



Accounting Transaction JSON API

Table of Contents

Document Version	2
	2
Introduction	3
Http Post Parameters	3
Account Status:	4
Get	4
Invoice:	5
Save	5
Receipt:	11
Save	11
	14
Transaction:	15
Get	15
List	15
Recurring Invoice:	16
Add	16
Update	20
Get	22
	22
Delete Recurring Invoice Line Items	22
List	23
Parameter Listing:	24
Charge Calculator	27
Sample Java code:	28

Document Version

Version	Date	Change
1	28/04/2015	Initial draft
2	14/10/2016	Invoice/Save, Receipt/Save and Receipt/Get methods
3	10/11/2017	Account status method and changes to add recurring invoice
4	17/11/2017	New Update recurring invoice and get list recurring invoices by client reference
5	23/11/2018	Delete Recurring Invoice Line Item.
6	07/02/2024	Changes to invoice/save and recurring invoice list
7	15/08/2024	Changes to invoice/save - line item tax rates and batching for immediate invoices
8	01/09/2024	Changes recurringInvoice/save and recurringInvoice/update - add tax rate to line item and additional property overwrite for replacing recurring invoice line items.
9	02/10/2025	posted parameter to push invoice to accounting system

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.
 - The Api Key will automatically be generated.
 - Must have the Accounting role selected.
 - Enter a password and save.

Http Post Parameters

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

Account Status:

This function locates a customer with the specified unique reference and returns the status of their account.

Get

Get URL:

<https://secure.billdozer.com/api/rest/accountstatus/{client reference}/{days overdue}>

Sample JSON response:

Client does not exist:

```
{  
  "id":null,  
  "status":"NOK",  
  "messages":["Cannot find any Active client with unique reference ABC10"],  
  "additionalProperties":{}  
}
```

Client is in arrears:

```
{  
  "id":"39664",  
  "status":"NOK",  
  "messages":["Balance outstanding 1881.00 as at 2017-11-04"],  
  "additionalProperties":{}  
}
```

Client is in good standing:

```
{  
  "id":"39664",  
  "status":"OK",  
  "messages":[],  
  "additionalProperties":{}  
}
```

Get Account Status Parameter Listing

Parameter	Description
Client reference	Text. Required. The unique reference of the client whose account you wish to enquire about.
Days overdue	Integer. The number of days back from today that must be used to determine if the account is overdue. If not specified it defaults to 1.

Invoice:

Save

Post URL:

<https://secure.billdozer.com/api/rest/invoice/save>

Saves or updates an invoice, update is based on either the id or invoice reference. The best way to apply tax rates is to use them inside clientTransactionLines tag (for each line item) where they can be referenced by either the id or by the tax rate code (preferably). If the tax rate is specified at the invoice level then it must be referenced using the id. Tax rates can be viewed and added under General Settings.

If Immediate is false (and collectOnContract is true) then the invoice will be batched for debit order using the contract debit order day. If immediate is true then it will be batched on the due date.

If your Billdozer account is connected to an accounting system (Xero) and you want the invoice pushed then specify "posted" : true

Sample JSON request to add:

```
{  
  "reference": "INV0010010",  
  "type": "INVOICE",  
  "collectOnContract": true,  
  "info": "Test WS Info",  
  "dateIssued": "2025-10-02T00:00:00UTC",  
  "dateDue": "2025-10-18T00:00:00UTC",  
  "currency": "ZAR",  
  "billerData1": "Test WS invoice/save",  
  "billerData2": "",  
  "arrears": false,  
  "immediate": false,  
  "testMode": false,  
  "posted": false,  
  "clientTransactionLines": [  
    {  
      "accountingCode": {  
        "code": "100"  
      },  
      "description": "sales one",  
      "quantity": 1.0,  
      "unitAmountExcludingTax": 900,  
      "unitAmountIncludingTax": 0,  
      "billerTaxRate": {  
        "accountingPackageTaxRateCode": "OUTPUT3"  
      },  
      "taxable": true  
    }  
  ],  
  "client": {  
    "id": 834805,  
    "accountNumber": "TOB248"  
  },  
  "clientContract": {  
    "id": 12641211,  
    "contractNumber": "TOB248"  
  },  
  "billerTaxRate": {  
    "id": 1651  
  },  
  "additionalProperties": {  
    "sendImmediate": false  
  }  
}
```

}

Sample JSON request to update:

```
{  
  "id":8824050,  
  "type": "INVOICE",  
  "clientContactExclusive": false,  
  "collectOnContract": false,  
  "info": "TEST INFO",  
  "dateIssued": "2024-02-28T00:00:00UTC",  
  "dateDelivery": "2024-02-28T00:00:00UTC",  
  "dateDue": "2024-02-28T00:00:00UTC",  
  "currency": "ZAR",  
  "billerData1": "Test WS invoice/save",  
  "billerData2": "",  
  "arrears": false,  
  "testMode": false,  
  "clientTransactionLines": [  
    {  
      "accountingCode": {  
        "code": "100"  
      },  
      "description": "sales one",  
      "quantity": 4.5,  
      "unitAmountExcludingTax": 295,  
      "unitAmountIncludingTax": 0,  
      "taxable": true  
    }  
  ],  
  "client": {  
    "id": 834805,  
    "accountNumber": "TOB248"  
  },  
  "clientContract": {  
    "id": 12641211,  
    "contractNumber": "TOB248"  
  },  
  "billerTaxRate": {  
    "id": 1651  
  },  
  "additionalProperties": {  
    "sendImmediate": false  
  }  
}
```

Sample JSON response:

Failed:

```
{  
  "id":null,  
  "status": "NOK",  
  "messages": [  
    "Reference: INV003 already in use."  
  ],  
  "additionalProperties": {  
    "isTestMode": true  
  }  
}
```

```
}

Success:
{
  "id":"20151",
  "status":"OK",
  "messages":[

  ],
  "additionalProperties":{

    "reference":"INV006",
    "isTestMode":false
  }
}

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

  ],
  "additionalProperties":{

    "reference":"INV006",
    "isTestMode":true,
    "resultJson":"{ full transaction object returned in json format }"
  }
}
```

Invoice Parameter Listing

*Flag: O = Optional, R = Required

Name	Flag	Availability	Description
id		get	Long. Unique id of this invoice.
type	R	add	String. The type of transaction object. Either "INVOICE", "RECEIPT", "CREDIT NOTE", "DEBIT NOTE", "ADJUSTMENT"
paymentStatus	R	add	get
			Ignore this parameter, it is automatically set to UNPAID. If you want to mark it as PAID then you need to create a receipt
dateIssued	R	add	get
			String. ISO8601. The date listed on the invoice. This is usually the same as the created date, however it is different when invoices are generated ahead of time.
dateDue	R	add	get
			String. ISO8601. The date at which the invoice is due.
dateDelivery	O	add	get
			String. ISO8601. The date the goods or service was delivered.
reference	O	add	get
			String. The unique reference of the invoice. This is not strictly digits only, for example "INV002/15" is valid. If this value is omitted then the reference is automatically generated.
poNumber	O	add	get
			String. An optional purchase number to show for the invoice.
currency	R	add	get
			String. Currency in ISO 4217 format charged in. e.g. ZAR
discount	O	add	get
			Integer. Indicating the discount that was applied on the invoice
paymentTerms	O	add	get
			Integer. Indicating the paymentTerms that was applied on the invoice
clientContactExclusive	O	add	get
			Boolean, set to true if the invoice must only be sent to the assigned contact in this transaction. If false, will be sent to all recipients marked as notifyEmail in the Client.
collectOnContract	O	add	get
			Boolean, set to true if you would like this invoice to be collected in a debit order batch.
info	O	add	get
			String. Text information field
billerData1	O	add	get
			String. Additional Text information field
billerData2	O	add	get
			String. Additional Text information field
arrears	O	add	get
			Boolean. If true then this is an arrears invoice.
testMode	O	add	get
			Boolean. Default false. If set to true then additions and updates are not committed to the database. The

Name	Flag	Availability		Description
				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.
clientTransactionLines	R	add	get	An array of all the invoice lines
↳ accountingCode	O	add	get	Accounting Item Code Object
↳↳ code	R	add	get	String. Code used to lookup the object
↳ id			get	The id of the invoice line
↳ description		add	get	<p>String The description displayed on the invoice.</p> <p>If description is left blank and AccountingCode object is set then the description is used from the AccountingCode object.</p>
↳ quantity	R	add	get	<p>Amount. If quantity is left blank and AccountingCode object is set then the quantity is used from the AccountingCode object.</p>
↳ unitAmountExcludingTax	R	add	get	<p>Amount. The cost per unit of the line excluding tax.</p> <p>PLEASE NOTE: This field is required if your tax settings is set to default exclude tax.</p> <p>If amount is left blank and AccountingCode object is set then the amount is used from the AccountingCode object.</p>
↳ unitAmountIncludingTax	O/R	add	get	<p>Amount. The cost per unit of the line including tax.</p> <p>PLEASE NOTE: This field is required if your tax settings is set to default include tax.</p> <p>If amount is left blank and AccountingCode object is set then the amount is used from the AccountingCode object.</p>
↳ taxable	R	add	get	Boolean. Indicate if tax must be calculated for this line. Please see BillerTaxRate section below.
client	R	add	get	A client object that the invoice must be generated for.
↳ id	R	add	get	Long. Unique id of the client used to lookup the client. Or
↳ accountNumber	O/R	add	get	If id is not available then can use the accountNumber to lookup the client object.
ClientContract	O	add	get	The contract that this transaction must be assigned to. If the collectOnContract property is set to true then this object is mandatory.
↳ id	R	add	get	Long. The unique id of the contract to perform a lookup.
↳ contractNumber	Or R	add	get	String. Optionally can use the contractNumber to find the

Name	Flag	Availability	Description	
			object if the id is unknown.	
ClientContact	O	add	get	The Client Contact that this invoice must be assigned to.
↳ id	R	add	get	Long. The unique id of the contract to perform a lookup.
↳ name	Or R	add	get	String. Additionally use the full name to lookup the contact.
↳ idNumber	Or R	add	get	String. Additionally use id number to lookup the contact.
↳ emailAddress	Or R	add	get	String. Additionally use email address to lookup the contact.
billerTaxRate	O	add	get	Object. Optional. The tax object to calculate tax. If this object is left blank then the default Biller Tax Rate object will be used.
↳ id	R	add	get	Long. The unique id of the BillerTaxRate object to perform the lookup.
↳ description			get	String. Short description.
↳ taxPercentage			get	Amount. Percentage. E.g. 14.000
↳ currency			get	String. Currency in ISO 4217 format. e.g. ZAR
↳ defaultTaxRate			get	Boolean. Is this the default tax rate to be used in tax calculations
additionalProperties		add	get	Name value pair list. Used for future requirements,

Response Parameter Listing:

Name	Description
id	The id of the accounting transaction 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.
↳ reference	String. The reference 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.

Receipt:

Save

Post URL:

<https://secure.billdozer.com/api/rest/receipt/save>

Sample JSON request:

```
{  
  "clientId":56451,  
  "contractId":null,  
  "reference":"",
  "info":"Payment thank-you",
  "dateIssued":"2016-10-15T09:00:00UTC",
  "currency":"ZAR",
  "amountIncludingTax":1425.00,
  "billerCode":null,
  "billerData1":"Created via API",
  "billerData2":null,
  "billerPaymentMethodId":38,
  "billerPaymentProviderId":null,
  "additionalProperties":{  
    "sendImmediate":false
  },
  "allocations":[
    {
      "invoiceId":173601,
      "reference":"",
      "amountAllocated":0.00
    },
    {
      "invoiceId":173202,
      "reference":"",
      "amountAllocated":0.00
    }
  ],
  "testMode":false
}
```

Sample JSON response:

Failed:

```
{  
  "id":null,  
  "status":"NOK",  
  "messages": [  
    "Receipt needs to have either a payment method or a payment provider id."  
  ],  
  "additionalProperties": {  
    "isTestMode":true  
  }  
}
```

Success:

```
{  
  "id":"20151",  
  "status":"OK",  
  "messages": [  
  ],  
  "additionalProperties": {  
    "reference":"R-546",  
    "isTestMode":false  
  }  
}  
  
{  
  "id":"0",  
  "status":"OK",  
  "messages": [  
  ],  
  "additionalProperties": {  
    "reference":"R-546",  
    "isTestMode":true,  
    "resultJson":"{ full transaction object returned in json format }"  
  }  
}
```

Receipt Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Description
clientId	R	Billdozer of the client for whom this receipt is issued
contractId	O	Billdozer id of the contract. Set to null if no value is provided
reference	O	String. The unique reference of the receipt. This is not strictly digits only, for example "REC00243" is valid. If this value is omitted then the reference is automatically generated.
Info	O	String. The info field is displayed when the receipt is listed under receipt management and also on the client statement receipt line items.
dateIssued	R	String. ISO8601. The date listed on the receipt. This is usually the same as the created date.
Currency	O	String. Currency in ISO 4217 format. e.g. ZAR. If not specified it will default the to biller currency.
amountIncludingTax	O	The total of the receipt
billerCode	O	String. Additional 50 character Text information field
billerData1	O	String. Additional 100 character Text information field
billerData2	O	String. Additional 100 character Text information field
billerPaymentMethodId	O	If the receipt is to a bank account then put the id here. To view bank account ids see General Settings – Bank Accounts, Billdozer Id column. Either this or billerPaymentProviderId must be populated.
billerPaymentProviderId	O	If the receipt is via a payment gateway then put the id here. To view the gateway ids see General Settings – Payment Provider, Billdozer Id column. Either this or billerPaymentMethodId must be populated
additionalProperties		Name value pair list. Used for future requirements,
Allocations	R	This is the list of invoices that the receipt is allocated and the amount that is allocated to each.

Response Parameter Listing:

Name	Description
id	The id of the accounting transaction 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.
↳ reference	String. The reference 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.

Transaction:

Get

Get URL:

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

List

Get URL:

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

Response:

```
{ "status": "OK", "message": " ClientId = 49247", "type": "INVOICE", "results": [ // List of accounting transactions ] }
```

}

List Transactions 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.
clientId	Long. Optional. The id of the client whom you want to retrieve a list of transactions. If a clientId is provided then no type is required because it will provide all transactions for that client.
fromDateTime	DateTime. Optional. Return all transactions created or modified from this date and time UTC.
type	String. Required if not Client id is provided. The type of transaction object. Either "INVOICE", "RECEIPT", "CREDIT NOTE", "DEBIT NOTE", "ADJUSTMENT" or "QUOTE"

Recurring Invoice:

Add

Post URL:

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

Parameter Listing:

*Flag: O = Optional, R = Required

active	R	Indicates whether this recurring invoice is active or not. An active repeating will be processed by Billdozer
runDate	R	The date that the first invoice must be generated
runInterval	R	WEEKLY, BI-WEEKLY, MONTHLY, ANNUALLY, OTHER. Where OTHER you must populate the runMonth fields to indicate which months.
RunMonth01 >> 12	R	Month of the year where runInterval = OTHER. 01 is January.
runDay	R	The day of the week to run where runInterval = WEEKLY or BI-WEEKLY
Term	R	The term in months
billerdData1	O	This is the main comments field
billerdData2	O	Additional comments field
collectOnContract	R	True or false flag that indicates whether the invoices are to be collected by debit order. If true then the payment method and debit order day from the contract will be used to calculate the collection date of each invoice.
clientContract	R	The id of the contract that must be assigned to this repeating invoice.
billerdTaxRate	O	The id of the tax rate applicable to this repeating invoice.
clientContractEscalation	O	The id of the annual escalation method that must be applied to the repeating invoice.
clientContact	O	The id of the contact person that must receive the invoices
clientContactExclusive	O	True or false flag that indicates that the contact person specified must exclusively receive the invoices rather than being mailed to all contact persons that are marked to receive email invoices.
additionalProperties	O	Name value pair list.
clientContractChargeLines	R	
↳ accountingCode	R	Item code of the line item
↳ quantity	R	Quantity
↳ description	R	Description
↳ UnitAmountExcluding	O	Use this if prices exclude tax (as defined in General Settings - Tax). If not specified the item price will be looked up from the standard item code.
↳ UnitAmountIncluding	O	Use this if working inclusive of tax (as defined in General Settings - Tax)

↳billerTaxRate	O	The id or code of the tax rate applicable to this repeating invoice. The Tax rate code can be set under General Settings - Tax Rates and the field accountingPackageTaxRateCode field can be specified in the BillerTaxRate JSON.
----------------	---	---

Sample JSON request:

```
{ "active":true,
  "runDate":"2024-08-21T00:00:00UTC",
  "runInterval":"MONTHLY",
  "runMonth01":false,
  "runMonth02":false,
  "runMonth03":false,
  "runMonth04":false,
  "runMonth05":false,
  "runMonth06":false,
  "runMonth07":false,
  "runMonth08":false,
  "runMonth09":false,
  "runMonth10":false,
  "runMonth11":false,
  "runMonth12":false,
  "runDay":1,
  "term":0,
  "billerData1":"Test Comment",
  "billerData2":null,
  "collectOnContract":true,
  "clientContract":{
    "id":59109
  },
  "billerTaxRate":{
    "id":51
  },
  "clientContractEscalation":{
    "id":58473
  },
  "testMode":false,
  "clientContractChargeLines":[
    {
      "accountingCode":{
        "code":"100",
        "description":"sales one",
        "quantity":2.000,
        "term":0,
        "unitAmountExcludingTax": 200.00,
        "unitAmountIncludingTax": 0.00,
        "billerTaxRate": {
          "accountingPackageTaxRateCode": "OUTPUT3"
        },
        "taxable":true
      }
    }
  ],
  "additionalProperties":{
    "example":"xyz"
  }
}
```

Sample JSON response:

Failed:

```
{  
  "id":null,  
  "status":"NOK",  
  "messages": [  
    "Error message."  
  ],  
  "additionalProperties": {  
    "isTestMode":true  
  }  
}
```

Success:

```
{  
  "id":"20151",  
  "status":"OK",  
  "messages": [],  
  "additionalProperties": {  
    "isTestMode":true,  
    "resultJson": "{ full transaction object returned in json format }"  
  }  
}
```

Response Parameter Listing:

Name	Description
id	The id of the recurring invoice to be 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.

Update

Post URL:

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

This web service method is used to amend the line items on a recurring invoice, and does not allow you to change the run date, start date or frequency. Any line items specified in this method will be added to the existing recurring invoice, unless the **overwrite** parameter in the additional properties is true in which case all the existing lines will be replaced.

Depending on the tax settings of your Billdozer account you can provide either an exclusive or inclusive unit price. If no unit amount is provided then the price will be taken from the item code.

Note that changes made to recurring invoices will not affect invoices that have already been generated within the days ahead period. Days ahead is the number of days from today that invoices with issue dates falling within this period will be generated. It can be set from General Settings - Billing Rules -Invoicing tab: # Days in advance to generate invoices

id	R	Id of the recurring invoice to be updated
clientContractChargeLines	R	
↳ accountingCode	R	Item code of the line item
↳ quantity	R	Quantity
↳ description	R	Description
↳UnitAmountExcluding	O	Use this if prices exclude tax (as defined in General Settings - Tax). If not specified the item price will be looked up from the standard item code.
↳UnitAmountIncluding	O	Use this if working inclusive of tax (as defined in General Settings - Tax)
↳billerTaxRate	O	The id or code of the tax rate applicable to this repeating invoice.
additionalProperties	O	Name value pair list. Use overwrite:true to replace existing

Sample JSON request:

```
{  
  "id":714100,  
  "testMode":false,  
  "clientContractChargeLines": [  
    {"accountingCode":{  
      "code":"100"  
    },  
      "description":"sales one",  
      "quantity":2.000,  
      "term":0,  
      "unitAmountExcludingTax": 200.00,  
      "unitAmountIncludingTax": 0.00,  
      "taxable":true  
    }  
  ],  
  "additionalProperties":{  
    "example":"xyz",  
    "overwrite":true  
  }  
}
```

Sample JSON response:

Failed:

```
{ "id":"1739276",  
  "status":"NOK",  
  "messages":["Invalid accounting code: ICE004"],  
  "additionalProperties":{  
    "isTestMode":false  
  }  
}
```

Success:

```
{ "id":"1739276",  
  "status":"OK",  
  "messages":[],  
  "additionalProperties":{  
    "isTestMode":false  
  }  
}
```

Get

Get URL:

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

Delete Recurring Invoice Line Items

This method will delete the recurring invoice line items that are supplied in the input request.

Post URL:

<https://secure.billdozer.com/api/rest/recurringInvoiceLines/delete>

Sample JSON request:

```
{  
  "id":714100,  
  "testMode":false,  
  "clientContractChargeLines": [  
    {  
      "id":456478  
    },  
    {  
      "id":456479  
    }  
  ]  
}
```

Sample JSON response:

Failed:

```
{ "id":"714100",  
  "status":"NOK",  
  "messages":["No recurring invoice id specified"],  
  "additionalProperties":{  
    "isTestMode":false  
  }  
}
```

Success:

```
{ "id":"714100",  
  "status":"OK",  
  "messages":[],  
  "additionalProperties":{  
    "isTestMode":false  
  }  
}
```

List

Get URL:

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

List Recurring Invoices Filters

Following parameters must be put into the Http Header. Note that if no client reference, client id or fromDate are provided then all repeating invoices that are active and have an active client are returned for the pages specified

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.
clientUniqueReference	Text. Optional. The unique reference of the client whose account you wish to enquire about. Where provided this is used in preference to the clientId parameter
clientId	Long. Optional. The id of the client whom you want to retrieve a list of recurring invoices.
fromDateTime	DateTime. Optional. Return all recurring invoices created or modified from this date and time UTC.

Response:

```
{  
  "status": "OK", // OK or NOK  
  "message": null, // Error message  
  "type": "RecurringInvoice",  
  "page": 1,  
  "perPage": 15,  
  "totalResults": 2,  
  "results": [ // List of recurring invoices  
  ]  
}
```

Parameter Listing:

*Flag: O = Optional, R = Required

Name	Flag	Availability	Description
id		get	Long. Unique id of this recurring invoice.
active	R	add	Boolean. Recurring Invoice is active or not.
runDate	R	add	Date. The date that this recurring invoice must run and generate an invoice.
runInterval	R	add	String. "DAILY", "WEEKLY", "MONTHLY", "ANNUALLY", or "OTHER" If "OTHER" is selected then runMonth01 must be set.
runMonth01 - runMonth02	O	add	Boolean. If "OTHER" runInterval selected then indicate which month the recurring invoice must run. E.g. runMonth01 = January, runMonth12 = December.
runDay	O	add	Integer. If runInterval is WEEKLY, then indicate which day the charge must run. 1= Monday 7 = Sunday
term	O	add	Integer. How many months must the recurring invoice run for. Default is 0, i.e. Recurring Invoice will continue collecting if the term is 0
billerData1	O	add	String. Additional comments.
billerData2	O	add	String. Additional comments.
collectOnContract	O	add	Boolean. If set to true then a debit order invoice will be created against the contract and payment method for this recurring invoice.
clientContactExclusive	O	add	Boolean, set to true if the invoice must only be sent to the assigned contact in this transaction. If false, will be sent to all recipients marked as notifyEmail in the Client.
testMode	O	add	Boolean. Default false. If set to true then additions and updates are not committed to the database. The transaction id will also not automatically created. "resultJson" property will also be returned in the Response which contains the full json of the transaction object for

Name	Flag	Availability	Description	
			debugging.	
ClientContract	O	add	get	The contract that the recurring invoice must be assigned to
↳ id	R	add	get	Long. The unique id of the contract to perform a lookup.
clientContractEscalation	O	add	get	The contract escalation.
↳ id	R	add	get	The id of the contract escalation.
ClientContact	O	add	get	The Client Contact that this recurring invoice must be assigned to.
↳ id	R	add	get	Long. The unique id of the contract to perform a lookup.
billerTaxRate	O	add	get	Object. Optional. The tax object to calculate tax. If this object is left blank then the default Biller Tax Rate object will be used.
↳ id	R	add	get	Long. The unique id of the BillerTaxRate object to perform the lookup.
clientContractChargeLines	R	add	get	An array of all the invoice lines
↳ accountingCode	O	add	get	Accounting Item Code Object
↳↳ code	R	add	get	String. Code used to lookup the object
↳ description	O/R	add	get	String The description displayed on the invoice. If description is left blank and AccountingCode object is set then the description is used from the AccountingCode object.
↳ quantity	R	add	get	Amount. If quantity is left blank and AccountingCode object is set then the quantity is used from the AccountingCode object.
↳ term	O	add	get	Integer. How many months must the recurring invoice line item run for. Default is 0, i.e. Recurring Invoice line item will continue collecting if the term is 0
↳ unitAmountExcludingTax	R	add	get	Amount. The cost per unit of the line. If amount is left blank and AccountingCode

Name	Flag	Availability		Description
				object is set then the amount is used from the AccountingCode object.
↳ taxable	R	add	get	Boolean. Indicate if tax must be calculated for this line.
additionalProperties		add	get	Name value pair list. Used for future requirements,

Charge Calculator

Get URL:

<https://secure.billdozer.com/api/rest/chargeCalculator>

Response:

```
{  
  "status": "OK", // OK or NOK  
  "message": null, // Error message  
  "type": "INVOICE",  
  "page": 1, // Will always be 1, i.e. no pagination  
  "perPage": 12, // Total invoices returned  
  "totalResults": 12, // Total invoices returned  
  "results": [ // List of invoices  
  ]  
}
```

Charge Calculator Filters

Following parameters must be put into the Http Header

Http Header Parameter	Description
chargeId	Long. This is the recurring invoice id.
processDate	Date from which to calculate future invoices. If this parameter is not provided the default will be today.
numberOfDays	Integer. Number of days ahead to calculate future invoices. Max 1095. If this parameter is not provided the default will be 365.

Sample Java code:

```
public void testAddInvoice() {
    DefaultHttpClient httpClient = new DefaultHttpClient();
    try {
        URI url = new
URIBuilder().setPath("https://secure.billdozer.com/api/rest/transaction/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();
    }
}
```