

Create and manage Odoo databases with ERPpeek or XML-RPC



FEBRUARY 21, 2016



YENTHE666

Hi guys,

In this tutorial I will learn you how to connect to your Odoo database with ERPpeek or with simple XML-RPC calls. In this tutorial you will learn how to create new databases, how to install modules, how to add models and how to search for data in the database or write data in the database. These XML-RPC calls can be done from anywhere, also remotely.

In this example I will learn you how to create a new database, how to list the installed modules, how to install a module, how to add models and eventually how to search for data and add data in the database! In this example I will be working with two ways: first with **ERPpeek** (which is an XML-RPC library wrapper which makes coding even easier) and I will then switch to XML-RPC calls, which are slightly more complex but do not need any remote libraries installed.

1. Installing ERPpeek

The first thing that you will need for this library is the ERPpeek library, which you can install very easily:

```
1 sudo pip install erppeek
```

1.1 CREATING A NEW DATABASE

One of the handy methods in ERPpeek is that you can very easily create a new database by command in a Python file. In this blogpost I will be separating all examples completely so I will also add the connection code again every time. This is because I will be switching from Erppeek to normal XML-RPC calls later on, to learn you both concepts.

Alright let us start! Create a new folder somewhere where you can create new Python files in. Now create a new Python file named 'create_database.py'. So, what do we need to create a new database? A database name, the path to the server (either on the same server or remote) and an administrator password for the admin user. A second thing that we need is to import the Python library ERPpeek, so we can use all the handy tools from there. The code will look like this:

```
1 import erppeek
2
3 DATABASE = 'ErpPeekDemoDatabase'
4 SERVER = 'http://localhost:8080'
5 ADMIN_PASSWORD = 'admin'
```

So, we now have the credentials but how do we create the database? We can access all functions with 'erppeek.client' ([you can find all methods in the ERPpeek documentation](#)) which opens up a broad range of options. The first thing we need to do is create a client:

```
1 client = erppeek.Client(server=SERVER)
```

So we have the credentials now, we have a client and now we need to create the database! From here on we can access everything we need from the variable 'client'. Let us first add a validation to see that the database doesn't exist yet and then create a new database with the function 'create_database!'. The code:

```
1 if not DATABASE in client.db.list():
2     print("The database does not exist yet,
3     client.create_database(ADMIN_PASSWORD, D
```

```
4 else:
5     print("The database " + DATABASE + " alr
```

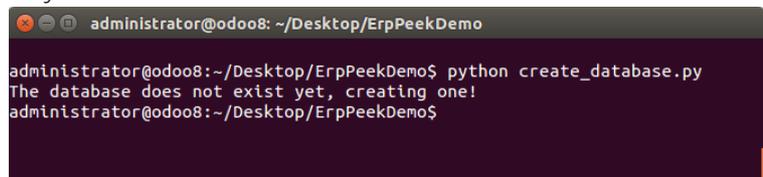
That is all you need to do! Just to be clear, here is whole code:

```
1 import erppeek
2
3 DATABASE = 'ErpPeekDemoDatabase'
4 SERVER = 'http://localhost:8080'
5 ADMIN_PASSWORD = 'admin'
6
7 client = erppeek.Client(server=SERVER)
8
9 if not DATABASE in client.db.list():
10     print("The database does not exist yet,
11         client.create_database(ADMIN_PASSWORD,
12 else:
13     print("The database " + DATABASE + " al
```

When you now execute this Python file it will create the new database for you. Save the Python file and execute it:

```
1 python create_database.py
```

After a few seconds you will see the following result in your terminal:



```
administrator@odoo8: ~/Desktop/ErpPeekDemo
administrator@odoo8:~/Desktop/ErpPeekDemo$ python create_database.py
The database does not exist yet, creating one!
administrator@odoo8:~/Desktop/ErpPeekDemo$
```

Congratulations, you've just created your first Odoo database by command. Now let us continue and see which modules are already installed.

1.2 GETTING INSTALLED MODULES

A handy thing to know is which modules are installed in your Odoo, so let us create a new Python file named 'list_installed_modules.py'. Import ERPpeek and create a new client to connect with:

```
1 import erppeek
2
3 client = erppeek.Client('http://localhost:80
4
```

The first value for the `erppeek.Client` is the server IP (or localhost, which translates to 0.0.0.0) and the port where the Odoo runs on, in this case 8080. The second value is the name of your database (ErpPeekDemoDatabase), the third is your Odoo username (admin) and the fourth is the user its password (admin).

So, we now have a working connection to the Odoo database and we now need to find out which modules are installed. This data is saved in the model `'ir.module.data'`, when the field `'state'` is set to installed we know the module is installed. This translates to the following code with ERPpeek:

```
1 proxy = client.model('ir.module.module')
2 installed_modules = proxy.browse([('state',
3
```

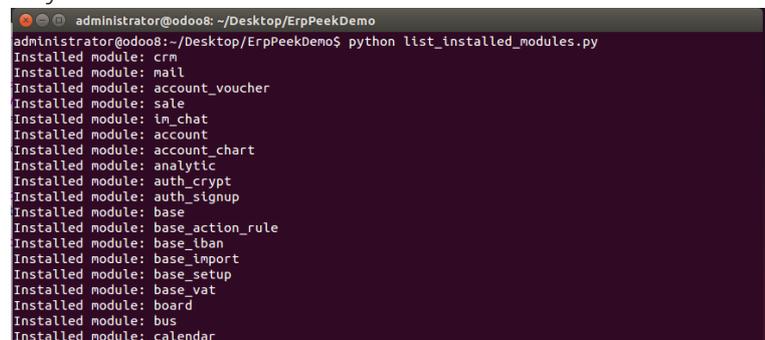
And for demo purposes let us simply print out this data in the terminal:

```
1 for module in installed_modules:
2     print('Installed module: ' + module.name
3
```

When you now execute this Python file it will it will print out all installed modules. Save the Python file and execute it:

```
1 python list_installed_modules.py
```

After a few seconds you will see the following result in your terminal:



```
administrator@odoo8: ~/Desktop/ErpPeekDemo
administrator@odoo8:~/Desktop/ErpPeekDemo$ python list_installed_modules.py
Installed module: crm
Installed module: mail
Installed module: account_voucher
Installed module: sale
Installed module: im_chat
Installed module: account
Installed module: account_chart
Installed module: analytic
Installed module: auth_crypt
Installed module: auth_signup
Installed module: base
Installed module: base_action_rule
Installed module: baseiban
Installed module: base_import
Installed module: base_setup
Installed module: base_vat
Installed module: board
Installed module: bus
Installed module: calendar
```

We now know which modules are installed, but wouldn't it be great if we could install modules by command? Read on in chapter 1.3!

1.3 INSTALLING A MODULE

Create a new Python file and name it 'install_module.py'. Import ERPpeek and create a client just like you did in the previous examples:

```
1 import erppeek
2 client = erppeek.Client('http://localhost:80
3
```

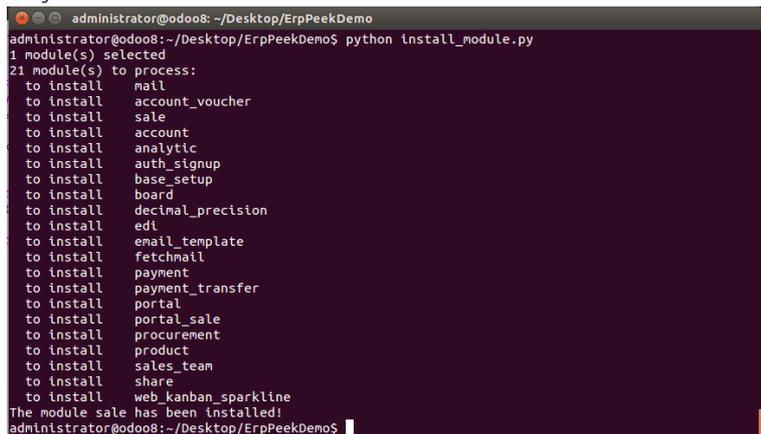
We can now get all modules with 'client.Modules' and we can check if it is installed with the setting installed=False. When the module is in the state uninstalled we can install it with client.install. In this example I will check for the module sale and if it isn't installed I will install it. The code:

```
1 modules = client.modules('sale', installed=False)
2 if 'sale' in modules['uninstalled']:
3     client.install('sale')
4     print('The module sale has been installed')
5
```

When you now execute this Python file it will install the module for you (if it isn't installed yet). Save the Python file and execute it:

```
1 python install_module.py
```

After a few seconds you will see the following result in your terminal:



```
administrator@odoo8: ~/Desktop/ErpPeekDemo
administrator@odoo8:~/Desktop/ErpPeekDemo$ python install_module.py
1 module(s) selected
21 module(s) to process:
to install mall
to install account_voucher
to install sale
to install account
to install analytic
to install auth_signup
to install base_setup
to install board
to install decimal_precision
to install edi
to install email_template
to install fetchmail
to install payment
to install payment_transfer
to install portal
to install portal_sale
to install procurement
to install product
to install sales_team
to install share
to install web_kanban_sparkline
The module sale has been installed!
administrator@odoo8:~/Desktop/ErpPeekDemo$
```

Do you notice anything interesting? Odoo will automatically check all dependencies and if some aren't met they will be installed by Odoo! So we have now created a new database, we've checked which modules are installed and installed one which wasn't installed yet. Now let us continue by adding a new contact to the 'res.partner' model.

1.4 ADDING NEW RECORDS TO A MODEL WITH XML-RPC

So we've seen a few examples with ERPpeek now but we do not really need this library. We can just as easy do simple XML-rpc calls! It involves a bit more coding but it gives you the same possibilities as with the ERPpeek library. From here on I will continue this tutorial with XML-RPC calls for learning purposes. Do know that you can also use the ERPpeek library if you'd like to!

So we no longer need the ERPpeek library but we now need the library 'xmlrpclib'. Import it and create a connection to the Odoo environment. We first need a ServerProxy which refers to the URL 'localhost:xxxx/xmlrpc/common' we then need an uid and then we need a socket to connect to. The code will look like this:

```
1 import xmlrpclib
2
3 username = 'admin' # The Odoo user
4 pwd = 'admin'# The password of the Odoo use
5 dbname = 'ErpPeekDemoDatabase' # The Odoo d
6
7 # OpenERP Common login Service proxy object
8 sock_common = xmlrpclib.ServerProxy ('http:
9 uid = sock_common.login(dbname, username, p
10
11 # replace localhost with the address of the
12 sock = xmlrpclib.ServerProxy('http://localh
13
```

Do you see the similarities between the previous examples where we instantiated a client in one line? The working method is in essence the same, the code just looks slightly different and it is a bit more complex. What we did in one line in the previous examples now takes 6 lines but there is no difference in the possibilities.

We now have a client to connect with Odoo so let us now create a new record. The new record should be written in a dictionary format where you can add values for every field you would like to fill. For example I will create a new contact on the model

'res.partner' with the name 'Fabien' and the e-mail 'example@odoo.com'. This translates in the following code:

```
1 partner_record = {
2     'name': 'Fabien',
3     'email': 'example@odoo.com'
4 }
```

The last part is to write the new record to Odoo (with XML-RPC) by using sock.execute. The code:

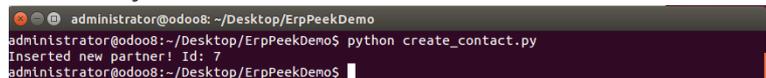
```
1 # Calling the remote ORM create method to cr
2 result = sock.execute(dbname, uid, pwd, 'res
3 print('Inserted new partner! Id: ' + str(res
```

What does this code do? It connects to the database with the credentials you've provided at the top of your script, it calls the model 'res.partner', uses the method 'create' (which creates a new record) and eventually we pass the dictionary to Odoo to write it in the database.

When you now execute this Python file it will create a new record in the database, on the model 'res.partner'. Save the Python file and execute it:

```
1 python create_contact.py
```

After a second or two you will see the following result in your terminal:



```
administrator@odoo8: ~/Desktop/ErpPeekDemo
administrator@odoo8:~/Desktop/ErpPeekDemo$ python create_contact.py
Inserted new partner! Id: 7
administrator@odoo8:~/Desktop/ErpPeekDemo$
```

The new contact has been created with the id 7.

When you now open up your Odoo and go to sales > contacts you will see the new contact:



Congratulations, you can now also connect with Odoo through XML-RPC calls and you can write data in the Odoo database! Now, let us learn how to fetch data from the database.

1.5 SEARCHING FOR DATA IN THE DATABASE

Create a new Python file named 'search_contact.py', import the xmlrpclib, create credentials and create a socket connection:

```
1 import xmlrpclib
2
3 username = 'admin' # The Odoo user
4 pwd = 'admin'# The password of the Odoo use
5 dbname = 'ErpPeekDemoDatabase' # The Odoo d
6
7 # OpenERP Common login Service proxy object
8 sock_common = xmlrpclib.ServerProxy('http:
9 uid = sock_common.login(dbname, username, p
10
11 # replace localhost with the address of the
12 sock = xmlrpclib.ServerProxy('http://localh
13
```

Create a new sock.execute, just like you did in the previous example, but this time use the search method on the 'res.partner' model to get all contacts:

```
1 # Calling the remote ORM create method to se
2 contact_ids = sock.execute(dbname, uid, pwd,
3 print('Contact IDs found: ' + str(contact_id
```

When we would execute the code at this point it would print us all the ID's from all records on the model 'res.partner', but we as humans don't know a lot by ids so let us translate these ids in to readable content! Loop over all the ids and create another sock.execute with the read function to get all the details. Finally print out the details of all contacts:

```
1 # Get the details from all contacts
2 for contact_id in contact_ids:
3     # Get the details by the id
4     contact = sock.execute(dbname, uid, pwd,
5     print('Contact name: ' + contact['name']
```

When you now execute this Python file it will search all records in the database, on the model 'res.partner'. Save the Python file and execute it:

```
1 python search_contact.py
```

After a second or two you will see the following result in your terminal:

```
administrator@odoo8: ~/Desktop/ErpPeekDemo
administrator@odoo8:~/Desktop/ErpPeekDemo$ python search_contact.py
contact IDs found: [3, 6, 5, 1]
contact name: Administrator
contact name: Fabien
contact name: Template User
contact name: Your Company
administrator@odoo8:~/Desktop/ErpPeekDemo$
```

Congratulations! You can now also search for data in Odoo with XML-RPC!

There is one more thing I would like to learn you: how to create new models through XML-RPC calls.

1.6 CREATING NEW MODELS WITH XML-RPC CALLS

A rather often needed thing in Odoo is to create new models and this can also be done through XML-RPC calls! There is one downside though: you will need to create the model and field names starting with `x_`. This is not the case when you create a new model in a new module so it is probably better to create models that way.

In case you should do it from somewhere remote you can do it with calls though! This could be done with the XML-RPC library but we would need a lot more code so for this example I'm going back to ERPpeek!

Create a new Python file named 'create_new_model.py', import ERPpeek and create a client:

```
1 import erppeek
2
3 client = erppeek.Client('http://localhost:80
4
```

In order to create a new model you will need access to the models 'ir.model' and 'ir.model.fields' so create two variables for this:

```
1 model_proxy = client.model('ir.model')
2 field_proxy = client.model('ir.model.fields')
```

The first step is to create a new model, which will be empty, and after this is done you can add fields to it. First create a new model:

```
1 values = {
2     'model': 'x_example_tutorial',
```

```
3     'name': 'Example model',
4     'state': 'manual',
5 }
6
7 # This will create the new model!
8 model = model_proxy.create(values)
9
```

So what does this do? It creates a new model with the name 'x_example_tutorial' and the text will be 'Example' model. The model_proxy.create will finally create the new model.

Now that we have a new model the last thing we need to do is add fields to the model. Create a new dictionary with the field and all settings you want to specify:

```
1 values = {
2     'name': 'x_name',
3     'ttype': 'char',
4     'size': 64,
5     'field_description': 'Firstname',
6     'model_id': model.id,
7     'model': model.model,
8     'domain': '[]',
9 }
10 field = field_proxy.create(values)
11 print('Congratulations your new model has b
12
```

This will create a new field with the name 'x_name', the type is a char with a length of 64 characters, the description of the field is 'Firstname' and we will simply add it on the just created model.

field_proxy.create will eventually insert the new field in the database!

Thats it, congratulations! You now know quite a lot about XML-RPC calls and ERPpeek to access Odoo databases.

2. Conclusion

Using simple XML-RPC calls is a very powerfull tool to access and manage your Odoo database but when you use the ERPpeek library life gets even more easy. You can create new databases in a few lines of

code, you can install modules and write to the database remotely.

Do you want to try the demo code and see the source code of this tutorial? [You can view on my Github account.](#)

Has this tutorial helped you, do you have any feedback or questions? Post away!



This blog is made possible by Odoo Experts.

Share this:



Like this:

Loading...

ERPPEEK XML-RPC

25 thoughts on “Create and manage Odoo databases with ERPpeek or XML-RPC”



Yves Goldberg

I find your blog most interesting. Thank you for the teaching.



REPLY FEBRUARY 22, 2016 AT 08:15