

# 18.10 REST Webservice

## REST Webservice Settings

From the CRM settings its possible to view the available web services (referred here as *webservice*, or *REST webservice*), to allow external applications to interact with the CRM itself, using the [REST paradigm](#). A webservice is accessible by an URL address (referred here as *endpoint*), usually made by the URL address of the vte installation, followed by `/restapi/v1/vtews/`, and the method name associated to the endpoint of interest.

Once configured, it is possible to invoke the desired endpoint with HTTP requests to execute the requested actions.

The screenshot shows the 'Settings > REST Webservices' page in the Vtenext application. The page has a dark blue sidebar on the left with icons for home, settings, users, and other functions. The main content area is white and contains the following elements:

- A search bar at the top with the text 'Search...'.
- A breadcrumb trail: 'Settings > REST Webservices'.
- A section titled 'To connect an application to Vtenext use the following parameters:' with the following fields:
  - 'Endpoint' with the value 'https://my\_vte.url/restapi/v1/vtews/...'.
  - 'User Name' with a dropdown menu showing 'admin (admin)'.
  - 'Webservice access key' with a 'Show' button.
  - 'Modules and fields usable' with a 'Show' button.
  - 'Display REST Documentation' with a 'Show' button.
- A link 'For more info click here'.
- A section titled 'Configure external Web services to be used with processes' with a green button '+ Add Webservice Client'.
- A section titled 'List of configured client webservices' with the text 'No external web services configured, press Add to create a new one'.

At the bottom of the page, there is a footer with the text 'VTENEXT 26.07', 'Server response time: 0.18 seconds.', and '© 2008-2026 vtenext.com | License'.

## HTTP requests format

HTTP requests support the `POST` method only and the `Basic` authentication type, meaning the definition of `Authorization` defined as `Basic <credentials>`, where `<credentials>` is the Base64 encoding of the user name and the accesskey, separated by `:`.

Webservice requests are intended to return a response in JSON format: on success they return a status and the data computed by the endpoint, and on error they return the error code and relative description.

## User selector and access keys, list of modules and fields views

By using the picklist associated to `User Name`, it is possible to select the user, and this allows, with related buttons, to view the accesskeys and the available modules to do HTTP request with the selected user.

The button associated with `Webservice access key` allows to view, after entering of one's own password, the access keys generated by the user: those are used to authorize webservices requests, as noted in the paragraph `HTTP requests format`. Note that to add more access keys, the user needs to enter in their preference page, and go to the section *Webservices, REST Api Configuration*: refer to the [section dedicated in the user preferences page](#).

The button associated with `Modules and fields usable` allows to view the list of the modules available to the user and their internal name, to use in the webservice requests where needed.

## REST Documentation Page (OpenAPI)

By clicking the button associated with `Display REST Documentation`, it is possible to display the page which lists the endpoints exposed by the CRM. Initially empty, click on `Generate OpenAPI Documentation` to get the definition of the endpoints that will be used for their layout.

It is then possible to test single endpoints by clicking the button `Test Request`, located in the example box adjacent to the description of the method, for the desired endpoint and user. The user can be selected with the related selector, located centrally at the top of the page.

Search...

---

Generate OpenAPI Documentation

User Name admin (admin) ▾

Search ⌵ K

**Introduction** >

Base ▾

- processes.process\_related\_to POST
- processes.processes\_related\_to POST
- touch.answer\_invitation POST
- touch.autocomplete POST
- touch.customview.get POST
- touch.customview.list POST
- touch.customview.save POST
- touch.customview.values POST
- touch.change\_password POST
- touch.check\_push\_token POST
- touch.compose\_message POST
- touch.convert\_lead POST
- touch.delete\_messages\_account POST
- touch.delete\_push\_shutdown\_interva POST

Powered by Scalar

v26.07 - Rev: 3503 OAS 3.0.0

## vtenext CRM - REST API

[Download OpenAPI Document](#)

vtenext CRM provides programmatic access via REST endpoints, allowing direct management of modules, records, metadata, and file attachments for integration with external systems and automation of data operations.

### Available operations

- CRUD Operations** - Create, read, update, and delete records across all modules, including standard (Accounts, Contacts, Leads) and custom modules.
- Metadata Access** - Retrieve module schemas, field definitions, data types, relationships, and picklist values.
- File Management** - Upload, download, and manage documents and attachments associated with CRM records.

| Server                    |
|---------------------------|
| https://my_vte.url        |
| Your vtenext CRM instance |

  

| Authentication Required                                       | BasicAuth ▾ |
|---------------------------------------------------------------|-------------|
| Standard HTTP Basic Authentication. Username: your vt... More |             |
| Username: admin                                               | Required ✕  |
| Password: .....                                               | ✕ ⓘ         |

  

| Client Libraries                                                           |
|----------------------------------------------------------------------------|
| <div style="text-align: right; padding-right: 10px;">&gt;_ Shell ...</div> |
| Shell Curl                                                                 |

The endpoints are viewed by scrolling directly on the page, by clicking on the endpoint listed in the sidebar, or by searching with the search bar located at the top of the endpoint list (which can also be opened by pressing `CTRL-K`, or `CMD-K` on macOS). Every single endpoint has then displayed the body of the request, the responses, and optional examples of each.

The screenshot displays the VTENEXT API Explorer web application. At the top, there's a search bar and a user profile icon. Below the header, a navigation sidebar on the left lists various API endpoints, with **touch.get\_currencies** selected and highlighted. The main area shows details for this endpoint:

- Method:** POST
- Description:** Retrieves the list of active or all configured currencies in the system.
- User Name:** admin (admin)
- Body:** Touch API request body structure. It indicates a required object of type `TouchGetCurrenciesRequest`. A "Show Child Attributes" button is present.
- Responses:** A list of expected status codes:
  - > 200 Currencies retrieved successfully
  - > 401 Unauthorized - Invalid or missing authentication
  - > 500 Internal Server Error

On the right side, a terminal window shows a cURL command for testing the endpoint:

```
POST /restapi/v1/vtews/touch.get_cur Shell Curl v
1 curl https://my_vte.url/restapi/v1/vtews/touch
2 --request POST \
3 --header 'Content-Type: application/json' \
4 --header 'Authorization: Basic ..... ' \
5 --data '{
6   "request": {
7     "getall": "false"
8   }
9 }
```

A "Test Request" button is located below the cURL command.

At the bottom right, a JSON response schema is shown for a 200 status code:

```
{
  "status": 200,
  "data": {
    "success": true,
    "error": "",
    "touch_session_id": "xxxxxxxxxxxxxxxxxxxxxx",
    "remote_wipe": false,
    "result": {
      "currencies": {
```

The footer contains the text "VTENEXT 26.07", "Server response time: 0.2 seconds.", and copyright information "© 2008-2026 vtentext.com | License".

# Configuration of external Web services for use in processes

In the settings page is possible to add external web service clients, based on Zapier (refer to the [dedicated chapter](#)), or by custom settings.

Once configured and activated, done by clicking the button `Add Webservice Client` and filling up the required settings, they can be viewed in the `List of configured client webservices` section, and used in processes.

## Additional information

You can see the additional [technical information of the REST Webservice here](#).

---

Revision #12

Created 2026-07-06 15:07:50 UTC by Admin

Updated 2026-07-09 12:13:15 UTC by f.fabris