

# REST API v2

The second version of the Standard ERP Rest API is used to read data from the Standard ERP database via HTTP requests.

## Compared to the first version

- Read values from any register or block in the database including customised registers and blocks.
- Expanded options to filter and sort records
- Retrieve changes since a previous API call
- Improved security features
- Fetch only wanted fields, making the response smaller

## Security

The API uses the access rights for the user. The user must be able to navigate to and open the register in a normal client and must thus have access to both a module the register is in, and the register itself. For registers that do not appear in any module a customisation to make them appear in some module is needed.

Examples of such registers are RHistVc, MailVc,

The user should authenticate with OAuth

The path "api" may be secured with the setting "access to functions on the web".

- The default access for HAL functions is off
- The default access for the "api" path is login required
- For all other resources it is public.

If the user has been restricted in viewing some fields on a register, for example with the "view cost price" access, he can still retrieve this field via the API and should thus not have access to it. This is the same as for exports.

## Setup

- In the system module, Optional Features setting, the checkbox Web Rest API is used to enable the API.
- The user must be given the access right to the action Rest API

## Basic Use

Requests specify the company and resource to read, in the most basic format:

<http://hostname.domain.top/api/1/IVVc>

where:

- "api" is a mandatory hardcoded string
- "1" is the company code from the companies setting (1st column)
- "IVVc" is the name of the register

This would fetch all invoices in company 1.

To retrieve information about base currencies, you similarly use:

<http://hostname/api/1/BaseCurBlock>

## Data format

The data format for request and returned data is the same and hardcoded.

- The decimal point is "." (period)
- There is no thousand separator
- Dates are in ISO format YYYY-MM-DD

## Parameters

The actual parameter values used for results such as key and range used, version of the server etc. are returned as attributes of the data tag.

## Sort

The sort parameter will sort the retrieved records on the specified field. The name of the index that was used will be returned in the result. Only one field can be sorted on, and only if there is a suitable index, if there is no suitable index the request will fail. The field name is case sensitive.

Example:

<http://hostname.domain.top/api/1/IVcs?sort=CustCode>

## Range

Requires the use of the sort parameter.

Retrieve only records where the sorted-on field is inside the specified range. The range is inclusive (values matching the start and end values are inside the range). The first and last value of the range are separated with the ":" (colon character). Open ranges where only the 1st or last value is specified are allowed, and will return all records before or after the specified value. If only a singular value is specified (no colon) only records matching that value will be retrieved.

Examples:

<http://hostname.domain.top/api/1/IVc?sort=CustCode&range=10101:10104>

Will return invoices with customers from 10101 to 10104

<http://hostname.domain.top/api/1/IVc?sort=CustCode&range=10104>

Will return invoices with customers from 10104 until the last customer

<http://hostname.domain.top/api/1/IVc?sort=CustCode&range=10104>

Will return invoices only for customer 10104

The range parameter is fast to use because it uses an index.

## Fields

The fields parameter specifies which fields are to be retrieved. The fields are specified comma separated. If the parameter is not present all fields are retrieved. If a field in the matrix and a field in the header has the same name, both will be retrieved. If no field in the matrix are retrieved then the matrix itself (number of rows etc.) will not be present in the result.

Example:

<http://hostname.domain.top/api/1/IVc?fields=SerNr,OKFlag,Addr0,ArtCode,CustCode,InvDate,TransDate,Objects>

## Filter

The data can be filtered with the filter parameter. it is specified like this:

<http://hostname.domain.top/api/1/IVc?filter.CustCode=10104>

The filter is significantly slower than range, as it will not use an index and scan all records. If you use a range the filter will only scan the records in the range, so try to use the most selective condition possible in the range and all other conditions in filters.

- There can only be one filter per field
- There can be multiple filters on different fields
- Filters can handle ranges of values, with the same syntax as range, including open ranges
- Filters work only on header fields
- Filtering on list fields such as Objects is done by the whole string. a filter.Objects=AB will not match "AB,D10101"

Example:

<http://hostname.domain.top/api/1/IVc?filter.CustCode=10100:10200&Sum4=100:1000>

Will retrieve invoices with a total sum of 100 to 1000 for customers in the range 10100 to 10200.

## Offset and limit

If the result is larger than the api user can handle in one request, the result can be retrieved in smaller pieces.

The offset will skip the specified number of records before producing output and the limit will restrict the number of records retrieved.

Eksempel:

<http://hostname/api/1/IWVc?offset=0&limit=5>

<http://hostname/api/1/IWVc?offset=5&limit=5>

<http://hostname/api/1/IWVc?offset=10&limit=5>

will retrieve the 15 first invoices in 3 separate requests.

offset and limit works together with all other parameters

## Updates\_after

returns all records that were updated after a given sequence number.

the sequence number is returned in each request and can be saved for later use with updates\_after

Example:

[http://hostname/api/1/IWVc?updates\\_after=5000](http://hostname/api/1/IWVc?updates_after=5000)

## Deletes\_after

returns all record that were deleted after a given sequence number.

the sequence number is returned in each request and can be saved for later use with deletes\_after

Example:

[http://hostname/api/1/IWVc?deletes\\_after=5000](http://hostname/api/1/IWVc?deletes_after=5000)