SmartAccounts API

Version	Date	Comment
1	30.11.2017	First version
1.1	06.09.2018	New method on client invoices and a few typo fixes

SmartAccounts API is in essence a RESTful JSON (not all the principles of REST are implemented 100%) communications channel between external applications and SmartAccounts. It can be used to synchronize data from external applications (for example e-commerce platform to SmartAccounts) or to query data from SmartAccounts for better reporting.

Contents

General principles

Depending on the data synchronization principles the API services are divided into two groups:

1. Data for which changes cannot be queried;
2. Data for which changes can be queried.

We'll cover these two options in more detail below.

Data for which changes cannot be queried

If the API service is of this type, the external application using SmartAccounts API cannot ask for changes in the data after date dd.MM.yyyy and GET method can only query the list of objects in the system. This means that to sync data between the systems the API client must get the list of objects from SmartAccounts and look for changes between its local state.

This type of service is typically implemented for objects which do not change often: currencies, countries, document templates etc.

Data for which changes can be queried

This type of service is implemented for data which changes often: client invoices, purchase invoices, clients etc etc. These services allow the API client to GET only objects which have been added, updated or deleted after the last sync with the external system with second precision (dd.MM.yyyy_HH:mm:ss).

In order to take advantage of this logic the external system must save the date of last sync and for changes after that time. Of course, you can also sync data by local list vs remote list method described above but that would be impractical.

Pagination

The responses to GET methods of these services are paginated. One page currently consists of 100 objects (this might change and your implementation must not rely on this) and if the filtering parameters match more objects than that your implementation must ask for the invoices again with a new pageNumber (1,2,3 etc). For example, to sync changes for sales invoices after the last sync you should make requests to our GET method until the response JSON contains a parameter value hasMoreEntries = false.

Deleted objects

For this type of data sync it's always a question – how does the external system know which invoices have been deleted after the last sync was made. The API has a solution for this as well.

So let's ask for changes in client invoices with these parameters:

- dateFrom = 01.01.2018
- dateType = modifydate
- pageNumber = 1 (or empty, 1 is default)

When the pageNumber is set to 1 or is empty the resulting JSON will contain a JSONArray with the name of "**deleted**" which contains the list of objects deleted after the last sync. So in order to remove the same invoices from your system you need to go through the list and remove the invoices from your local system with SmartAccounts id in the deleted list.

Security

SmartAccounts uses the following measures to secure the requests:

1. Stale requests – requests with a time difference more than 15 minutes are ignored as stale. **Important! You must send requests with Estonian timezone!**
2. Signatures – all the requests must be signed by the external application with a secret key.
3. Request limits for 1h and 24h.
4. Privileges and rights. By default, the API user group is created with privileges to access every function available. It is strongly recommended to narrow down the scope of access of the API user group under Settings – User groups to allow access to only those functions needed by the external application.

Every API request must contain the following parameters in the URL section:

Parameter	Data type	Description
timestamp	String	A timestamp with the date and time of the request. Format: ddMMyyyyHHmmss
apikey	String	The public API key of the company account
signature	String	Signature of the whole request – see „Signing the request"

Request limits

SmartAccounts only processes:

- 1000 requests per 24 hours;
- 60 requests per minute.

Requests exceeding these limits will get the response code of 503.

Respnse codes and validation

Code	Meaning	Description
200	OK	
400	Bad request	There is something wrong with the request itself. The most likely reason is that something which is not allowed to be null is null.
401	Unauthorised	The signature is missing or is not correct
500	Internal error	There was an internal error while processing the request. If the error persists please contact SmartAccounts.
503	Rate Limit Exceeded	See the request limits section of this document.
503	Service unavailable (billing error)	You have a billing issue with SmartAccounts.

Signing the request

In order to verify the origin of the request (the originator must know the secret API key) a signature is calculated and passed as a URL parameter to SmartAccounts. The signature is calculated with **HMAC-SHA-256** algorithm and then converted to **HEX**.

- If you're signing a GET request (only URL parameters are provided) include only the URL parameters as the input to the signature calculation;
- If you're signing a request with a body part as well, include that as well (**IMPORTANT! No space or line change in between!);**
- Do not include SmartAccounts server URL in the signature calculation.

To sign a request:

1. Fist compile the request URL up to the last parameter (which must be obviously be "signature") and BODY.
   a. URL
      If the URL of the request was:
      https://sa.smartaccounts.eu/api/articles/getarticleswarehousequantities:run?codes=kood1%2Ckood2%2Ckood3&timestamp=14022013133802&apikey=a066f7de6042458da916

      Only the following part is used as signature calculation:
      codes=kood1%2Ckood2%2Ckood3&timestamp=14022013133802&apikey=a066f7de6042458da916
   b. BODY
      If the request also includes a body part it must immediately follow the URL part with no spaces or any other characters in between.
2. 1.a + 1.b (optionally) is the input to the signature calculation algorithm. So now calculate the hash with your secret key and convert it to HEX value.

We've also included a calculator in SmartAccounts GUI to check if your signature is correct for the request. Please have a look at: Settings -> Connected services.

Add / edit request validation

The validation of the requests is done in two separate stages:

1. **Very basic validation** – for example, if a required field is null. The request will fail as soon as one of these basic validation rules is not met and SmartAccounts will not process the request any further. The reponse body JSON gives you more details on this so you can correct the issue and try again.
2. **Complex validation** – the request is trying to add an invoice to a period which is marked as locked by the users. Basic validation is passed but this complex validation is not. The request will fail with the same error code but this time the reponse error JSON will contain the list of all the rules which must be met to pass validation rules.

If validation fails the resulting JSON contains error keys which can be used systematically and a more human readable message as well. The messages for basic validation are currently only in English and you can choose which language is used to return complex validation messages via the URL of the API call:

- Use sa.smartaccounts.eu/et/api for Estonian messages;
- Use sa.smartaccounts.eu/en/api for English messages.

Other useful notes

These notes might come in handy when dealing with SmartAccounts API:

- All the API services reside at https://sa.smartaccounts.eu/api/...
- The decimal places are separated by a full stop. For example, the correct decimal for one hundred euros and two cents is 100.02
- Use URL encoding to send more complex values to SmartAccounts server. For example, if the request accepts a comma separated list of values as URL parameter it must be urlencoded first: codes=kood1%2Ckood2%2Ckood3.
- Design your application to respect the request limits of SmartAccounts. Is it really absolutely necessary to sync warehouse stock every 15 minutes?
- Design your application to handle failed requests as well. For example your requests will fail when you've forgot to pay your bill and it'd be very bad for your main business processes to fail for this simple reason. It's a good practice not to tie your main business processes with an external system so when the system is down or requests fail for any other reason your world stops as well.
- Read the response body part as well. It gives you more information about the reason why the request failed.
- POST body encoding must be UTF-8!
- Implement your API client side with a test environment first. Please ask SmartAccounts support for help on this issue. If your implementation makes changes to live accounting data, they cannot be rolled back.
- Edit requests must contain all the changed fields not just changes – in order to post the change to one data field only you must pass the whole object to SmartAccounts.
- The "uniqueness" of the objects in SmartAccounts is provided via the following parameters (in order of importance):
    - Id – used for invoices for example;
    - Code – used for articles for example;

- Name - used

One of these parameters must exist for every object.

API services

General - Files

SERVICE TYPE: Data for which changes cannot be queried

URL: /api/general/files:<methodname>

Available methods
get - get the list of files for one specific document
getDetails - get the details (incl content) of one specific file
add
delete

GET: possible input parameters					
docType					Document type to ask files for. Possible values are: CLIENT_INVOICE, CLIENT, VENDOR, VENDOR_INVOICE, ARTICLE, CLIENT_OFFER, CLIENT_ORDER
docId					Document id

GETDETAILS: possible input parameters					
docType					Document type to ask files for. Possible values are: CLIENT_INVOICE, CLIENT, VENDOR, VENDOR_INVOICE, ARTICLE, CLIENT_OFFER, CLIENT_ORDER
docId					Document id
fileId					File to fetch details for

GET / ADD - data fields					
	get	**.det**	**add**	**datatype**	**description**
docId		🟩	🟦	String	Document id to which this file belongs to
docType		🟩	🟦	String	Document type to which this file belongs to
fileId	🟩	🟩			Unique id for this file
fileName	🟩	🟩	🟦		Filename
description	🟩	🟩	▨		Additional description
fileContent		🟩	🟦		Content - BASE64 encoded string
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
fileId					Unique id for this object
fileName					Filename
description					Additional description
docId					Document id to which this file belongs to
docType					Document type to which this file belongs to

DELETE: input parameters					
docId					Document id to which this file belongs to
docType					Document type to which this file belongs to
fileId					Unique id for this file

Additional methods					

General – Entries

SERVICE TYPE: Data with change history
URL: /api/general/entries:<methodname>

Available methods
get
add
edit
delete
getpdf

GET: possible input parameters					
dateFrom					Filter objects by date
dateTo					Filter objects by date
dateType					Sets which dates are used in filtering. Possible values: **empty, date** - filter by entry date **modifydate** - added / modified date
filterDescription					Filter by document number or row description
id					Search for a specific entry
docType					Filter by entry type. Possible values are: **LEDGER_ENTRY** - manual ledger entry **CLIENT_INVOICE** **VENDOR_INVOICE** **BANK_PAYMENT** **CASH_PAYMENT** **VAT_RETURN** **WAREHOUSE_MOVEMENT** **PERIOD_DEPRECIATION** **PERIOD_SALARY**
paymentStatus					Filter by payment status. Possible values are: **emtpy** - filter not applied **unpaid** - unpaid invoices **overdue** - unpaid and overdue invoices
pageNumber					Page number
fetchRows					Return the invoice rows as well
fetchDocuments					Retrun the list of uploaded documents
fetchComments					Return internal information comments

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
docType	🟩		🟦	String	General ledger entry type. Only manual ledger entries can be added but all entries can be edited.

date	🟩	🟦	🟦	Date	Date
document	🟩	░	░	Date	Textfield for document number
currency	🟩			String	Only EUR entries can be added or edited through the API so currency is not needed as an input parameter
number	🟩			String	Document number in accounting.
amount	🟩			Decimal	Debit and credit sum amount. Cannot be passed to add or edit requests
comment	🟩	░	░	String	Internal information
documents	🟩			JSONArr	JSONArray of uploaded documents
fileName	🟩			String	
description	🟩			String	
rows	🟩	🟦	🟦		JSONArray of entry rows
account	🟩	🟦	🟦	String	Row account
description	🟩	🟦	🟦	String	
debitAmount	🟩	░	░	Decimal	Defaults to 0.00 if missing
creditAmount	🟩	░	░	Decimal	Defaults to 0.00 if missing
objectId	🟩	░	░	String	Row object
LEGEND		🟦			**mandatory**
		░			**optional**

ADD / EDIT: return value					
id					Id of the entry
amount					The total amount of the entry
number					Internal number for the entry
docType					LEDGER_ENTRY

DELETE: inout parameters					
id					Unique id for this object

Additional methods					
/entries:getpdf					Render a PDF file of the entry. Input parameters are: **id** - entry id

Settings – Accounts

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/accounts:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all accounts of the company

GET / ADD / EDIT - data fields

	get	**add**	**edit**	**datatype**	**description**
code	🟩	🟦	🟦	String	The code of the account
type	🟩	🟦	🟦	String	Account type. Possible values are: ASSET LIABILITY INCOME EXPENSE
descriptionEt	🟩	🟦	🟦	String	Description: Estonian
descriptionEn	🟩	░	░	String	Description: English
balanceSheetRow	🟩	░	░	String	Report row code: balance sheet
incomeStatementRow	🟩	░	░	String	Report row code: income statement
cashflowRow	🟩	░	░	String	Report row code: cashflow
LEGEND		🟦			**mandatory**
		░			**optional**

ADD / EDIT: return value					
code					The code of the account

DELETE: input parameters					
code					The code of the account

Additional methods					

Settings – Bank Accounts

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/bankaccounts:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all vat accounts

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
name	🟩	🟦	🟦	String	The name of the bank account
account	🟩	🟦	🟦	String	The general ledger account associated with the bank account
order	🟩	🟦	🟦	String	Display order
currency	🟩	🟦	🟦	String	Currency
iban	🟩	⬜	⬜	String	Bank account number
swift	🟩	⬜	⬜	String	Swift
forNetting	🟩	⬜	⬜	Boolean	Is this the default account for netting credit invoices and invoices automatically
showOnInvoice	🟩	⬜	⬜	Boolean	Show this account on invoice PDFs
otherInvoiceInfo	🟩	⬜	⬜	String	Additional information printed to PDFs
defaultEInvoiceAccount	🟩	⬜	⬜	Boolean	Is this the default account for e-invoices
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value					
name					The name of the bank account

DELETE: input parameters					
name					The name of the bank account

Additional methods					

Settings – Cash Accounts

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/cashaccounts:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
name	🟩	🟦	🟦	String	The name of the cash account
account	🟩	🟦	🟦	String	The general ledger account associated with the cash account
order	🟩	🟦	🟦	String	Display order
currency	🟩	🟦	🟦	String	Currency
incomingOrderNumber	🟩	□	□	String	The number set to the next incoming cash order
outgoingOrderNumber	🟩	□	□	Boolean	The number set to the next outgoing cash order
LEGEND		🟦			**mandatory**
		□			**optional**

ADD / EDIT: return value					
name					The name of the cash account

DELETE: input parameters					
name					The name of the cash account

Additional methods					

Settings – Countries

SERVICE TYPE: Data for which changes cannot be queried

URL: /api/settings/countries:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
code	🟩	🟦	🟦	String	The ISO code for the country
name	🟩	🟦	🟦	String	Name
euMember	🟩	▨	▨	Boolean	Is the country a member of the EU or not
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
code					The ISO code for the country

DELETE: input parameters					
code					The ISO code for the country

Additional methods					

Settings – Document templates

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/documenttemplates:<methodname>

Available methods
get

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
code	🟩			String	Unique code for the template
name	🟩			String	The user readable name for the template
available	🟩			String	Comma separated list of places where the template is used
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value

DELETE: input parameters

Additional methods					

Settings – Groups

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/groups:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
name	🟩	🟦	🟦	String	Group name
clients	🟩	░	░	Boolean	Can be used to group clients
vendors	🟩	░	░	Boolean	Can be used to group vendors
LEGEND		🟦			**mandatory**
		░			**optional**

ADD / EDIT: return value					
name					Group name

DELETE: input parameters					
name					Group name

Additional methods					

Settings – Objects

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/objects:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩	🟦	🟦	String	Unique id for this object
code	🟩	🟦	🟦	String	The code or class of the object
name	🟩	🟦	🟦	Boolean	The name of the object
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value					
id					Unique id for this object

DELETE: input parameters					
id					Unique id for this object

Additional methods					

Settings – Payment methods

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/paymentmethods:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
name	🟩	🟦	🟦	String	Unique name for payment method
account	🟩	🟦	🟦	String	The code or class of the object
currency	🟩	🟦	🟦	String	Currency
order	🟩	⬜	⬜	Integer	Display order for the payment method
isCash	🟩	⬜	⬜	Boolean	Is the payment method for cash payments
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value					
name					Unique name for payment method

DELETE: input parameters					
name					Unique name for payment method

Additional methods					

Settings – Report rows

SERVICE TYPE:	Data for which changes cannot be queried
URL:	/api/settings/reportrows:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
code	🟩	🟦	🟦	String	Unique code for the report row
reportType	🟩	🟦	🟦	String	Report type for this report row. Possible values: BALANCE INCOME CASHFLOW
descriptionEt	🟩	⬜	⬜	String	Estonian description
descriptionEn	🟩	⬜	⬜	String	English description
valueType	🟩	🟦	🟦	String	Report row type. Possible values: EMPTY BALANCE FORMULA TURNOVER
style	🟩	🟦	🟦	String	Apply styling in the report. Possible values: REGULAR INDENT BOLD
displayCondition	🟩	🟦	🟦	String	When to display this row in the report. Possible values: ALWAYS IFNOTNULL NEVER
formula	🟩	⬜	⬜	String	Formula for the row if the row has a type of FORMULA
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value					
code					Unique code for the report row

DELETE: input parameters					
code					Unique code for the report row

Additional methods					

Settings – Warehouses

GET: possible input parameters
None, the service always returns all objects

GET / ADD / EDIT - data fields

	get	add	edit	datatype	description
name	🟩			String	Warehouse name
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value

DELETE: input parameters

Additional methods

Settings – VatPcs

SERVICE TYPE: Data for which changes cannot be queried

URL: /api/settings/vatpcs:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters
None, the service always returns all vat percentages

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
vatPc	🟩		🟦	String	The code of the vat percentage
pc	🟩	🟦	🟦	Decimal	Percentage (20 is 20% for example)
descriptionEt	🟩	🟦	🟦	String	Description: Estonian
descriptionEn	🟩	▨	▨	String	Description: English
descriptionRu	🟩	▨	▨	String	Description: Russian
accountPurchase	🟩	▨	▨	String	Account code: purchase
accountSales	🟩	▨	▨	String	Account code: sales
order	🟩	▨	▨	String	Order number
custom	🟩	▨	▨	boolean	Shows if the vat percentage is company specific or in general use
inUse	🟩	▨	▨	boolean	Shows if the vat pc is in use currently (can be deleted or not)
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
vatPc					The code of the vat percentage

DELETE: input parameters					
vatPc					The code of the vat percentage to be deleted

Additional methods					

Purchase / Sales – Articles

SERVICE TYPE:	Data with change history
URL:	/api/purchasesales/articles:<methodname>

Available methods
get
add
edit
delete
getwarehousequantites

GET: possible input parameters					
searchString					Filter articles by code or name
code					Find specific article by code
modifiedFrom					Get articles modified after
modifiedTo					Get articles modified before
pageNumber					Page number
fetchDocuments					Fetch the list of uploaded documents

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
code	🟩	🟦	🟦	String	Unique identifier
description	🟩	🟦	🟦	String	Description
type	🟩	🟦	🟦	String	Article type. Possible values are: **PRODUCT** - product **SERVICE** - service **WH** - warehouse item
unit	🟩	▨	▨	String	Unit (kg, t, pc etc)
vatPc	🟩	▨	▨	String	Vat percentage
activeSales	🟩	▨	▨	Boolean	Is article active in sales
activePurchase	🟩	▨	▨	Boolean	Is article acitve in purchase
priceSales	🟩	▨	▨	Decimal	Sales price
pricePurchase	🟩	▨	▨	Decimal	Purchase price
priceOwn	🟩	▨	▨	Decimal	Own price
accountSales	🟩	▨	▨	String	General ledger account for sales
accountPurchase	🟩	▨	▨	String	General ledger account for purchase
accountWarehouse	🟩	▨	▨	String	General ledger account for warehouse
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	
updatedBy	🟩			String	

documents	🟩			JSONArr	Uploaded documents
fileName	🟩			String	File name
description	🟩			String	Description
LEGEND		🟦			**mandatory**
		⬜			**optional**

ADD / EDIT: return value					
code					Unique identifier

DELETE: input parameters					
code					Unique identifier

Additional methods					
/articles:getwarehousequantities					Get the warehouse quantities of specified articles. Input parameters **codes** - urlencoded string of comma separated values of max 10 article codes. The default system warehouse is used for the query. The result is a JSONObject with JSONArray **"quantities"** inside which have the following parameters: **code** - article code **quantity** - article quantity in warehouse **status** - possible values are: **OK** - warehouse article found **NOT-FOUND** - article was not found **NOT-WH** - not a warehouse article

Purchase / Sales – Clients

SERVICE TYPE:	Data with change history
URL:	/api/purchasesales/clients:<methodname>

Available methods
get
add
edit
delete
getbalance

GET: possible input parameters					
nameOrRegCode					Filter by name or regcode
id					Fetch a specific client
modifiedFrom					Fetch clients modified after
modifiedTo					Fetch clients modified until
pageNumber					Page number
fetchDocuments					Fetch the list of uploaded documents
fetchComments					Fetch internal comments
fetchAddresses					Fetch addresses
fetchContacts					Fetch contacts

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
group	🟩	░	░	String	Group name
name	🟩	🟦	🟦	String	
regCode	🟩	░	░	String	
vatNumber	🟩	░	░	String	VAT registration number
bankAccount	🟩	░	░	String	Bank account number
referenceNumber	🟩	░	░	String	Reference number
ignoreOnVatDeclaration	🟩	░	░	Boolean	Used to not include transactions with this partner in VAT declaration form
invoiceDueDate	🟩	░	░	Integer	Due date in days
invoiceInterest	🟩	░	░	Decimal	Interest on unpaid invoices
discount	🟩	░	░	Decimal	Client based discount
vatPc	🟩	🟦	🟦	String	Client based VAT percentage
pdfAdditionaldata	🟩	░	░	String	Additional information on PDF - always printed for this client
pdfTemplateInvoice	🟩	░	░	String	Template for sales invoice
pdfTemplateOffer	🟩	░	░	String	Template for offer

pdfTemplateOrder	🟩	▨	▨	String	Template for order
accountUnpaid	🟩	▨	▨	String	General ledger unpaid account
accountPrepayment	🟩	▨	▨	String	General ledger prepayment account
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	
updatedBy	🟩			String	
address	🟩	▨	▨	JSONObj	Address
country	🟩	▨	▨	String	2-letter ISO code for country
county	🟩	▨	▨	String	County
city	🟩	▨	▨	String	City
address1	🟩	▨	▨	String	Address1
address2	🟩	▨	▨	String	Address2
postalCode	🟩	▨	▨	String	Postcode
contacts	🟩	▨	▨	JSONArr	List of contacts
type	🟩	▨	▨	String	Type of contact. Possible values are: **EMAIL, PHONE, FAX, WWW, SKYPE, OTHER, PERSON, POST_ADDRESS**
value	🟩	▨	▨	String	Content (info@smartaccounts.eu for example)
description	🟩	▨	▨	String	Internal information ("The phone number of our contact person" for example)
comment	🟩	▨	▨	String	Internal information, comment
documents	🟩			JSONArr	The list of uploaded documents
fileName	🟩			String	File name
description	🟩			String	Description
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
clientId					Unique identifier
referenceNumber					Generated reference number

DELETE: input parameters					
id					Unique identifier

Additional methods					
/clients:getbalance					Feth outstanding balance of the partner: **id** - identifier **balanceDate** - the date of the balance **fetchDetails** - fetch details

Purchase / Sales – Client Invoices

SERVICE TYPE:	Data with change history
URL:	/api/purchasesales/clientinvoices:<methodname>

Available methods
get
add
edit
delete
getpdf

GET: possible input parameters					
dateFrom					Filter objects by date
dateTo					Filter objects by date
dateType					Sets which dates are used in filtering. Possible values: **empty** - same as "date" **date** -invoice date **entrydate** - entry date **duedate** - due date **modifydate** - added / modified date
clientName					Filter by client name
clientId					Filter by client id
id					Search for a specific invoice
invoiceNumber					Filter by invoice number
paymentStatus					Filter by payment status. Possible values are: **emtpy** - filter not applied **unpaid** - unpaid invoices **overdue** - unpaid and overdue invoices
pageNumber					Page number
fetchRows					Return the invoice rows as well
fetchDocuments					Retrun the list of uploaded documents
fetchComments					Return internal information comments

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
clientId	🟩	🟦	🟦	String	Unique identifer for the client
client	🟩			JSONObj	JSONObject with data for client, GET only!
id	🟩			String	
name	🟩			String	

regCode	🟩			String	
vatNumber	🟩			String	
type	🟩	░	░	String	Type. Null or CRE - credit invoice
date	🟩	🟦	🟦	Date	Date
dueDate	🟩	░	░	Date	Due date
entryDate	🟩	░	░	Date	Entry date
number	🟩			String	Number in accounting
invoiceNumber	🟩	░	░	String	Invoice number. If not provided (usual usecase), will be generated by SmartAccounts
invoiceNote	🟩	░	░	String	Additional information for PDF invoice
referenceNumber	🟩	░	░	String	Reference number for payment orders
currency	🟩	🟦	🟦	String	Currency (EUR for example)
exchangeRate	🟩	░	░	Decimal	Exchange rate (can be null or 1.00 for EUR). Mandatory for those currencies for which no ECB exchange rate can be found (Bitcoin for example).
interest	🟩	░	░	Decimal	Interest rate
amount	🟩			Decimal	Amount without VAT
roundAmount	🟩	░	░	Decimal	Rounding
vatAmount	🟩			Decimal	VAT amount
totalAmount	🟩	░	░	Decimal	Total amount incl VAT. If provided for add / edit requestts the total amount will be calculated by SmartAccounts as well and the difference will go to roundAmount. This is used to guarantee there is no difference in the amounts between the systems (because of different rounding rules for example)
outstandingAmount	🟩			Decimal	The outstanding amount of the invoice
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	
updatedBy	🟩			String	
pdfTemplate	🟩	░	░	String	Template to be used for PDFs
accountUnpaid	🟩			String	The general ledger account for the invoice (for add / edit this is taken from the client's data)
objectId	🟩	░	░	String	Object
emailLastSentBy	🟩			String	Details of the last e-mail sent
emailLastSentDate	🟩			Date	Details of the last e-mail sent
emailLastSentTo	🟩			String	Details of the last e-mail sent
omnivaLastSentDate	🟩			Date	Details of sending the invoice to Omniva

paymentMethod	🟩	▨	▨	String	If the invoice is marked as paid along with the invoice creation this field contains the reference to payment method used
paymentAmount	🟩	▨	▨	Decimal	Payment amount when paymentMethod is used
warehouseId	🟩	▨	▨	String	
baseForCreditInvoiceId	🟩	▨	▨	String	The id of the initial invoice when the type of the current invoice is CRE
offerId	🟩	▨	▨	String	The id of the offer when invoice is based on an offer
orderId	🟩	▨	▨	String	The id of the order when invoice is based on an order
periodicInvoiceId	🟩			String	Name of the periodic invoice when the invoice is created based on a periodic invoice
comment	🟩	▨	▨	String	Internal information
documents	🟩			JSONArr	JSONArray of uploaded documents
fileName	🟩			String	
description	🟩			String	
rows	🟩	🟦	🟦		JSONArray of invoice rows
code	🟩	🟦	🟦	String	Article code
description	🟩	🟦	🟦	String	
price	🟩	🟦	🟦	Decimal	
quantity	🟩	🟦	🟦	Decimal	
unit	🟩	▨	▨	String	
discount	🟩	▨	▨	Decimal	Discount percentage (for example 10% = 10)
vat	🟩			Decimal	VAT
vatPc	🟩	🟦	🟦	String	VAT percentage
sum	🟩			Decimal	Row total amount with VAT
order	🟩	▨	▨	Integer	Order
objectId	🟩	▨	▨	String	Row object
accountSales	🟩	▨	▨	String	General ledger sales account
accountWarehouse	🟩	▨	▨	String	General ledger warehouse account
accountPurchase	🟩	▨	▨	String	General ledger purchase account
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
invoiceId					Invoice id
clientId					The id of the client

number					Acconting number for the invoice
invoiceNumber					Invoice number
referenceNumber					
amount					Amount without VAT
vatAmount					VAT
totalAmount					Amount with VAT
roundAmount					Rounding amount (see description of totalAmount above)
dueDate					
interest					
warehouseId					

DELETE: input parameters					
id					Unique id for this object

Additional methods					
/clientinvoices:getpdf					Returns the PDF of the invoice. Input parameters **id** - invoice id **template** - optional, if left empty the default templates will be used
/clientinvoices:getxml					Returns the E-Invoice XML file. Input parameters **id** - invoice id

Purchase / Sales – Client Offers / Client Orders

These are two services with almost exact same behaviour but different URLs.

SERVICE TYPE:	Data with change history
URL:	/api/purchasesales/clientoffers(/clientorders):<methodname>

Available methods
get
add
edit
delete
getpdf
getstatuses
addstatus

GET: possible input parameters					
dateFrom					Filter objects by date
dateTo					Filter objects by date
dateType					Sets which dates are used in filtering. Possible values: **empty** - same as "date" **date** -offer / order date **modifydate** - added / modified date
clientName					Filter by client name
clientId					Filter by client id
id					Search for a specific offer / order
offerNumber / orderNumber					Filter by offer / order number
pageNumber					Page number
fetchRows					Return the offer / order rows as well
fetchDocuments					Retrun the list of uploaded documents
fetchComments					Return internal information comments

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
clientId	🟩	🟦	🟦	String	Unique identifer for the client
client	🟩			JSONObj	JSONObject with data for client, GET only!
id	🟩			String	
name	🟩			String	
regCode	🟩			String	
vatNumber	🟩			String	

date	🟩	🟦	🟦	Date	Date
dueDate	🟩	░	░	Date	Due date
nextActionDate	🟩	░	░	Date	Date of the next action
paymentDueDate	🟩	░	░	Date	Payment due date of the offer
number	🟩			String	Number in accounting
offerNumber / orderNumber	🟩	░	░	String	Offer / order number. If not provided (usual usecase), will be generated by SmartAccounts
offerNote / orderNote	🟩	░	░	String	Additional information for PDF
currency	🟩	🟦	🟦	String	Currency (EUR for example)
exchangeRate	🟩	░	░	Decimal	Exchange rate (can be null or 1.00 for EUR). Mandatory for those currencies for which no ECB exchange rate can be found (Bitcoin for example).
status	🟩	░	░		Current status of the offer / order
interest	🟩	░	░	Decimal	Interest rate
amount	🟩			Decimal	Amount without VAT
roundAmount	🟩	░	░	Decimal	Rounding
vatAmount	🟩			Decimal	VAT amount
totalAmount	🟩	░	░	Decimal	Total amount incl VAT. If provided for add / edit requestts the total amount will be calculated by SmartAccounts as well and the difference will go to roundAmount. This is used to guarantee there is no difference in the amounts between the systems (because of different rounding rules for example)
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	
updatedBy	🟩			String	
pdfTemplate	🟩	░	░	String	Template to be used for PDFs
objectId	🟩	░	░	String	Object
emailLastSentBy	🟩			String	Details of the last e-mail sent
emailLastSentDate	🟩			Date	Details of the last e-mail sent
emailLastSentTo	🟩			String	Details of the last e-mail sent
offerId	🟩	░	░	String	The id of the offer when the order is based on an offer. Only for client offers
comment	🟩	░	░	String	Internal information
documents	🟩			JSONArr	JSONArray of uploaded documents
fileName	🟩			String	
description	🟩			String	
rows	🟩	🟦	🟦		JSONArray of rows

code	🟩	🟦	🟦	String	Article code
description	🟩	🟦	🟦	String	
price	🟩	🟦	🟦	Decimal	
quantity	🟩	🟦	🟦	Decimal	
unit	🟩	▢	▢	String	
discount	🟩	▢	▢	Decimal	Discount percentage (for example 10% = 10)
vat	🟩			Decimal	VAT
vatPc	🟩	🟦	🟦	String	VAT percentage
sum	🟩			Decimal	Row total amount with VAT
order	🟩	▢	▢	Integer	Order
objectId	🟩	▢	▢	String	Row object
LEGEND		🟦			**mandatory**
		▢			**optional**

ADD / EDIT: return value					
offerId / orderId					Unique identifier
clientId					The id of the client
number					Internal number for the offer / order
offerNumber / orderNumber					Number
amount					Amount without VAT
vatAmount					VAT
totalAmount					Amount with VAT
roundAmount					Rounding amount (see description of totalAmount above)

DELETE: inout parameters					
id					Unique id for this object

Additional methods					
/clientoffers:getpdf /clientorders:getpdf					Render the PDF. Input parameters **id** - offer / order id **template** - optional, if left empty the default templates will be used
/clientoffers:getstatuses /clientorders:getstatuses					Get the list of statuses for this offer or order. Input parameters are: **id** - offer / order id
/clientoffers:addstatus /clientorders:addstatus					Sets the status of the offer / order to a new value. Possible input parameters are: **id, status**.

Purchase / Sales – Vendors

SERVICE TYPE: Data with change history

URL: /api/purchasesales/vendors:<methodname>

Available methods
get
add
edit
delete
getbalance

GET: possible input parameters					
nameOrRegCode					Filter by name or regcode
id					Fetch a specific vendor
modifiedFrom					Fetch vendors modified after
modifiedTo					Fetch vendors modified until
pageNumber					Page number
fetchDocuments					Fetch the list of uploaded documents
fetchComments					Fetch internal comments
fetchAddresses					Fetch addresses
fetchContacts					Fetch contacts

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
group	🟩	▨	▨	String	Group name
name	🟩	🟦	🟦	String	
regCode	🟩	▨	▨	String	
vatNumber	🟩	▨	▨	String	VAT registration number
bankAccount	🟩	▨	▨	String	Bank account number
referenceNumber	🟩	▨	▨	String	Reference number
ignoreOnVatDeclaration	🟩	▨	▨	Boolean	Used to not include transactions with this partner in VAT declaration form
invoiceDueDate	🟩	▨	▨	Integer	Due date in days
invoiceInterest	🟩	▨	▨	Decimal	Interest on unpaid invoices
vatPc	🟩	🟦	🟦	String	Vendor based VAT percentage
accountUnpaid	🟩	▨	▨	String	General ledger unpaid account
accountPrepayment	🟩	▨	▨	String	General ledger prepayment account
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	

updatedBy	🟩			String	
address	🟩	░	░	JSONObj	Address
country	🟩	░	░	String	2-letter ISO code for country
county	🟩	░	░	String	County
city	🟩	░	░	String	City
address1	🟩	░	░	String	Address1
address2	🟩	░	░	String	Address2
postalCode	🟩	░	░	String	Postcode
contacts	🟩	░	░	JSONArr	List of contacts
type	🟩	░	░	String	Type of contact. Possible values are: **EMAIL, PHONE, FAX, WWW, SKYPE, OTHER, PERSON, POST_ADDRESS**
value	🟩	░	░	String	Content (info@smartaccounts.eu for example)
description	🟩	░	░	String	Internal information ("The phone number of our contact person" for example)
comment	🟩	░	░	String	Internal information, comment
documents	🟩			JSONArr	The list of uploaded documents
fileName	🟩			String	File name
description	🟩			String	Description
LEGEND		🟦			**mandatory**
		░			**optional**

ADD / EDIT: return value					
vendorId					Unique identifier
referenceNumber					Generated reference number

DELETE: input parameters					
id					Unique identifier

Additional methods					
/vendors:getbalance					Feth outstanding balance of the partner: **id** - identifier **balanceDate** - the date of the balance **fetchDetails** - fetch details

Purchase / Sales – Vendor Invoices

SERVICE TYPE:	Data with change history
URL:	/api/purchasesales/vendorinvoices:<methodname>

Available methods
get
add
edit
delete
getpdf

GET: possible input parameters					
dateFrom					Filter objects by date
dateTo					Filter objects by date
dateType					Sets which dates are used in filtering. Possible values: **empty** - same as "date" **date** -invoice date **entrydate** - entry date **duedate** - due date **modifydate** - added / modified date
vendorName					Filter by vendor name
vendorId					Filter by vendor id
id					Search for a specific invoice
invoiceNumber					Filter by invoice number
paymentStatus					Filter by payment status. Possible values are: **emtpy** - filter not applied **unpaid** - unpaid invoices **overdue** - unpaid and overdue invoices
pageNumber					Page number
fetchRows					Return the invoice rows as well
fetchDocuments					Retrun the list of uploaded documents
fetchComments					Return internal information comments

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
vendorId	🟩	🟦	🟦	String	Unique identifer for the vendor
vendor	🟩			JSONObj	JSONObject with data for vendor GET only!
id	🟩			String	
name	🟩			String	

regCode	🟩			String	
vatNumber	🟩			String	
date	🟩	🟦	🟦	Date	Date
dueDate	🟩	░	░	Date	Due date
entryDate	🟩	░	░	Date	Entry date
number	🟩			String	Number in accounting
invoiceNumber	🟩	🟦	🟦	String	Invoice number
referenceNumber	🟩	░	░	String	Reference number for payment orders
currency	🟩	🟦	🟦	String	Currency (EUR for example)
exchangeRate	🟩	░	░	Decimal	Exchange rate (can be null or 1.00 for EUR). Mandatory for those currencies for which no ECB exchange rate can be found (Bitcoin for example).
interest	🟩	░	░	Decimal	Interest rate
amount	🟩			Decimal	Amount without VAT
roundAmount	🟩	░	░	Decimal	Rounding
vatAmount	🟩	░	░	Decimal	VAT amount. IF isCalculateVat = false - mandatory isCalculateVat = true - ignored, VAT is calculated by SmartAccounts
totalAmount	🟩	░	░	Decimal	Total amount incl VAT. If provided for add / edit requestts the total amount will be calculated by SmartAccounts as well and the difference will go to roundAmount. This is used to guarantee there is no difference in the amounts between the systems (because of different rounding rules for example)
outstandingAmount	🟩			Decimal	The outstanding amount of the invoice
isCalculateVat	🟩	░	░	Boolean	Is VAT calculated by SmartAccounts or is it provided in API call. Possible values: **true** = SmartAccounts will calculate VAT **false** = vatAmount is mandatory
reverseVatChargeType	🟩				Reverse charge type. For add and edit requests this is evaluated by the country of the vendor and vat percentages of the rows. Possible values: NATIONAL - for example KMS411 documents UNION - EU reverse charge OTHER - reverse charge outside the EU
dateCreated	🟩			Date	
dateUpdated	🟩			Date	
createdBy	🟩			String	
updatedBy	🟩			String	

accountUnpaid	🟩			String	The general ledger account for the invoice (for add / edit this is taken from the vendor's data)
objectId	🟩	▨	▨	String	Object
paymentMethod	🟩	▨	▨	String	If the invoice is marked as paid along with the invoice creation this field contains the reference to payment method used
paymentAmount	🟩	▨	▨	Decimal	Payment amount when paymentMethod is used
warehouseId	🟩	▨	▨	String	
comment	🟩	▨	▨	String	Internal information
documents	🟩			JSONArr	JSONArray of uploaded documents
fileName	🟩			String	
description	🟩			String	
rows	🟩	🟦	🟦		JSONArray of invoice rows
code	🟩	🟦	🟦	String	Article code
description	🟩	🟦	🟦	String	
price	🟩	🟦	🟦	Decimal	
quantity	🟩	🟦	🟦	Decimal	
unit	🟩	▨	▨	String	
vat	🟩			Decimal	VAT
vatPc	🟩	🟦	🟦	String	VAT percentage
sum	🟩			Decimal	Row total amount with VAT
order	🟩	▨	▨	Integer	Order
objectId	🟩	▨	▨	String	Row object
additionalExpenses	🟩	▨	▨	Decimal	Additional warehouse expenses which are calculated into warehouse FIFO price
accountWarehouse	🟩	▨	▨	String	General ledger warehouse account
accountPurchase	🟩	▨	▨	String	General ledger purchase account
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
invoiceId					Invoice id
vendorId					The id of the vendor
number					Acconting number for the invoice
invoiceNumber					Invoice number
referenceNumber					
amount					Amount without VAT

vatAmount					VAT
totalAmount					Amount with VAT
roundAmount					Rounding amount (see description of totalAmount above)
dueDate					
interest					
warehouseId					

DELETE: inout parameters					
id					Unique id for this object

Additional methods					
/vendorinvoices:getpdf					Render the invoice PDF (not the initial invoice PDF but SmartAccounts representation of the document). Input parameters **id** - invoice id

Purchase / Sales – Payments

SERVICE TYPE: Data with change history

URL: /api/purchasesales/payments:<methodname>

Available methods
get
add
edit
delete

GET: possible input parameters					
dateFrom					Filter objects by date
dateTo					Filter objects by date
dateType					Sets which dates are used in filtering. Possible values: **empty** - same as "date" **date** -payment date **modifydate** - added / modified date
partnerType					Filter payments by vendor type. Possible values are CLIENT, VENDOR
accountType					Filter payments by account type. Possible values are BANK, CASH
accountName					Filter payments by account name
partnerId					Filter by client / vendor id
partnerName					Filter by client / vendor name
id					Search for a specific payment
pageNumber					Page number
fetchRows					Return the payment rows as well
fetchComments					Return internal information comments

GET / ADD / EDIT - data fields					
	get	**add**	**edit**	**datatype**	**description**
id	🟩		🟦	String	Unique identifier
date	🟩	🟦	🟦	Date	Date
number	🟩			String	Number in accounting
document	🟩	▨	▨	String	Additional data: document number
orderNumber	🟩			Integer	Cash order number
partnerType	🟩	🟦	🟦		Possible values: CLIENT, VENDOR
clientId / vendorId	🟩	🟦	🟦	String	Unique identifer for the vendor
client / vendor	🟩			JSONObj	JSONObject with data for partner data
id	🟩			String	
name	🟩			String	

regCode	🟩			String	
vatNumber	🟩			String	
accountType	🟩	🟦	🟦	String	Possible values: BANK, CASH
accountName	🟩	🟦	🟦	String	The name of the bank or cash account used
currency	🟩	🟦	🟦	String	Currency (EUR for example)
exchangeRate	🟩	▨	▨	Decimal	Exchange rate (can be null or 1.00 for EUR). Mandatory for those currencies for which no ECB exchange rate can be found (Bitcoin for example).
amount	🟩			Decimal	Payment amount
comment	🟩	▨	▨	String	Internal information
rows	🟩	▨	▨	JSONArr	JSONArray of payment rows
description	🟩			String	The description of the payment rows
amount	🟩	🟦	🟦	Decimal	Payment amount
type	🟩	🟦	🟦	String	Possible values: CLIENT_INVOICE, VENDOR_INVOICE, VAT_RETURN, PREPAYMENT_PAYMENT, SALARY
id	🟩	🟦	🟦	Decimal	Id of the object for this row (for example an invoice)
prepayment	🟩	▨	▨	JSONObj	If the client made a prepayment or a prepayment was made to a vendor
amount	🟩	🟦	🟦	Decimal	
description	🟩	🟦	🟦	String	
account	🟩			String	The general ledger account
extras	🟩	▨	▨	JSONArr	Extra rows for the payment
price	🟩	🟦	🟦	Decimal	Price of the extra
quantity	🟩	🟦	🟦	Decimal	Quantity of the extra
description	🟩	🟦	🟦	String	Description for the extra
amount	🟩			Decimal	Price x quantity
account	🟩	🟦	🟦	String	General ledger account
objectId	🟩	▨	▨	String	
LEGEND		🟦			**mandatory**
		▨			**optional**

ADD / EDIT: return value					
paymentId					Payment id
number					Acconting number for the payment
amount					

DELETE: input parameters

id					Unique id for this object

Additional methods