**Odoo API**

**Restful API for Odoo**

*Posted on January 21st, 2019 • ☕ 3 min read*

**ODOO REST API** documentation, the module can be downladed here <https://www.odoo.com/apps/modules/12.0/restful> it rely on the existing Odoo RPC API interface implementation. It is kind of JSON on http implementation :(

It helps to ease the integration of any third party application with Odoo without the need mastering RPC protocol, it exposes all Odoo RPC methods over HTTP in a fluent and easy way without secrificing speed and performance.

To use the module a deep knowledge of Odoo and its RPC API interface is required without that you may find it a bit difficult to navigate your way through especially when dealing with \*2many.

**How to request for access token**

An access token is a credential that can be used by an application to access an API or restricted information on a server, it is required in other to be able to perform any operations and this token once generated should alway be send a long side any subsequents request either through the headers information or as part of the requests body.

import requests, json

headers = {

'charset':'utf-8',

'content-type': 'application/x-www-form-urlencoded',

}

data = {

'login': 'admin',

'password': 'admin',

'db': 'demo\_db' # This should be replace with your database.

}

base\_url = 'http://example.com'

req = requests.get('{}/api/auth/token'.format(base\_url),

data=data, headers=headers)

content = json.loads(req.content.decode('utf-8'))

# or add the access token to the headers

headers['access-token'] = content.get('access\_token')

print(headers)

**Deleting An Acccess Token**

Some times, an access token might have been compromised, or there may be need to change the existing token as a result of security concerned, below request can be make to delete the user access token.

For any request to be approved after deleting the token, a new token need to be generated as described above

req = requests.delete('%s/api/auth/token'%base\_url, data=data, headers=headers)

**Search [GET] Read Request**

Takes a search domain, returns a recordset of matching records. Can return a subset of matching records (offset and limit parameters) and be ordered (order parameter).

req = requests.get('{}/api/sale.order/'.format(base\_url), headers=headers,

data={'limit': 10, 'domain': []})

# \*\*\*Pass optional parameter like this \*\*\*

{

'limit': 10, 'domain': "[('supplier','=',True),('parent\_id','=', False)]",

'order': 'name asc', 'offset': 10

}

print(req.content)

**Create [POST] Create Request**

Takes a dictionary of field values, or a list of such dictionaries, and returns a recordset containing the records created:

p = requests.post('%s/api/res.partner/'%base\_url, headers=headers,

data=json.dumps({

'name':'John',

'country\_id': 105,

'child\_ids': [{'name': 'Contact', 'type':'contact'},

{'name': 'Invoice', 'type':'invoice'}],

'category\_id': [{'id':9}, {'id': 10}]

}

))

print(p.content)

**PUT Request [Write]**

Takes a number of field values, writes them to all the records in its recordset. Does not return anything:

p = requests.put('http://theninnercicle.com.ng/api/res.partner/68', headers=headers,

data=json.dumps({

'name':'John Doe',

'country\_id': 107,

'category\_id': [{'id': 10}]

}

))

print(p.content)

**DELETE Request**

Deletes the records of the current set

p = requests.delete('http://theninnercicle.com.ng/api/res.partner/68', headers=headers)

print(p.content)