email address or account number. If more than one client is found on email address the latest client will be updated

Sample JSON response:

Failed:

```
{
  "id": null,
  "status": "NOK",
  "messages": ["Client account number already exists."],
  "additionalProperties": { "isTestMode": true }
}
```

Success:

```
{
  "id": "6301",
  "status": "OK",
  "messages": [ ],
  "additionalProperties": {
    "accountNumber": "ROB001",
    "isTestMode": false
  }
}
```

```
{
  "id": "0",
  "status": "OK",
  "messages": [ ],
  "additionalProperties": {
    "accountNumber": "ROB001",
    "isTestMode": true,
    "resultJson": "{ full client object returned in json format }"
  }
}
```

Response Parameter Listing:

Name	Description
id	The id of the object updated or created in the system. If status is NOK then this value will be set to 0.
status	OK or NOK
messages	If status is NOT then a String list will be returned of errors.
additionalProperties	Name value pair list.
→ accountNumber	String. The account number of the transaction created or updated. If status is NOK then this value will be set to null.
→ testMode	Boolean. The test mode in the post request.
→ resultJson	String. Encoded json of the transaction created. Used for debugging. This is only returned if the status is OK and testMode is true.

Client Get

Get URL:

<https://secure.billdozer.com/api/rest/client/get/{id}>

(Client) List

Get URL:

<https://secure.billdozer.com/api/rest/client/list>

List Clients Filters

Following parameters must be put into the Http Header

Http Header Parameter	Description
page	Integer. Required. The page of results to return.
perPage	Integer. Required. The number of results per page to return. Max 100
fromDateTime	DateTime. Optional. Return all clients created or modified from this date and time UTC.
goodStandingMarker	String. Optional – Values: IGS (In Good Standing, this includes any clients who have no overdue invoices, including new clients who have only recently been invoiced) IGSWPI (In Good Standing with paid invoices. This is a subset of the in good standing list, i.e. those with at least 1 successful payment) IGSWNOPI (In Good Standing with NO paid invoices. This is a subset of the in good standing list, i.e. those who are new)
numberOfDaysAfterDueDate	Numeric. Optional. Default is set to 5. This is linked to the goodStandingMarker parameter. It is meaningless if the goodStandingMarker is not set. Its primary purpose is to reflect payment terms. E.g. you might have an invoice that is due on the 1 st of the month. But you wish to give some leeway before excluding them from good standing as the payment terms might be 30 days.
searchFieldName	Valid Search Field values are: name, email, first name, last name
searchFieldValue	Value to be searched for in search field

Response:

```
{
  "status":"OK", // OK or NOK
  "message":null, // Error message
  "type":"Client",
  "page":1,
  "perPage":15,
  "totalResults":4390,
  "results":[ // Client list
  ]
}
```

Portal User:

Get

A Portal User is a user account that is given to a consumer / customer in order to access the client zone. The client zone enables consumers to view and download their invoices.

This function returns the registration key for a Portal User so they can be signed into the client zone. The user is identified by their email address

Get URL:

<https://secure.billdozer.com/api/rest/portaluser/registrationkey/{email Address}>

Sample JSON response:

Failed:

```
{
  "id":null,
  "status":"NOK",
  "messages":["Cannot find any Active Portal users with email address zzz@billdozer.com"],
  "additionalProperties":{}
}
```

Success:

```
{
  "id":"f6f8d740-4fe8-461a-b740-bfd1d321da35",
  "status":"OK",
  "messages":["Registration key for tohy@billdozer.com"],
  "additionalProperties":{}
}
```

Redirect user to Portal

Once you have the registration key you can re-direct the user the Billdozer clientzone:

<https://secure.billdozer.com/portal/pages/secure/view?registrationKey=0c5ae7e9-d29d-4074-9d0f-45388d1151a6>

Contact Person:

Add

Post URL:

<https://secure.billdozer.com/api/rest/contact/add>

Adds a contact person to a client record.

Sample JSON request:

```
{
  "client":{
    "id":6301
  },
  "title":"Mr",
  "jobTitle":"IT Architect",
  "line2":"",
  "town":"Edgemoor",
  "city":"Cape Town",
  "state":"Western Cape",
  "postCode":"7441",
  "country":"ZA",
  "notifyEmail":true,
  "testMode":false,
  "additionalProperties":{
  }
}
```

Sample JSON response:

Failed:

```
{
  "id":null,
  "status":"NOK",
  "messages":[
    "Please enter the contacts first name."
  ],
  "additionalProperties":{
    "isTestMode":true
  }
}
```

Success:

```
{
  "id":"59063",
  "status":"OK",
  "messages":[
  ],
  "additionalProperties":{
    "isTestMode":false
  }
}

{
  "id":"0",
  "status":"OK",
  "messages":[
  ],
}
```

```

"additionalProperties":{
  "isTestMode":true,
  "resultJson":"{ full client contact object returned in json format }"
}
}

```

Response Parameter Listing:

Name	Description
id	The id of the object updated or created in the system. If status is NOK then this value will be set to 0.
status	OK or NOK
messages	If status is NOT then a String list will be returned of errors.
additionalProperties	Name value pair list.
→ testMode	Boolean. The test mode in the post request.
→ resultJson	String. Encoded json of the transaction created. Used for debugging. This is only returned if the status is OK and testMode is true.

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Availability		Description
id			get	Long. Unique id of this contact.
client	R	add	get	Client object that this contact belongs to.
→ id	R	add/ update	get	Long. Client id number.
title	O	add/ update	get	String. Title.
jobTitle				String. Job Title.
initials	O	add/ update	get	String. Initials.
firstName	R	add/ update	get	String. First Name.
lastName	R	add/ update	get	String. Last Name.
name			get	String. Concatenated first and last name.
dob	O	add/ update	get	Date. Date of birth. E.g. 1975-03-13T00:00:00UTC
occupation	O	add/ update	get	String. Contacts occupation.
gender	O	add/ update	get	String. M – Male, F – Female.
idNumber	O	add/ update	get	String. Contacts ID Number.
relationship	O	add/ update	get	String. Relationship to client.
companyName	R	add/ update	get	String. Company contact works for.
emailAddress	O/R	add/ update	get	Required if notifyEmail is true and is a natural person.
landline	O	add/ update	get	String. Telephone landline.
fax	O	add/ update	get	String. Fax number.
mobile	O	add/ update	get	String. Mobile number.
line1	O	add/ update	get	String. Address line 1.
line2	O	add/ update	get	String. Address line 2.
town	O	add/ update	get	String. Address Town/Suburb.
city	O	add/ update	get	String. Address City.
state	O	add/ update	get	String. Address State/Province.
postCode	O	add/ update	get	String. Address post/zip code.
country	O	add/ update	get	String ISO 3166 2 digit country code. http://27.org/isocountrylist/ Example. ZA for South Africa.
notifyEmail	O	add/ update	get	Boolean. If a natural person and want this client to automatically receive invoices, receipts, debit notes and statement notifications via email.
testMode	O	add/ update	get	Boolean. Default false. If set to true then additions and updates are not committed to the database. The transaction id and reference will also not

Name	Flag	Availability		Description
				automatically created. "resultJson" property will also be returned in the Response which contains the full json of the transaction object for debugging.
additionalProperties	O	add/ update	get	Name value pair list. Used for future requirements,

Payment Method:

Validate

Post URL:

<https://secure.billdozer.com/api/rest/paymentMethod/validate>

Sample JSON request for **CREDIT CARD**:

```
{
  "type": "CREDIT CARD",
  "bankName": "",
  "bankSortCode": "",
  "bankAccountNumber": "",
  "bankAccountName": "",
  "accountType": "",
  "cardNumber": "4242424242424242",
  "cardName": "RJ Pretorius",
  "cardExpiryMonth": 7,
  "cardExpiryYear": 2018,
  "cardExpired": false,
  "cardType": "VISA",
}
```

Sample JSON response:

Failed:

```
{ "id": null,
  "status": "NOK",
  "messages": [ "Please enter the Card Number." ],
  "additionalProperties": { "isTestMode": true }
}
```

Success:

```
{ "id": null,
  "status": "OK",
  "messages": [ ],
  "additionalProperties": { "isTestMode": false }
}
```

Sample JSON request for **BANK ACCOUNT**:

```
{
  "type": "BANK ACCOUNT",
  "bankName": "FIRST NATIONAL BANK",
  "bankSortCode": "250655",
  "bankAccountNumber": "62401258274",
  "bankAccountName": "Spartan IT Services",
  "accountType": "1",
  "cardNumber": "",
  "cardName": "",
  "cardExpiryMonth": "",
  "cardExpiryYear": "",
  "cardExpired": false,
  "cardType": ""
}
```

Sample JSON response:

Failed:

```
{ "id":null,
  "status":"NOK",
  "messages":["Account number failed check digit validation"],
  "additionalProperties":{"isTestMode":false}
}
```

Success:

```
{ "id": null,
  "status":"OK",
  "messages":[ ],
  "additionalProperties":{"isTestMode":false }
}
```

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Description
type	R	String. Either "BANK ACCOUNT" or "CREDIT CARD"
bankName	O/R	String. Required if type = "BANK ACCOUNT". Bank name code combination: 'ABSA BANK' 'FIRST NATIONAL BANK SOUTH AFRICA' 'NEDBANK SOUTH AFRICA' 'STANDARD BANK SOUTH AFRICA' 'CAPITEC BANK' 'AFRICAN BANK' 'INVESTEC BANK' 'MTN BANKING' 'POSTBANK' 'MEEG BANK'
bankSortCode	O/R	String. Required if type = "BANK ACCOUNT". See above.
branchName	O	String. Required if type = "BANK ACCOUNT". Branch Name.
bankAccountNumber	O/R	String. Required if type = "BANK ACCOUNT". Bank account number.
bankAccountName	O/R	String. Required if type = "BANK ACCOUNT". Bank account name. The field cannot contain special characters only A-Za-z.
accountType	O/R	String. Required if type = "BANK ACCOUNT". Values: 1 = Current / Checking 2 = Savings 3 = Transmission
cardNumber	O/R	String. Required if type = "CREDIT CARD". Credit card number.
cardName	O/R	String. Required if type = "CREDIT CARD". Name on the card. Cannot contain any special characters.
cardExpiryMonth	O/R	Integer. Required if type = "CREDIT CARD". Card expiry month.
cardExpiryYear	O/R	Integer. Required if type = "CREDIT CARD". Card expiry year.
cardType	R	MASTERCARD, VISA, AMEX, DINERS
reference	O/R	If the card is already tokenized i.e. via an online payment call the token value is provided here.
cardMaskDigits	O/T	Use this in conjunction with the reference when submitting an already tokenized card i.e. 5223000000005071 << masked with zeros.

Add<List of Bank Accounts>

Post URL:

<https://secure.billdozer.com/api/rest/paymentMethods/add>

Validates and stores a list of up to **100** bank account payment methods on the queue for insert.

Basic validation ensures the account numbers are numeric and that all required fields are populated. Any account that fails **basic validation** will not be saved and a message will be returned:

Invalid type should be BANK ACCOUNT
Invalid account name
Invalid account number
Invalid branch code

The client as defined by reference or id must exist in Billdozer and should the same bank account already exist for the client (as defined by client + branch code + bank account number) it will not be added again and the existing payment method will be assigned to all the contracts of that client.

A message will be returned for each bank account submitted, indicating whether or not it will be added (OK).

After the bank accounts are saved, the **bank account validation** will be run on any that do not have a dateValidated field value. If you have already used the Validate web service and it was successful then set the dateValidated. The bank account validation is free to use and ensures that the details provided match the specific requirements of various banks. It is not the same as **Bank Verification** which has a cost and verifies that the account exists and can accept debit orders.

Sample JSON request

```
{
  "testMode" : false,
  "paymentMethods" : [
    {
      "client":{ "accountNumber":"GRANTN001A" },
      "type":"BANK ACCOUNT",
      "bankName":"NEDBANK BANK SOUTH AFRICA",
      "bankSortCode":"196905",
      "bankAccountNumber":"1969111615",
      "bankAccountName":"Test NedBank Name",
      "dateValidated":"2022-05-15",
      "accountType":"1"
    },
    {
      "client":{ "accountNumber":"Lance007" },
      "type":"BANK ACCOUNT",
      "bankName":"FIRST NATIONAL BANK SOUTH AFRICA",
      "bankSortCode":"250655",
      "bankAccountNumber":"62881550988",
      "bankAccountName":"Test Name",
      "dateValidated":"2022-05-15",
      "accountType":"1"
    },
    {
      "client":{ "accountNumber":"DT000001AAA" },
      "type":"BANK ACCOUNT",
      "bankName":"NEDBANK BANK SOUTH AFRICA",
      "bankSortCode":"196905",
      "bankAccountNumber":"1969111615",
      "bankAccountName":"Test NedBank Name",
      "accountType":"1"
    },
    {
      "client":{ "accountNumber":"MMA0001A" },
      "type":"BANK ACCOUNT",
      "bankName":"NEDBANK BANK SOUTH AFRICA",
      "bankSortCode":"196905",
      "bankAccountNumber":"1969111615",
      "bankAccountName":"Test Name",
      "accountType":"1"
    }
  ]
}
```

```
}  
]  
}
```

Sample JSON response:

Success with an invalid bank account

```
{"id":null,"status":"OK","messages":["LOUTEST00005A | Invalid account number","AA001 | OK","GRANTN001A | OK","Lance007 | OK"],"additionalProperties":{"isTestMode":false}}
```

Success with a client not found

```
{"id":null,"status":"OK","messages":["PALTROW001 | Client not found","GRANTN001A | OK","Lance007 | OK","DT000001AAA | OK","MMA0001A | OK"],"additionalProperties":{"isTestMode":false}}
```

Success with existing bank accounts:

```
{"id":null,"status":"OK","messages":["GRANTN001A | bank account exists","Lance007 | bank account exists","DT000001AAA | bank account exists","MMA0001A | bank account exists"],"additionalProperties":{"isTestMode":false}}
```

Success all accepted

```
{"id":null,"status":"OK","messages":["GRANTN001A | OK","Lance007 | OK","DT000001AAA | OK","MMA0001A | OK"],"additionalProperties":{"isTestMode":false}}
```

Add

Post URL:

<https://secure.billdozer.com/api/rest/paymentMethod/add>

For use cases where the client has not been added via the API, either the id or the accountNumber (unique reference) can be used to lookup the client. If no id is provided or if the id is 0 then the client will be located using the accountNumber value. If the input JSON client.collectOnContract marker = true then the bank details will be assigned to the contract and the collectOnContract will be updated in Billdozer i.e the client will be marked for debit order (if not already as such). Should the client.collectOnContract = false then the payment method will be added, however it will not be assigned to the contract. The **Mark for Debit Order** web service can be used to perform this later. Note also that should the payment method (bank account number + branch code) already exists for the client then no update will be performed and the response will be NOK.

Sample JSON request:

```
{
  "client":{
    "id":753856,
    "accountNumber":"GRANTN001A",
    "collectOnContract": true
  },
  "type":"BANK ACCOUNT",
  "bankName":"FIRST NATIONAL BANK SOUTH AFRICA",
  "bankSortCode":"250655",
  "branchName": "",
  "bankAccountNumber":"62401258274",
  "bankAccountName":"Test Bank Name",
  "accountType":"1",
  "verify":false
}
```

This next example is for a card that already been tokenized via an online payment call – the card number is masked with '0' only showing 1st and last 4 digits i.e.:

```
{ "client": {
  "id":430865
},
  "type":"CREDIT CARD",
  "reference":"b2bb0947-b1b5-45c2-bc2b-e0a228b825e8",
  "cardMaskDigits":"5223000000005071",
  "cardName":"T Hewlett",
  "cardExpiryMonth":4,
  "cardExpiryYear":2020,
  "cardExpired":false,
  "cardType":"MASTERCARD"
}
```

Sample JSON response:

Failed:

```
{
  "id":null,
  "status":"NOK",
  "messages":[
    "Please enter the Card Number."
  ],
  "additionalProperties":{"isTestMode":true }
}
```

Where the client already has the bank account on their profile:

```
{
  "id" : null,"
  "status" : "NOK",
  "messages":["Payment Method already exists"],
  "additionalProperties" : {"isTestMode" : false}
}
```

Success:

```
{
  "id":" 59064",
  "status":"OK",
  "messages":[

  ],
  "additionalProperties":{"
    "isTestMode":false
  }
}

{
  "id":"0",
  "status":"OK",
  "messages":[

  ],
  "additionalProperties":{"
    "isTestMode":true,
    "resultJson":{" full client object returned in json format }"
  }
}
```

Response Parameter Listing:

Name	Description
id	The id of the object updated or created in the system. If status is NOK then this value will be set to 0.
status	OK or NOK
messages	If status is NOT then a String list will be returned of errors.
additionalProperties	Name value pair list.
→ testMode	Boolean. The test mode in the post request.
→ resultJson	String. Encoded json of the transaction created. Used for debugging. This is only returned if the status is OK and testMode is true.

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Availability	Description
id		get	Long. Unique id of this payment method.
client	R	add get	Client object that this contact belongs to.
→ id	R	add/ update get	Long. Client id number.
type	R	add/ update get	String. Either "BANK ACCOUNT" or "CREDIT CARD"
bankName	O/R	add/ update get	String. Required if type = "BANK ACCOUNT". Bank name code combination: 'ABSA BANK' code '632005' 'FIRST NATIONAL BANK SOUTH AFRICA' code '250655' 'NEDBANK SOUTH AFRICA' branch/sort code '198765' 'STANDARD BANK SOUTH AFRICA' code '051001' 'CAPITEC BANK' code '470010' 'AFRICAN BANK' code '430000' 'INVESTEC BANK' code '580105' 'MTN BANKING' code '490991' 'POSTBANK' code '460005' 'MEEG BANK' code '470010'
bankSortCode	O/R	add/ update get	String. Required if type = "BANK ACCOUNT". See above.
branchName	O	add/ update get	String. Required if type = "BANK ACCOUNT". Branch Name.
bankAccountNumber	O/R	add/ update get	String. Required if type = "BANK ACCOUNT". Bank account number.
bankAccountName	O/R	add/ update get	String. Required if type = "BANK ACCOUNT". Bank account name. The name of the account holder for the homing bank account. This will in most cases be the same as the "Account Name" field. This will be different in some situations e.g.: Your client wants? you to debit/credit his business account instead of his personal account, thus, the Bank Account Name will be his business name. The field cannot contain special characters only A-Za-z.
accountType	O/R	add/ update get	String. Required if type = "BANK ACCOUNT". Values: 1 = Current / Checking 2 = Savings 3 = Transmission
cardNumber	O/R	add/ update get	String. Required if type = "CREDIT CARD". Credit card number.
cardLast4Digits	O/R	add/ update get	This is the last 4 digits of the card number.
reference	O/R	add/ update get	This is the token received from the Payment Gateway.
cardName	O/R	add/ update get	String. Required if type = "CREDIT CARD". Name on the card. Cannot contain any special characters.
cardCvv	O	add/ update get	Integer. Optional. Card cvv number.
cardExpiryMonth	O/R	add/ update get	Integer. Required if type = "CREDIT CARD". Card expiry month.
cardExpiryYear	O/R	add/ update get	Integer. Required if type = "CREDIT CARD". Card expiry year.

Name	Flag	Availability		Description
cardExpired			get	Boolean. Is the card expired.
cardMaskDigits			get	String. Required if type = "CREDIT CARD". Masked card digits after token is created.
cardType	R	add		MASTERCARD, VISA, AMEX, DINERS
verify	O	add/ update	get	Boolean. True if the bank account must be verified.
dateValidated			get	Date. The date the bank account was validated.
dateVerified			get	Date. The date the bank account was verified.
testMode	O	add/ update	get	Boolean. Default false. If set to true then additions and updates are not committed to the database. The transaction id and reference will also not automatically created. "resultJson" property will also be returned in the Response which contains the full json of the transaction object for debugging.

Mark for Debit Order

This function will set the collect on contract marker to true on the repeating invoices for the customer.

URL: <https://secure.billdozer.com/api/rest/client/markDebitOrder>

Method: POST

Sample request:

```
{
  "client": {
    "id" : 316451,
    "accountNumber" : null
  },
  "markForDebitOrderFlag": true,
  "debitOrderDay": "12",
  "testMode": false,
  "ipAddress" : "127.0.0.1",
  "userAgent" : "Mozilla",
  "agreedBy" : "Toby Hewlett"
}
```

Sample success response:

```
{
  "id": 1,
  "status": "OK",
  "messages": [],
  "additionalProperties": {
    "isTestMode": false
  }
}
```

Sample failure response:

```
{
  "id": 2,
  "status": "NOK",
  "messages": [
    "No client found for id: 2"
  ],
  "additionalProperties": {
    "isTestMode": false
  }
}
```

DebiCheck

Send DebiCheck Authentication Request

Get URL:

<https://secure.billdozer.com/api/rest/mandate/debichек/request/{debiCheckMandateDeliveryType}/{debiCheckMandateType}/{contractNumber}>

Params:

Field	Values
debiCheckMandateDeliveryType	For Capitec. "REAL_TIME" or "BATCH". For Netcash: "REAL_TIME", "REAL_TIME_DELAYED", or "BATCH"
debiCheckMandateType	"USAGE_BASED" or "VARIABLE" or "FIXED"
contractNumber	eg. "ROB001"

Sample JSON Response:

Success:

```
{  "id": null,
  "status": "OK",
  "messages": ["DebiCheck Mandate Request sent for contract ROB001"],
  "additionalProperties": {}
}
```

Failed:

```
{  "id": null,
  "status": "NOK",
  "messages": ["No contract found for contract number: ROB001" or
    "Incorrect DebiCheck delivery type"    ],
  "additionalProperties": {}
}
```

Notes for Netcash users

1. Masterfile Update. Before you can send a Netcash DebiCheck authentication request the Contract and bank account must be added to the masterfile. This can be done from the administration console (Contract Management - select action Update Netcash Masterfile) or using the mandate/masterfile/add web service. This process pushes the contract and bank details to Netcash and if successful the Mandate field (datePaymentProviderMasterFileUpdate) is set so authentication requests can be sent.
2. A DebiCheck template must be configured on Netcash. This specifies the type (usage, variable, fixed), the delivery mechanism (real time, delayed and batch and other parameters the authentication request must comply with. Each template has a unique code.
3. A DebiCheck template with the same code and parameters must be configured on Billdozer under General Settings - DebiCheck Templates.
4. The relevant template code value must be assigned to the debiCheckTemplateCode field on the Billdozer Mandate . Mandate requests must comply with the template rules otherwise they will be rejected.

Add contract to Netcash masterfile

Get URL:

<https://secure.billdozer.com/api/rest/mandate/masterfile/add/{contractNumber}>

This web service will add or update the contract and associated bank details to the Netcash master file. Note that it is the contract number and not the client account reference that gets added to the master file.

Send DebiCheck Cancellation Request

Post URL:

<https://secure.billdozer.com/api/rest/mandate/debichack/cancel>

Cancellation Reason	Corresponding Code
Cancellation/amendment requested by the creditor	MD17
Mandate cancelled due to early settlement	MCES
Contract Expired	CEXP
Mandate cancellation due to fraud	MCFR
Cancellation of a mandate initiation	MICN
Cancellation of a mandate amendment	MACN

The **trackingCancellationIndicator** is to cancel existing debit orders for this contract that may still be in tracking.

Sample JSON body:

```
{"contractNumber": "SBTEST012",  
"cancellationReason": "MD17",  
"trackingCancellationIndicator": false }
```

Success:

```
{ "id": "12999780",  
  "status": "OK",  
  "messages": ["DebiCheck Mandate Cancellation Request sent for contract SBTEST012"],  
  "additionalProperties": {} }
```

Failed:

```
{  
  "id": null,  
  "status": "NOK",  
  "messages": [  
    "No contract found for contract number: ROB001"  
  ],  
  "additionalProperties": {}  
}
```

Get Contract (DebiCheck Response)

Get URL:

<https://secure.billdozer.com/api/rest/contract/get/{contractNumber}>

Params:

contractNumber: eg. "ROB001"

Sample JSON Response:

Success (Mandate Accepted):

```
{
  "client": {
    "id": 6301
  },
  "contractNumber": "ROB001",
  "mandate": {
    "paymentMethod": 55062,
    "collectionAmount": 100.00,
    "dateFirstCollection": "2025-01-01T00:00:00Z",
    "dateAcceptanceDebiCheck": "2025-01-01T00:00:00Z",
    "responseCode01": ACCP,
    "responseCode02": Transaction Successfull,
    "responseCode03": null,
    "responseCode04": null
  },
  "description": null,
  "currency": "ZAR",
  "active": true,
  "testMode": false,
}
```

Failed (Mandate not accepted yet):

```
{
  "client": {
    "id": 6301
  },
  "contractNumber": "ROB001",
  "mandate": {
    "paymentMethod": 55062,
    "collectionAmount": 100.00,
    "dateFirstCollection": "2025-01-01T00:00:00Z",
    "dateAcceptanceDebiCheck": null,
    "responseCode01": null,
    "responseCode02": null,,
    "responseCode03": null,
    "responseCode04": null
  },
}
```

Failed (No contract found):

```
{
  "id": "0",
  "status": "NOK",
  "messages": [
    "No contract found for contract number: ROB001"
  ],
  "additionalProperties": {}
}
```

Contract:

Add

Post URL:

<https://secure.billdozer.com/api/rest/contract/add>

Sample JSON request:

```
{
  "client":{
    "id":6301
  },
  "contractNumber":"ROB001",
  "mandate":{
    "paymentMethod":{
      "id":55062
    },
    "debitDay1":"1",
    "daySaturday":6,
    "daySunday":6
  },
  "description":null,
  "currency":"ZAR",
  "active":true,
  "testMode":false,
  "additionalProperties":{
  }
}
```

Sample JSON response:

Failed:

```
{
  "id":null,
  "status":"NOK",
  "messages":["Please enter a Contract Number." ],
  "additionalProperties":{"isTestMode":true }
}
```

Success:

```
{
  "id":"59102",
  "status":"OK",
  "messages":[ ],
  "additionalProperties":{
    "contractNumber":"ROB001",
    "isTestMode":false }
  }
  {
    "id":"0",
    "status":"OK",
    "messages":[ ],
    "additionalProperties":{
      "contractNumber":"ROB001",
      "isTestMode":true,
      "resultJson":"{ full contract object returned in json format }"
    }
  }
}
```


Response Parameter Listing:

Name	Description
id	The id of the object updated or created in the system. If status is NOK then this value will be set to 0.
status	OK or NOK
messages	If status is NOT then a String list will be returned of errors.
additionalProperties	Name value pair list.
↳ contractNumber	String. The contract number of the transaction created or updated. If status is NOK then this value will be set to null.
↳ testMode	Boolean. The test mode in the post request.
↳ resultJson	String. Encoded json of the transaction created. Used for debugging. This is only returned if the status is OK and testMode is true.

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Availability		Description
id			get	Long. Unique id of this payment method.
client	R	add	get	Client object that this contact belongs to.
↳ id	R	add	get	Long. Client id that this contract must be created for.
contractNumber	O	add/ update	get	String. Unique contract number. If a contract number is not supplied then the client's accountNumber can be used.
mandate	O	add/ update	get	Mandate object
↳ paymentMethod	O/R	add	get	Payment Method object. Required if the mandate debitDay1 has a value.
↳↳ id	R	add	get	Long. The id of the payment method to be assigned to this contract.
↳ debitDay1	O	add/ update	get	String. The day of the month the debit order must be collected. Values: 1 – 30 or Last day
↳ daySaturday	O	add/ update	get	Integer. Values: 6 = Friday, 2 = Monday. If debit order collection day falls on a Saturday, then collect either on the Friday or Monday.
↳ daySunday	O	add/ update	get	Integer. Values: 6 = Friday, 2 = Monday. If debit order collection day falls on a Sunday or public holiday, then collect either on the Friday/Preceding day or Monday.
description	O	add/ update	get	String. Description.
currency	R	add/ update	get	String. Currency in ISO 4217 format charged in. e.g. ZAR. If not supplied will default to the default currency set in the General Settings tab.
active	R	add/ update	get	Boolean. Contract active. When a new contract is added the default is True.
testMode	O	add/ update	get	Boolean. Default false. If set to true then additions and updates are not committed to the database. The transaction id and reference will also not automatically created. "resultJson" property will also be returned in the Response which contains the full json of the transaction object for debugging.
additionalProperties	O	add/ update	get	Name value pair list. Used for future requirements,

Adding a contract for sending DebiCheck mandates

If you wish to send DebiCheck mandates for the contract then you need to ensure all the relevant fields are populated correctly, especially if there are no repeating invoices associated with the contract

Billdozer will use the Mandate DebiCheck fields first, and if they do not have values it will attempt to calculate them from the Repeating Invoices. There are also some differences if you are using Capitec or Netcash:

Required for	Field
Capitec	debiCheckDebtorAuthenticationCode
Netcash	debiCheckTemplateCode The template is configured in Netcash and a corresponding template must be created in Netcash.

Sample DebiCheck request JSON for Capitec

```
{ "client": { "id": 870973 },
  "contractNumber": "TOBYDEBICHECK001A",
  "mandate": { "paymentMethod": { "id": 11958699 },
    "debitDay1": "1",
    "daySaturday": 6,
    "daySunday": 6,
    "dateFirstCollection": "2025-06-30T00:00:00UTC",
    "debiCheckMandateType": "VARIABLE",
    "debiCheckDebtorAuthenticationCode": "0230",
    "amountRecurringIncludingTax": 750,
    "debiCheckMandateMaximumAmount": 2000, // must be specified if there are no repeating invoices
    "firstCollectionAmount": 245, // must be specified if there are no repeating invoices
    "adjustmentRate": 5 // must be specified if there are no repeating invoices
  },
  "description": null,
  "currency": "ZAR",
  "active": true,
  "testMode": false,
  "additionalProperties": {} }
```

Sample DebiCheck request JSON for Netcash

```
{ "client": { "id": 870973 },
  "contractNumber": "TOBYDEBICHECK001A",
  "mandate": { "paymentMethod": { "id": 11958699 },
    "debitDay1": "1",
    "daySaturday": 6,
    "daySunday": 6,
    "dateFirstCollection": "2025-06-30T00:00:00UTC",
    "debiCheckMandateType": "USAGE_BASED",
    "debiCheckTemplateCode": "NCDCT000003080", // must match configured templates
    "amountRecurringIncludingTax": 750,
    "firstCollectionAmount": 245 // must be specified if there are no repeating invoices
  },
  "description": null,
  "currency": "ZAR",
  "active": true,
  "testMode": false,
  "additionalProperties": {}
}
```

Mandate Parameter Listing for DebiCheck:

Name	Description
Mandate	Mandate object
↳dateFirstCollection	Date of first debit order in UTC format Required if there are no repeating invoices
↳debiCheckMandateType	USAGE BASED, VARIABLE, FIXED If not provided then USAGE BASED is the default.
↳amountRecurringIncludingTax	The recurring (monthly) amount, required if there are no repeating invoices
↳debiCheckMandateMaximumAmount	Required if there are no repeating invoices
↳firstCollectionAmount	Required if there are no repeating invoices
↳adjustmentRate	Required percent amount, required if there are no repeating invoices. If there are then it will be the assigned escalation percentage
↳debiCheckDebtorAuthenticationCode	"0230" : Use contact details at bank "0229" : Use contact details provided in request

Contract Update

This is used to change the debit order day on the Contract – Mandate. The contract can be specified either using the contract id or the contract number.

Post URL:

<https://secure.billdozer.com/api/rest/contract/update>

Sample JSON request:

```
{
  "client":{
    "id":659152
  },
  "id": 917283,
  "contractNumber":"LANCE001",
  "mandate":{
    "debitDay1":"1",
    "daySaturday":6,
    "daySunday":6
  },
  "active":true,
  "testMode":false,
  "additionalProperties":{
  }
}
```

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Availability		Description
id			get	Long. Unique id of this payment method.
client	R	update	get	Client object that this contact belongs to.
→ id	R	update	get	Long. Client id that this contract must be created for.
id	O	update	get	Long. Unique contract id if available
contractNumber	O	update	get	String. Unique contract number. If a contract number is not supplied then the client's accountNumber can be used.
mandate	O	update	get	Mandate object
→debitDay1	O	update	get	String. The day of the month the debit order must be collected. Values: 1 – 30 or Last day

Sample Java code:

```
public void testAddClient() {
    DefaultHttpClient httpClient = new DefaultHttpClient();
    try {

        URI url = new
        URIBuilder().setPath("https://secure.billdozer.com/api/rest/client/add").build();

        HttpPost httpPost = new HttpPost(url);

        httpPost.setHeader("apikey", "xyz");
        httpPost.setHeader("password", "xyz");

        StringEntity input = new StringEntity("{json formatted request}");
        input.setContentType(MediaType.TEXT_PLAIN);
        httpPost.setEntity(input);

        HttpResponse httpResponse = httpClient.execute(httpPost);
        if (httpResponse.getStatusLine().getStatusCode() != 200) {
            result = httpResponse.getStatusLine().getStatusCode() + " " +
            httpResponse.getStatusLine().getReasonPhrase();
            HttpEntity httpEntity = httpResponse.getEntity();
            result = result + ": " + EntityUtils.toString(httpEntity);
            return;
        }

        HttpEntity httpEntity = httpResponse.getEntity();
        result = EntityUtils.toString(httpEntity);

        ObjectMapper objectMapper = new ObjectMapper();
        com.billdozer.model.api.Response response = objectMapper.readValue(result,
        com.billdozer.model.api.Response.class);

        System.out.println(response.toString());

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        httpClient.getConnectionManager().shutdown();
    }
}
```