

Webservice REST

These Web services allow you to do HTTP requests to the specified endpoints (VTE_URL/restapi/v1/vtews/METHOD_NAME).

The requests need POST method, basic authentication (you must set 'Authentication basic' into the header which is calculated through a function based on username* and password*) and their relative parameters.

Each one method will give a JSON response with a status and the data or in case of error an error code and its relative message.

Notes:

- The "id" parameter is always specified as 'moduleid* x recordid' (ex: 2x313)
*: You can get the the moduleid from webservice 'describe' method (idPrefix)

Headers:

- Authorization: Basic base64_encode("username*:password*")
- Content-Type : application/json

username* (VTE) password* (User preferences' Webservice Access Key)

Authorization Example:

'Basic ZmVkZXJpY28ucGVybGluOjB5cm5MRjNhS2RhMDZ2c3E='

Requirements for Apache2:

- Activate apache rewrite module
- Set AllowOverride All into the virtualhost file to the vte path folder.

Example:

```
<Directory /var/www/html/VTE_FOLDER>
Options -Indexes +FollowSymLinks +MultiViews
AllowOverride All
Require all granted
</Directory>
```

Requirements for Nginx + PHP FPM

As Nginx does not handle htaccess file, some special rewrites are needed in order to handle WS requests.

Below an example of nginx conf file:

```
server {
    listen 80;

    root /var/www/vtenext/;
    autoindex off;
    index index.php index.html index.htm;

    # Make site accessible from http://localhost/
    server_name localhost;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to index.html
        try_files $uri $uri/ index.html;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    location ~ /\.ht {
        deny all;
    }

    # special folder handling, since .htaccess files are not supported

    # REST API

    location /restapi/v1 {
        rewrite ^(+)/vtews/(.*)$ /restapi/v1/index.php/vtews/$2;
    }

    # PORTAL

    location ~ /portal/v2/(?!public) {
        rewrite ^/portal/v2/(.*)$ /portal/v2/public/$1;
    }

    # STORAGE
```

```

[]
[]location /storage {
[]deny all;
[]
[]location /storage/uploads_emails_ {
[]allow all;
[]}
[]
[]location /storage/images_uploaded/ {
[]allow all;
[]}
[]
[]location ~* /storage/logo/.\.(jpg|jpeg|png|gif)$ {
[]allow all;
[]}
[]
[]rewrite ^/storage/(.*)$ /getStorage.php?file=$1;
[]}
[]
[]# PROTECTED FOLDERS
[]
[]location /logs {
[]deny all;
[]}
[]
[]location /plugins/dataimporter {
[]deny all;
[]}
[]
[]location /cache/sys {
[]deny all;
[]}
[]
[]location /cache/pdfmaker {
[]deny all;
[]}
[]
[]location /cache/import {
[]deny all;
[]}
[]

```

```

[]
[]location /cache/session {
[]deny all;
[]}
[]
[]location /cache/pdf {
[]deny all;
[]}
[]
[]location /dataimport {
[]deny all;
[]}
[]
[]location /modules/VteSync/VteSyncLib/storage {
[]deny all;
[]}
[]
[]location /modules/Messages/src/attachment_tnef/plugins/attachment_tnef/class/ {
[]deny all;
[]}
[]
[]# SMARTOPTIMIZER
[]
[]# disabled by default, enable it if needed
[]#location ~* \.(gif|jpe?g|png|swf|css|js|html?|xml|txt|ico)$ {
[]#[]rewrite ^(.*)$ /smartooptimizer/?$1;
[]#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
[]location ~ [^/]\.php(/|$) {
[]include snippets/fastcgi-php.conf;
[]
[]# With php cgi alone:
[]#fastcgi_pass 127.0.0.1:9000;
[]
[]# With php fpm:
[]fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
[]fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
[]}
}

```

Web Service Methods:

Name	Description	Parameters	Response
create	Create a record of the specified module	elementType (String)element (Encoded)	Return all fields and values of the created record
	Ex: Url: VTE_URL/restapi/v1/vtews/create Parameters: {"elementType":"Accounts", "element":{"\accountname\":"account1", \assigned_user_id\":"19x1"}}		
update	Update the specified record	id (String)columns (Encoded)	Return all fields and values of the updated record
	Ex: Url: VTE_URL/restapi/v1/vtews/update Parameters: {"id":"3x310", "columns":{"\accountname\":"test fede postman"}}		
revise	Update the specified record. It's different only for parameters, the result is the same	element (Encoded)	Return all fields and updated values
	Ex: Url: VTE_URL/restapi/v1/vtews/revise Parameters: {"element":{"\id\":"3x27", \employees\":"5", \industry\":"Banking"}}		
retrieve	Illustrate the fields of the specified record and their relative values	id (String)	Return all fields and values of the specified record
	Ex: Url: VTE_URL/restapi/v1/vtews/retrieve Parameters: {"id":"3x310"}		
retrieveinventory	Illustrate the fields of the specified inventory record, their relative values and the product block's information	id (String)	Return all fields and values of the specified record and product's block information
	Ex: Url: VTE_URL/restapi/v1/vtews/retrieveinventory Parameters: {"id":"16x104"}		
delete	Delete the specified record	id (String)	Return the request status (successful or not)
	Ex: Url: VTE_URL/restapi/v1/vtews/delete Parameters: {"id":"3x306"}		
query	Execute a query and return the result's rows	query (String)	Return all rows of the executed query

Ex: **Url:** VTE_URL/restapi/v1/vtews/query

Parameters:

```
{"query": "SELECT * FROM Accounts WHERE accountname like '%vte%';"}
```

listtypes	Describe each one module which contains uitype of the specified format	fieldTypeList (Encoded)	Return module information of the specified fieldtypes
	Ex: Url: VTE_URL/restapi/v1/vtews/listtypes Parameters: { "fieldTypeList": ["integer", "file"] }		
describe	Describe the specified module and their relative fields	elementType (String)	Return all module information and its fields properties(no hidden fields)
	Ex: Url: VTE_URL/restapi/v1/vtews/describe Parameters: { "elementType": "Accounts" }		
describeall	The describeall method is different from describe one because it shows hidden fields too	ElementType (Encoded)	Return all module information and its fields properties(hidden fields too)
	Ex: Url: VTE_URL/restapi/v1/vtews/describeall Parameters: { "elementType": "Accounts" }		
getlabels	Return all labels and translated ones of the specified module, language	username (String) language (String) module (String)	Return labels and translated ones of the specified module, language
	Ex: Url: VTE_URL/restapi/v1/vtews/getlabels Parameters: { "username": "admin", "language": "it_it", "module": "Accounts" }		
getlangs	Return all languages installed into the CRM		Return all CRM languages
loginpwd	Return user webservice access key if username and password are valid	username (String) password (String)	Return user webservice access key
	Ex: Url: VTE_URL/restapi/v1/vtews/loginpwd Parameters: { "username": "admin", "password": "123456789" }		
getmenulist	Return all modules information (visibility, tabid, name, sequence, ...)		Return modules information and properties
oldoquery	Return records that contain the searched value into specified module's fields	module (String) search_fields (Encoded) search_value (String)	Return some record basic information where searched value is contained into the specified module's fields

Ex: **Url:** VTE_URL/restapi/v1/vtews/oldoquery

Parameters:

```
{"module":"Accounts", "search_fields":["accountname", "website"],  
"search_value":"vtenext"}
```

Technical documentation for SDK is available [here](#).

Revision #1

Created 2026-07-08 12:37:38 UTC by m.maporti

Updated 2026-07-08 12:38:28 UTC by m.maporti