
Beast - Ralph API Client Documentation

Release 1.0

Allegro Group

March 12, 2014

Contents:

How to install

1.1 Linux

The simplest way to install ralph beast is (you need curl):

```
$ curl https://raw.githubusercontent.com/allegro/ralph_beast/master/install.sh | bash -
```

or if you feel comfortable with pip:

```
$ pip install ralph-beast
```

1.2 Windows

We provide windows binary. Just Download *beast.exe* file and prepare the configuration file.

The configuration file should be placed in the same directory as *beast.exe* or home directory. Config file should be contain such data *Config file - example*

1.3 MacOS

Put on your console below command to install.:

```
$ pip install ralph-beast
```

or:

```
$ curl https://raw.githubusercontent.com/allegro/ralph_beast/master/install.sh | bash -
```

1.3.1 Configuration

Now you need configuration file. Create file

```
~/beast/config
```

Windows binary already contains the file inside directory.

Config file should contain such data *Config file - example*

1.4 Config file - example

```
username="jan.kowalski"  
api_key="478457f9f32323201ebde8ef79cd9d3a028ced56747"  
url="https://ralph-url.com"  
version="0.9"
```

How to use

Use help command to display available options:

```
$ ~/beast -h
```

otherwise...

2.1 List all available resources

If you want to preview all API resources:

```
$ ~/beast show
```

Output:

```
Ralph API, schema:
```

```
-----  
bladeserver  
businessline  
ci  
...  
virtualserver  
windowsdevice
```

You can also list all the resource fields:

```
$ ~/beast show venture --schema
```

Output:

```
Ralph API > venture, schema
```

```
-----  
name*  
cache_version  
created  
symbol*  
modified  
devices  
roles  
show_in_ralph*  
department*  
path  
is_infrastructure
```

```
id*
resource_uri
```

If field name has *, that you can execute filter on this field

2.2 Show details of the selected resource

```
$ ~/beast show venture
```

Output:

```
-----
| name      | cache_version| created   | symbol    | modified  |
|-----|-----|-----|-----|-----|
| Venture1  | 2            | 2012-06-13| venture1  | 2012-06-14|
|-----|-----|-----|-----|-----|
| Venture2  | 5            | 2012-01-23| venture2  | 2012-06-14|
|-----|-----|-----|-----|-----|
| Venture3  | 8            | 2011-10-12| venture3  | 2012-10-31|
|-----|-----|-----|-----|-----|
```

2.2.1 Filter

If you need to see filtered data use:

```
$ ~/beast show venture --filter="symbol=venture2&modified=2012-06-14"
```

Available filters: exact, iexact, contains, icontains, in, gt, gte, lt, lte, startswith, istartswith, endswith, iendswith, range, year, month, day, week_day, hour, minute, second, isnull, search, regex, iregex:

Read more about Django ORM [filters](#).

Output:

```
-----
| name      | cache_version| created   | symbol    | modified  | devices| roles| show_in_ralph| department|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Venture2  | 4            | 2012-01-23| venture2  | 2012-06-14||      |      | x            | 2
```

2.2.2 Fields

If you need to filter data, and no builtin API filter is available, you can use additional filtering by ordinary python expressions using row dict variable

```
$ ~/beast show venture --fields="name, symbol"
```

Output:

```
-----
| name      | symbol      |
|-----|-----|
| Venture1  | venture1    |
|-----|-----|
| Venture2  | venture2    |
|-----|-----|
```

```
-----  
| Venture3 | venture2 |  
-----
```

2.2.3 Limit

Specifies the number of results:

```
~/beast show venture --limit=1
```

Output:

```
Limited rows requested: 1
```

```
-----  
| name      | cache_version| created    | symbol     | modified   |  
-----  
| Venture1  | 2            | 2012-06-13| venture1   | 2012-06-14|  
-----
```

2.2.4 Trim

Use to better trim data:

```
~/beast show venture --trim
```

2.2.5 Width

Limit table width to the specified number of characters:

```
~/beast show venture --width=100
```

2.2.6 Debug

Shows request time:

```
~/beast show venture --debug
```

2.2.7 Export to the file

Beast can export to the csv format.

```
~/beast show venture --csv > ~/ralph_ventures.csv
```

CSV file is encoding to Unicode (UTF-8) and separated by comma.:

Indices and tables

- *genindex*
- *modindex*
- *search*