Realpad API

for real estate development project websites

Created by Matej Vitásek matej@realpadsoftware.com

This document is intended for web developers integrating a real estate development project website with Realpad CRM in two areas:

- **Fetching pricelist data** (units, their prices and status, links to their floorplans, ...).
- **Sending leads** from the contact form (name, surname, email, phone, additional info).

Top-level summary

1. Obtain login credentials from our support team support@realpadsoftware.com. You'll typically receive one pair of credentials per project - one for fetching pricelist, one for sending leads.
    - *Example: for **My New Project** you'd receive*
    - *`my-new-project-`**`pricelist`** / `some_password`*
    - *`my-new-project-`**`leads`** / `some_other_password`*
2. Write **backend** code that will invoke a HTTP POST request to the endpoints mentioned below and process the API response appropriately.
3. Test the whole solution together with our support team on a **test environment of the project website**.
4. After a **3-way confirmation** that all is running correctly deploy the solution to the **production website**.
5. If you have any questions contact us on support@realpadsoftware.com.

Fetching pricelist data

Best practice successfully followed by the hundred+ connected project webs is the following:

- Fetch the pricelist **every hour** and store the result in a relational database for a quick frontend pricelist / unit detail rendering.
- Look at what resource UIDs your **local cache** is missing and fetch them.
- If there is any problem with the API **log** it and **retry** at some later time.

Request

Use **HTTP POST** and the endpoint **`https://cms.realpad.eu/ws/v10/get-project`**
Send the following **required** parameters as a URL-encoded form:

- **`login`** - string login obtained from our support.
- **`password`** - string password obtained from our support.
- **`screenid`** - integer value, typically constant, obtained from our support.
- **`developerid`** - integer value, typically constant, obtained from our support.
- **`projectid`** - integer value identifying the project, will change between projects.

Example call in cURL:

```
curl \
	--data "login=...&password=...&screenid=...&projectid=...&developerid=..." \
	https://cms.realpad.eu/ws/v10/get-project
```

Response

The response consists of the HTTP Status Code and, if the call was successful, XML payload.

Status codes:

- `200 OK` - All was fine, the call succeeded and the payload contains the XML data.
- `400 BAD REQUEST` - One of the IDs was invalid, e.g. unknown project was requested.
- `401 NOT AUTHORIZED` - The credentials are invalid or the login or requesting IP is banned.

Payload: the file is structured using this hierarchy: `export -> project -> building -> floor -> flat`. Typically you will only need to use the data in the `<flat>` elements containing the relevant units (so not only flats but also cellars parking spots etc). The unit plan and PDF UIDs (see below) can be found in `<flat>` XML attributes, the rest of the important data is stored in `<flat-attribute>` sub-elements identified by their *keys*. Explanation follows:

- `flat_area` - Unit area in square meters (sqm).
- `flat_area_balcony` - Balcony area in sqm.[1]
- `flat_area_garden` - Garden area in sqm.
- `flat_area_terrace` - Terrace area in sqm.
- `flat_discount_vat` - Unit pricelist discount with VAT.
- `flat_disposition` - Unit disposition (1+kk, 2+1 etc).
- `flat_floor_number` - Floor number. May be negative for floors under the surface.
- `flat_internal_id` - Unit name / ID in the Realpad system.
- `flat_orientation` - Unit orientation (SV, J, W, S-E etc).
- `flat_price` - Unit price with VAT.
- `flat_price_without_vat` - Unit price without VAT.
- `flat_status` - Unit status (availability), typically project webs will only display units with availability `0` or `1`. Possible values are:
  - 0 - free.
  - 1 - pre-reserved.
  - 2 - reserved.
  - 3 - sold.
  - 4 - not for sale.
  - 5 - delayed.
- `flat_type` - Unit type, possible values are:
  - 1 - flat.
  - 2 - parking.
  - 3 - cellar.
  - 4 - outdoor parking.
  - 5 - garage.
  - 6 - commercial space.
  - 7 - family house.
  - 8 - land.
  - 9 - atelier.

[1] If the unit has more than 1 balcony / terrace / garden, the area is the sum of all the areas.

    - 10 - office.
    - 11 - art workshop.
    - 12 - non-residential unit.
    - 13 - motorbike parking.
    - 14 - creative workshop.
    - 15 - townhouse.
    - 16 - utility room.
    - 17 - condominium.
    - 18 - storage.
    - 19 - apartment.
    - 20 - accommodation unit.
    - 21 - bike stand.
    - 22 - communal area.

Resources

XML payload contains attributes and values in the form of `bd5563ae-abc...`. These are the UIDs (Unique IDentifiers) of the *resources* you can retrieve: unit plans, PDFs, gallery pictures etc. You can fetch resources using **HTTP GET** by retrieving a URL in the following form: **`https://cms.realpad.eu/resource/<UID>`**

Example call in cURL:

```
curl \
	--output cached_resource \
	https://cms.realpad.eu/resource/bd5563ae-abc...
```

Important note: always cache resources "forever". The resources you obtain this way are *immutable*. This means that the same UID will always refer to exactly the same data. If the data changes, you will see a different UID in one of the XMLs. You may freely cache the resources forever: you don't have to re-download the same UID ever again.

Sending leads

Best practices for sending leads include:

- Include a **captcha** on the web to filter out spam at the most opportune moment (nothing down the line has more context than the website itself).
- Always keep a **full log of the leads**, be ready to replay them to the API in case there is any problem at the first try. The CRM can easily handle a lead sent multiple times, but in case the initial call fails for any reason the lead should not be lost!
- If possible include the name of the enquired unit in the call (parameter `internalid`).

Request

Use **HTTP POST** and the endpoint **`https://cms.realpad.eu/ws/v10/create-lead`**

Send the following parameters as a URL-encoded form (**bold** means a required parameter):

- **`login`** - string login obtained from our support.
- **`password`** - string password obtained from our support.
- **`name`** - string first name of the lead.
- **`surname`** - string last name (surname) of the lead.
- `email` - string email address of the lead. No specific format is required.

- `phone` - string phone number of the lead. No specific format is required.
  - **Important note: either `email` of `phone` (or both) must be provided.**
- `note` - string free-form note. Often this is used to transport some note field from the contact form.
- `referral` - integer or omitted, **our support team will provide the value to use**.
- `campaign` - string ID of the campaign that attracted the lead to the web of the project.
- `language` - string, valid ISO 639-1 code of the customer's preferred language. *Examples: sk, cs, en, de.*
- `internalid` - string "internal ID" (sometimes called "number in project") of the unit that the customer sent the request from.
- `tags` - string, comma-separated list of tags to add to the customer.
- `salesman` - integer Realpad database ID of the salesman to assign the lead to, skipping the round-robin algorithm. Only send if instructed so by our support team.
- `partner` - integer Realpad database ID of the partner to limit the round-robin algorithm to. Only send if instructed so by our support team.
- `project` - integer Realpad database ID of the project to assign the lead to.
- `consentsource` - string free-form text describing the source of the consents being obtained. The original intention for this parameter is to contain the IP address from which the lead submitted the contact form.
- `consents` - string comma-separated list of consent IDs (**exact IDs and their purpose will be provided by our support team**).
- The endpoint can accept additional parameters ("attributes"). Our support team will instruct you how to send these in case it's needed.

Example call in cURL:

```
curl \
        --data "login=...&password=...&name=John&surname=Doe&phone=...&email=..." \
        https://cms.realpad.eu/ws/v10/create-lead
```

Response

Same HTTP codes apply as above. **The body of the response upon a successful call (HTTP 200) is just an integer in plain text**: it signifies the Realpad database ID of the lead - either a newly created one or an existing one, if there was a duplicity on email or phone that Realpad resolved.

Testing

Please coordinate with the real estate developer and our support team. When it comes to the pricelist testing we'll trigger a change in the CRM data of a unit (or perhaps replace its floor plan) and observe whether the change is correctly shown on the website. In the opposite direction we'll fill in a few test leads via the contact form and observe whether they are correctly saved to the CRM. In addition to that feel free to check the status of a created lead using a call to the **`get-customer-details`** endpoint:

```
curl \
        --data "login=...&password=...&customerid=..." \
        https://cms.realpad.eu/ws/v10/get-customer-details
```