

The Merit Aktiva API allows 3rd party developers to expand and build on the Merit Aktiva platform.

The Merit Aktiva API is a RESTful API that is used to access Merit Aktiva companies using HTTP and JSON. The API makes it easy to create web, mobile and desktop applications that integrate with your account.

Some potential application ideas:

- Manage customer and vendor databases.
- Manage sales-side and purchase-side transactions.
- Data analysis
- Custom integrations for data import and export

Version 1.6

2018.01

Table of Contents

Overview	3
Successful responses	4
Unsuccessful responses	4
Authentication	5
Creating Sales Invoices	6
Getting a list of Invoices	9
Getting Invoice Details	10
Creating Recurring Invoices	12
Creating Purchase Invoices	13
Getting a list of Purchase Invoices	17
Getting Purchase Invoice Details	18
Creating Purchase Invoices waiting for Approval	20
Creating Payments	24
Creating General Ledger Transactions	25
Getting a list of GL Transactions	27
Getting GL Transaction Details	28
Taxes list	30
Customers list	31
Vendors list	33
Accounts list	35
Projects list	37
Cost Centers list	38
Updating Customer data	39
Updating Vendor data	40
Items list	41
Updating Item data	43
Example in NodeJS	44

Overview

Service URL

The Merit Aktiva API has a single point of entry. The URL depends which localization the company is using:

For Estonian localization: <https://aktiva.merit.ee/api/v1/>

For Finnish localization: <https://aktiva.meritaktiva.fi/api/v1/>

For Polish localization: <https://program.360ksiegowosc.pl/api/v1/>

Please note the use of `https://` in the URL above. *All Merit Aktiva API communication is encrypted over HTTPS.* Any non-secure requests are automatically rejected.

Request Limits

There are no limit on the number of API requests per day. However, requests will be rate-limited if too many calls are made within a short period of time. Additionally, a maximum of 100 results will be returned for growing list methods.

Representation Formats

All Merit Aktiva API requests are composed of JSON and delivered as an HTTP POST request to the endpoint URL.

- All JSON should be UTF-8 encoded.
- Date and time values are of the form `yyyyMMddHHmmss`.
- Dot (.) is used as a decimal separator.
- Booleans are either “true” or “false”, lowercase (not 1 or 0).
- Empty values, represented as null should be lowercase.
- Fields denoting percentages are passed as whole values, not decimals (e.g. 5 for five percent).

Successful responses

Successful response is represented as JSON with response code 200. In many cases, when needed and possible, response will also include details of fulfilled requests (like added document id, transaction id or customer id).

Response to the successful request is usually something like this:

```
{"CustomerId":"665f01a4-357a-4a6b-a565-2f17e6e1da13","InvoiceId":"5f91033c-9d0f-416e-a079-d3c892b8c317","InvoiceNo":"1121","RefNo":"11219"}
```

Unsuccessful responses

In the case of a failure response, the response body will contain the error code and one or more error messages. The error codes can either be 400 or 401.

If the error code is 400, the data sent to the API is not appropriate. Message tag will give you more info about what went wrong.

In case of the error 401, either ApiId or the signature were incorrect. It could also be that someone has meanwhile changed the API ID and Api Key for security reasons thus making the "old" IDs and Keys obsolete.

Authentication

For your company, you'll be given a unique Api ID and Api Key. The admin user can generate and reset API ID and Key on the My Account page (Program Settings).

Keep your Api ID and Api Key secret. They should be guarded just as you would your regular account password. If you feel your ID and Key has been compromised, you can reset it by clicking the "Generate" button on the "My Account" page.

For every API request you make, you'll need to include ApiID, timestamp and signature as request parameters.

Signature is in Base64 format and is calculated using your ApiKey with HMAC-SHA-256, based on the ApiID+timestamp+RequestJSON.

Creating Sales Invoices

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendinvoice>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendinvoice>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendinvoice>

Elements for Adding Invoices:

Customer Object		
Field Name	Type	Comment
Id	Guid	If filled and customer is found in the database then following fields are not important. If not found, the customer is added using the following fields.
Name	Str 150	Required when customer is added
RegNo	Str 30	
NotTDCustomer	Bool	Required when customer is added. True for physical persons and foreign companies. Allowed "true" or "false" (lowercase).
VatRegNo	Str 30	
CurrencyCode	Str 4	
PaymentDeadLine	Int	If missing then taken from default settings.
OverDueCharge	Decimal 5.2	If missing then taken from default settings.
Address	Str 100	
City	Str 100	
Country	Str 100	
PostalCode	Str 15	
CountryCode	Str 2	Required when adding
PhoneNo	Str 50	
PhoneNo2	Str 50	
HomePage	Str 80	
Email	Str 80	

Article (Item) Object

Field Name	Type	Comment
Code	Str 20	Required
Description	Str 100	Required
Type	Int	1 = stock item, 2 = service, 3 = item. Required.
UOMName	Str 64	Name for the unit

VAT Object		
Field Name	Type	Comment
TaxId	Guid	Required. Use gettaxes endpoint to detect the guid needed
Amount	Decimal 18.2	

Payment Object		
Field Name	Type	Comment
PaymentMethod	Str 150	Name of the payment method. Must be found in the company database.
PaidAmount	Decimal 18.2	
PaymDate	Date	

InvoiceRow Object		
Field Name	Type	Comment
Item	Article Object	<p>Sometimes the volume of transactions in the sales software is very high and there is no need to duplicate all the data in accounting. In those cases, you could consider using the same item code for the items with the same VAT rate.</p> <p>Items with different VAT rates must have a different item codes.</p> <p>Also, for the VAT declarations to work, goods and services may not be summed up and must have different item codes.</p>
Quantity	Decimal 18.3	

Price	Decimal 18.7	
DiscountPct	Decimal 18.2	
DiscountAmount	Decimal 18.2	
TaxId	Guid	Required. Use gettaxes endpoint to detect the guid needed
LocationCode	Str 20	Used for stock items and multiple stocks. If used then must be found in the company database.
DepartmentCode	Str 20	If used then must be found in the company database.

Invoice Object		
Field Name	Type	Comment
Customer	Customer Object	
DocDate	Date	
DueDate	Date	
InvoiceNo	Str 35	Required
RefNo	Str 36	
CurrencyCode	Str 4	
DepartmentCode	Str 20	If used then must be found in the company database.
ProjectCode	Str 20	If used then must be found in the company database.
InvoiceRow	Array of InvoiceRow objects	
TaxAmount	Array of VAT objects	Required
RoundingAmount	Decimal 18.2	
TotalAmount	Decimal 18.2	Amount without VAT
Payment	Payment object	
Hcomment	String 4K	Comment before invoice rows
Fcomment	String 4K	Comment at the invoice footer

Getting a list of Invoices

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getinvoices>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getinvoices>

Poland: <https://program.360ksiegowosc.pl/api/v1/getinvoices>

Elements for getting Invoices list:

GetInvoices Object		
Field Name	Type	Comment
PeriodStart	Date	In format yyyyymmdd
PeriodEnd	Date	In format yyyyymmdd

The result, if not empty:

- SIHId
- DepartmentName
- ProjectCode
- ProjectName
- BatchInfo
- InvoiceNo
- DocumentDate
- TransactionDate
- CustomerName
- HComment
- FComment
- DueDate
- CurrencyCode
- CurrencyRate
- TaxAmount
- RoundingAmount
- TotalAmount
- ProfitAmount
- TotalSum
- UserName
- ReferenceNo
- PriceInclVat
- VatRegNo
- PaidAmount

Getting Invoice Details

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getinvoice>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getinvoice>

Poland: <https://program.360ksiegowosc.pl/api/v1/getinvoice>

Elements for the Invoice Details request:

GetInvoice Object		
Field Name	Type	Comment
Id	Guid	

The result, if not empty:

Response		
Field Name	Type	Comment
Header	SalesHeader object	
Lines	Array of Line objects	
Payments	Array of Payment objects	

SalesHeader object	
Field Name	Comment
SIHId	
DepartmentName	
ProjectCode	
ProjectName	
BatchInfo	GL transaction code and #
InvoiceNo	
DocumentDate	
TransactionDate	
CustomerName	
HComment	
FComment	
DueDate	

CurrencyCode	
CurrencyRate	
TaxAmount	VAT amount
RoundingAmount	
TotalAmount	Amount wo. VAT
ProfitAmount	Margin amount
TotalSum	Total amount with taxes and rounding
UserName	
ReferenceNo	
PriceInclVat	
VatRegNo	

Line object	
Field Name	Comment
ArticleCode	
LocationCode	
Quantity	
Price	
TaxName	
TaxPct	
AmountExclVat	
AmountInclVat	
VatAmount	
AccountCode	
DepartmentName	
ItemCostAmount	
ProfitAmount	
DiscountPct	
DicountAmount	
Description	
UOMName	
FixAsset	

Payment object	
Field Name	Comment
PaymDate	
Amount	

Creating Recurring Invoices

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendcontract>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendcontract>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendcontract>

Elements for Adding Recurring Invoices:

Customer Object		
Field Name	Type	Comment
Name	Str 150	
RegNo	Str 30	
Address	Str 100	
City	Str 100	
PostalCode	Str 15	
PhoneNo	Str 50	
Email	Str 80	Invoice email
UserEmail	Str 80	User account email
Contact	Str 50	

Contract Object		
Field Name	Type	Comment
Period	Str	“Aasta” for yearly, “Kuu” for monthly
Item	Str	Item code in the database
DiscPct	Decimal 18.2	
UserCount	Int	
PaymentMethod	Str	“ARVEGA” if paid by invoice, otherwise the IBAN
OrderDate	Date	
Customer	Customer	

Creating Purchase Invoices

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendpurchinvoice>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendpurchinvoice>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendpurchinvoice>

Elements for Adding Purchase Invoices:

Vendor Object		
Field Name	Type	Comment
Id	Guid	If filled and vendor is found in the database then following fields are not important. If not found, the vendor is added using the following fields.
Name	Str 150	Required when vendor is added
RegNo	Str 30	
VatAccountable	Bool	Required when vendor is added. Allowed "true" or "false" (lowercase).
VatRegNo	Str 30	
CurrencyCode	Str 4	
PaymentDeadLine	Int	If missing then taken from default settings.
OverDueCharge	Decimal 5.2	If missing then taken from default settings.
Address	Str 100	
City	Str 100	
Country	Str 100	
PostalCode	Str 15	
CountryCode	Str 2	Required when adding
PhoneNo	Str 50	
PhoneNo2	Str 50	
HomePage	Str 80	
Email	Str 80	

Article (Item) Object		
Field Name	Type	Comment
Code	Str 20	Required. Must be found in company database.
Description	Str 100	Required
Type	Int	1 = stock item, 2 = service, 3 = item. Required.
UOMName	Str 64	Name for the unit

VAT Object		
Field Name	Type	Comment
TaxId	Str 16	Required. Use gettaxes endpoint to detect the guid needed
Amount	Decimal 18.2	

Payment Object		
Field Name	Type	Comment
PaymentMethod	Str 150	Name of the payment method. Must be found in the company database.
PaidAmount	Decimal 18.2	
PaymDate	Date	

InvoiceRow Object		
Field Name	Type	Comment
Item	Article Object	
Quantity	Decimal 18.3	
Price	Decimal 18.7	
TaxId	Str 16	Required. Use gettaxes endpoint to detect the guid needed

LocationCode	Str 20	Used for stock items and multiple stocks. If used then must be found in the company database.
DepartmentCode	Str 20	If used then must be found in the company database.
GLAccountCode	Str 10	General ledger Account code. If used then must be found in the company database
ProjectCode	Str 20	If used then must be found in the company database
CostCenterCode	Str 20	If used then must be found in the company database

Attachment Object

Field Name	Type	Comment
FileName	Str	
FileContent	String	In Base64 format

Invoice Object

Field Name	Type	Comment
Vendor	Vendor Object	
ExpenseClaim	bool	If false, it indicates a normal purchase invoice. If true, the purchase invoice is handled as presented by responsible employee for expense claim.
DocDate	Date	
DueDate	Date	
BillNo	Str 35	
RefNo	Str 36	
CurrencyCode	Str 4	
DepartmentCode	Str 20	If used then must be found in the company database.

ProjectCode	Str 20	If used then must be found in the company database.
InvoiceRow	Array of InvoiceRow objects	
TaxAmount	Array of VAT objects	Required
RoundingAmount	Decimal 18.2	
TotalAmount	Decimal 18.2	Amount without VAT
Payment	Payment object	
Hcomment	String 4K	Comment before invoice rows
Fcomment	String 4K	Comment at the invoice footer
Attachment	Attachment object	

Getting a list of Purchase Invoices

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getpurchorders>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getpurchorders>

Poland: <https://program.360ksiegowosc.pl/api/v1/getpurchorders>

Elements for getting Purchase Invoices list:

GetPurchOrders Object		
Field Name	Type	Comment
PeriodStart	Date	In format yyyyymmdd
PeriodEnd	Date	In format yyyyymmdd

The result, if not empty:

- PIHId
- DepartmentName
- ProjectCode
- BatchInfo
- BillNo
- DocumentDate
- TransactionDate
- VendorName
- DueDate
- Fine
- CurrencyCode
- CurrencyRate
- TaxAmount
- RoundingAmount
- TotalAmount
- ProfitAmount
- TotalSum
- ReferenceNo
- PriceInclVat
- VatRegNo
- PaidAmount

Getting Purchase Invoice Details

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getpurchorder>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getpurchorder>

Poland: <https://program.360ksiegowosc.pl/api/v1/getpurchorder>

Elements for the Purchase Invoice Details request:

GetPurchOrder Object		
Field Name	Type	Comment
Id	Guid	

The result, if not empty:

Response		
Field Name	Type	Comment
Header	PurchHeader object	
Lines	Array of Line objects	
Payments	Array of Payment objects	

PurchHeader object	
Field Name	Comment
PIHId	
DepartmentName	
ProjectCode	
BatchInfo	GL transaction code and #
BillNo	
DocumentDate	
TransactionDate	
VendorName	
DueDate	
Fine	
CurrencyCode	
CurrencyRate	

TaxAmount	VAT amount
RoundingAmount	
TotalAmount	Amount wo. VAT
ProfitAmount	Margin amount
TotalSum	Total amount
ReferenceNo	
PriceInclVat	
VatRegNo	

Line object	
Field Name	Comment
ArticleCode	
LocationCode	
Quantity	
Price	
TaxName	
TaxPct	
AmountExclVat	
AmountInclVat	
VatAmount	
AccountCode	
DepartmentName	
ItemCostAmount	
ProfitAmount	
Description	
UOMName	
FixAsset	

Payment object	
Field Name	Comment
PaymDate	
Amount	

Creating Purchase Invoices waiting for Approval

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendpurchorder>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendpurchorder>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendpurchorder>

Elements for Adding Purchase invoices that need approval:

Vendor Object		
Field Name	Type	Comment
Id	Guid	If filled and vendor is found in the database then following fields are not important. If not found, the vendor is added using the following fields.
Name	Str 150	Required when vendor is added
RegNo	Str 30	
VatAccountable	Bool	Required when vendor is added. Allowed "true" or "false" (lowercase).
VatRegNo	Str 30	
CurrencyCode	Str 4	
PaymentDeadLine	Int	If missing then taken from default settings.
OverDueCharge	Decimal 5.2	If missing then taken from default settings.
Address	Str 100	
City	Str 100	
Country	Str 100	
PostalCode	Str 15	
CountryCode	Str 2	Required when adding
PhoneNo	Str 50	
PhoneNo2	Str 50	
HomePage	Str 80	
Email	Str 80	

Article (Item) Object		
Field Name	Type	Comment
Code	Str 20	If found in the company database, that item will be used. If code is empty, a new item will be created using the first 20 characters of the Description field. If such item exists, that item will be used instead creating a new duplicate.
Description	Str 100	Required.
Type	Int	1 = stock item, 2 = service, 3 = item. Required.
UOMName	Str 64	Name for the unit

VAT Object		
Field Name	Type	Comment
TaxId	Str 16	Required. Use gettaxes endpoint to detect the guid needed
Amount	Decimal 18.2	

Payment Object		
Field Name	Type	Comment
PaymentMethod	Str 150	Name of the payment method. Must be found in the company database.
PaidAmount	Decimal 18.2	
PaymDate	Date	

InvoiceRow Object		
Field Name	Type	Comment
Item	Article Object	
Quantity	Decimal 18.3	

Price	Decimal 18.7	
TaxId	Str 16	Required. Use gettaxes endpoint to detect the guid needed
LocationCode	Str 20	Used for stock items and multiple stocks. If used then must be found in the company database.
DepartmentCode	Str 20	If used then must be found in the company database.
GLAccountCode	Str 10	General ledger Account code. If used then must be found in the company database
ProjectCode	Str 20	If used then must be found in the company database
CostCenterCode	Str 20	If used then must be found in the company database
Payment	Payment object	

Attachment Object

Field Name	Type	Comment
FileName	Str	
FileContent	String	In Base64 format

Invoice Object

Field Name	Type	Comment
Vendor	Vendor Object	
ExpenseClaim	bool	If false, it indicates a normal purchase invoice. If true, the purchase invoice is handled as presented by responsible employee for expense claim.
DocDate	Date	
DueDate	Date	
BillNo	Str 35	
RefNo	Str 36	

CurrencyCode	Str 4	
DepartmentCode	Str 20	If used then must be found in the company database.
ProjectCode	Str 20	If used then must be found in the company database.
InvoiceRow	Array of InvoiceRow objects	
TaxAmount	Array of VAT objects	Required
RoundingAmount	Decimal 18.2	
TotalAmount	Decimal 18.2	Amount without VAT
Hcomment	String 4K	Comment before invoice rows
Fcomment	String 4K	Comment at the invoice footer

Creating Payments

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendpayment>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendpayment>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendpayment>

Elements for Adding a Payment:

Payment Object		
Field Name	Type	Comment
IBAN	Str	Must match with one of the payment methods in the database
CustomerName	Str 150	
InvoiceNo	Str 35	
RefNo	Str 36	
Amount	Decimal 18.2	

Creating General Ledger Transactions

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/sendglbatch>

Finland: <https://aktiva.meritaktiva.fi/api/v1/sendglbatch>

Poland: <https://program.360ksiegowosc.pl/api/v1/sendglbatch>

PLEASE NOTE: This endpoint is designed to import raw general ledger transactions. It is not suitable for transactions involved debts (except when Tax Accounts are used). To take up debts please import the data using API endpoints for Invoices and Purchase Invoices.

Elements for General Ledger Transactions:

EntryRow Object		
Field Name	Type	Comment
AccountCode	Str 8	Must be found in the company database
DepartmentCode	Str 16	Must be found in the company database
Debit	Decimal 18.2	
Credit	Decimal 18.2	

GLBatch Object		
Field Name	Type	Comment
DocNo	Str 35	
BatchDate	Date	
EntryRow	Array of EntryRow objects	

Attachment Object		
Field Name	Type	Comment

FileName	Str	
FileContent	String	In Base64 format

Getting a list of GL Transactions

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getglbatches>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getglbatches>

Poland: <https://program.360ksiegowosc.pl/api/v1/getglbatches>

Elements for getting GL Transactions list:

GetPurchOrders Object		
Field Name	Type	Comment
PeriodStart	Date	In format yyyyymmdd
PeriodEnd	Date	In format yyyyymmdd

The result, if not empty:

- GLBId
- BatchCode
- No
- Document
- BatchDate
- CurrencyCode
- CurrencyRate
- TotalAmount
- PriceInclVat

Getting GL Transaction Details

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getpurchorder>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getpurchorder>

Poland: <https://program.360ksiegowosc.pl/api/v1/getpurchorder>

Elements for the Purchase Invoice Details request:

GetPurchOrder Object		
Field Name	Type	Comment
Id	Guid	

The result, if not empty:

Response		
Field Name	Type	Comment
Header	GLBHeader object	
Lines	Array of GLEntry objects	

GLBHeader object	
Field Name	Comment
GLBId	
BatchCode	
No	
Document	
BatchDate	
CurrencyCode	
CurrencyRate	
TotalAmount	
PriceInclVat	

GLEntry object	
Field Name	Comment
AccountCode	

Memo	
DepartmentCode	
TaxName	
DebitAmount	
DebitCurrency	
CreditAmount	
CreditCurrency	

Taxes list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/gettaxes>

Finland: <https://aktiva.meritaktiva.fi/api/v1/gettaxes>

Poland: <https://program.360ksiegowosc.pl/api/v1/gettaxes>

Elements for getting Taxes list:

Taxes Object		
Field Name	Type	Comment
Id	Guid	
Code	Str 16	
Name	Str 40	

Customers list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getcustomers>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getcustomers>

Poland: <https://program.360ksiegowosc.pl/api/v1/getcustomers>

Elements for getting Customers list:

GetCustomers Object		
Field Name	Type	Comment
Id	Guid	If filled, next fields will be ignored
RegNo	Str	Exact match.
VatRegNo	Str	Exact match.
Name	Str	Broad match.

The result for this request can be:

- 1) empty result
- 2) a certain customer (if exact match is used and/or only one customer is found)
- 3) a list of customers matching the broad match term.

The result, if not empty, is a Customer object with following fields:

- CustomerId
- Name
- RegNo
- Contact
- PhoneNo
- Address
- City
- Email
- CurrencyCode
- PostalCode
- CountryName
- County
- PhoneNo2
- FaxNo
- HomePage

- PaymentDeadLine
- OverdueCharge
- CustomerGroupName
- VatRegNo
- NotTDCustomer
- BankName
- SalesInvLang
- RefNoBase

Vendors list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getvendors>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getvendors>

Poland: <https://program.360ksiegowosc.pl/api/v1/getvendors>

Elements for getting Vendors list:

GetVendors Object		
Field Name	Type	Comment
Id	Guid	If filled, next fields will be ignored
RegNo	Str	Exact match
VatRegNo	Str	Exact match
Name	Str	Broad match

The result for this request can be:

- 4) empty result
- 5) a certain vendor (if exact match is used and/or only one vendor is found)
- 6) a list of vendors matching the broad match term.

The result, if not empty, is a Customer object with following fields:

- VendorId
- Name
- RegNo
- Contact
- PhoneNo
- Email
- CurrencyCode
- PaymentDeadLine
- BankAccount
- HomePage
- ReferenceNo
- Address
- City
- County
- PostalCode

- CountryName
- PhoneNo2
- FaxNo
- OverdueCharge
- VendorGroupName
- VatRegNo
- VatAccountable

Accounts list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getaccounts>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getaccounts>

Poland: <https://program.360ksiegowosc.pl/api/v1/getaccounts>

Elements for getting Accounts list:

GetAccounts Object		
Field Name	Type	Comment
AccountId	Guid	
NoActive	Bool	
Code	Str	
Name	Str	
TaxName	Str	
LinkedVendorName	Str	Filled for tax accounts
IsParent	Int	1 - details, 2 - summary
UsageFilter	Int	0 or missing will output all accounts. 1 for cost accounts, 2 for cost contra-accounts, 3 for purchase VAT accounts

The result for this request can be:

- 1) empty result
- 2) a certain account (if exact match is used and/or only one account is found)
- 3) a list of accounts matching the broad match term.

The result, if not empty, is a Customer object with following fields:

- AccountID
- NoActive
- Code
- Name
- TaxName
- LinkedVendorName

- IsParent

Projects list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getprojects>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getprojects>

Poland: <https://program.360ksiegowosc.pl/api/v1/getprojects>

Result for this request:

GetAccounts Object		
Field Name	Type	Comment
Code	Str 20	
Name	Str 64	
EndDate	date	

Cost Centers list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getcostcenters>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getcostcenters>

Poland: <https://program.360ksiegowosc.pl/api/v1/getcostcenters>

Result for this request:

GetAccounts Object		
Field Name	Type	Comment
Code	Str 20	
Name	Str 64	
EndDate	date	

Updating Customer data

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/updatecustomer>

Finland: <https://aktiva.meritaktiva.fi/api/v1/updatecustomer>

Poland: <https://program.360ksiegowosc.pl/api/v1/updatecustomer>

Elements for updating the customers:

UpdateCustomer Object		
Field Name	Type	Comment
Id	Guid	Required
Name	Str 150	
Address	Str 100	
City	Str 100	
PostalCode	Str 15	
PhoneNo	Str 50	
PhoneNo2	Str 50	
Email	Str 80	
RegNo	Str 30	
VatRegNo	Str 30	

Updating Vendor data

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/updatevendor>

Finland: <https://aktiva.meritaktiva.fi/api/v1/updatevendor>

Poland: <https://program.360ksiegowosc.pl/api/v1/updatevendor>

Elements for updating the vendor:

UpdateVendor Object		
Field Name	Type	Comment
Id	Guid	Required
Name	Str 150	
Address	Str 100	
City	Str 100	
PostalCode	Str 15	
PhoneNo	Str 50	
PhoneNo2	Str 50	
Email	Str 80	
RegNo	Str 30	
VatRegNo	Str 30	

Items list

Endpoint URL (GET):

Estonia: <https://aktiva.merit.ee/api/v1/getitems>

Finland: <https://aktiva.meritaktiva.fi/api/v1/getitems>

Poland: <https://program.360ksiegowosc.pl/api/v1/getitems>

Elements for getting Items list:

GetItems Object		
Field Name	Type	Comment
Id	Guid	If filled, next fields will be ignored
Code	Str	Exact match
Description	Str	Broad match

The result for this request can be:

- 1) empty result
- 2) a certain item (if exact match is used and/or only one item is found)
- 3) a list of items matching the broad match term.

The result, if not empty, is the following object:

- ItemId
- Code
- Name
- UnitofMeasureName
- Type
- SalesPrice
- InventoryQty
- VatTaxName
- Usage
- SalesAccountCode
- PurchaseAccountCode
- InventoryAccountCode
- ItemCostAccountCode
- DiscountPct
- LastPurchasePrice
- ItemUnitCost
- InventoryCost
- ItemGroupName

- DefLoc_Name

Updating Item data

Endpoint URL (POST):

Estonia: <https://aktiva.merit.ee/api/v1/updatevendor>

Finland: <https://aktiva.meritaktiva.fi/api/v1/updatevendor>

Poland: <https://program.360ksiegowosc.pl/api/v1/updatevendor>

Elements for updating the Item:

UpdateVendor Object		
Field Name	Type	Comment
Id	Guid	Required
Code	Str 20	
Description	Str 100	
SalesPrice	Decimal 18.2	

Example in NodeJS

The simple example below will show you how to use API and post the invoice:

```
var CryptoJS = require('crypto-js');
var request = require('request');

var reqJson = {
  Customer: {
    //Id: null,
    Name: 'FirstCustomer Inc',
    RegNo: '1122334455',
    NotTDCustomer: false,
    VatRegNo: '11223344',
    CurrencyCode: 'EUR',
    PaymentDeadLine: 7,
    OverDueCharge: 0,
    RefNoBase: 1,
    Address: 'Merimiehenkatu 31',
    CountryCode: 'FI',
    County: 'Finland',
    City: 'Helsinki',
    PostalCode: '',
    PhoneNo: '6548765',
    PhoneNo2: '',
    HomePage: '',
    Email: 'customermail@gmail.com',
  },
  DocDate: '20170113131239',
  DueDate: '20170125131239',
  InvoiceNo: '123',
  RefNo: '222',
  DepartmentCode: '',
  ProjectCode: '',
  InvoiceRow: [
    {
      Item:
        {
          Code: 1234567,
          Description: 'Bag of goldflakes',
          Type: 3,
          UOMName: 'kg',
        },
      Quantity: '2',
      Price: '1000',
      DiscountPct: 0,
      DiscountAmount: 0,
      TaxId: '665f01a4-357a-4a6b-a565-2f17e6e1da13',
      LocationCode: 1,
    },
  ],
  TotalAmount: 2000,
  RoundingAmount: 0,
  TaxAmount: [{ TaxId: '665f01a4-357a-4a6b-a565-2f17e6e1da13', Amount: 400 }],
  HComment: '',
  FComment: ''
}
```

```

};

var ApiId = 'Your-api-id';
var ApiKey = 'Your-api-key';

function pad2(n) {
  return n > 9 ? '' + n : '0' + n;
}

function getTimestamp() {
  var d = new Date();
  var yyyy = d.getFullYear();
  var MM = pad2(d.getMonth() + 1);
  var dd = pad2(d.getDate());
  var HH = pad2(d.getHours());
  var mm = pad2(d.getMinutes());
  var ss = pad2(d.getSeconds());
  return yyyy + MM + dd + HH + mm + ss;
}

var timestamp = getTimestamp();
var dataString = ApiId + timestamp + JSON.stringify(reqJson);
var hash = CryptoJS.HmacSHA256(dataString, ApiKey);
var signature = CryptoJS.enc.Base64.stringify(hash);

console.log(dataString);
console.log(hash);
console.log(signature);

var url = 'https://aktiva.merit.ee/api/v1/sendinvoice' +
  '?ApiId=' + ApiId + '&timestamp=' + timestamp + '&signature=' +
  signature;

request({
  url: url,
  method: 'POST',
  json: true,
  headers: { 'content-type': 'application/json', },
  body: reqJson,
},
function (request, response) {
  console.log('Status code: ', response.statusCode, ' -- ',
response.statusMessage);
  console.log('Headers: ', response.headers['content-type']);
  console.log('Body: ', response.body);
}
);

```