A newbie introduction to proteus

N. Évrard

B_2CK

Tryton Unconference Barcelona 2013

TUB2013

Outline

1. What is proteus?
   - A python library
   - To access a tryton server
   - As a client
2. Use cases
   - Testing
   - Scripting
   - Development
3. Nifty features
   - Field types
   - Buttons
   - Wizards
4. Some drawbacks
5. A small demo of proteus
6. Summary

TUB2013

A python library

proteus is just like any another (good) python library

- `pip` installable
- usable in scripts, but also in the python shell, `ipython`, `bpython`, . . .

But it is also a good starting point to get familiar with some advanced features of python like descriptors, metaclasses, context managers, . . .

Accessing a local server

proteus can be used to access a local Tryton server.

Example

```
>>> from proteus import config
>>> config = config.set_trytond(config_file=’foo.cfg’,
...     user=’admin’, password=’admin’)
```

This command will import a local trytond module initialized with configuration specified in `foo.cfg` using the provided credentials.

Accessing a remote server

But **proteus** can also be used to access a remote Tryton server.

Example

```
>>> from proteus import config
>>> config = config.set_xmlrpc(
...     ’http://admin:admin@localhost:8069/foo’)
```

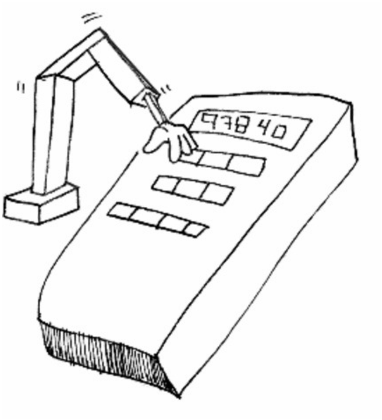
In this case, proteus will connect to the Trytond server listening for XML-RPC connections on port 8069 on localhost. The usual syntax is used to specify the credentials.

As a client

proteus is built to mimic the Tryton client.

- default values are set when creating new object
- `on_change`, `on_change_with` calls are triggered
- context on the fields is used and eventually PYSON-ly evaluated

proteus can be used for automated testing



- 29 **doctests** files are included into the trytond and modules repositories.
- **proteus** mimics the client and allows to make functional testing through the use of scenarios
- We’re testing access rights, workflows, wizards, . . .

proteus is used for automating tasks

- **proteus** is used to populate `http://demo.tryton.org`
- Importing data from a legacy database
- Generating timesheet reports
- ...

proteus can be used while developing

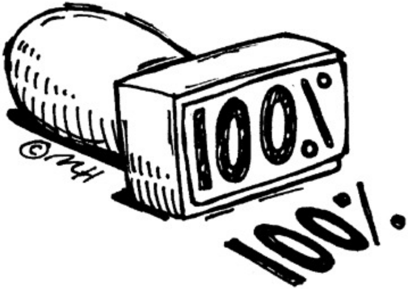
Tired of repeating the same action on the client while developing? **proteus** can help you

- enter the same data
- execute the same wizard
- click the same buttons

Just from your shell prompt or your favorite python interpreter!

The supported field types

Every tryton type is supported by proteus. **All of them**.



proteus can call buttons

It's just a matter of calling a `classmethod`.

Example

```
sale = Sale()
sale.party = customer
sale.sale_date = sale_date
sale_line = sale.lines.new()
sale_line.product = product
sale_line.quantity = random.randint(1, 50)
sale.save()
Sale.quote([sale.id], config.context)
```

proteus can also execute wizards

Example

```
company_config = Wizard('company.company.config')
company_config.execute('company')
company = company_config.form
party = Party(name='Michael Scott Paper Company')
party.save()
company.party = party
company.currency = usd
company_config.execute('add')
```

- Created with `Wizard`
- `.form` is used to access `ModelViews`
- Transitions are called with `execute`

TUB2013

Where proteus could be better

- Importation of bazillions of records: **proteus** is not ment for speed.
- The API could be better for the buttons
- proteus does not speak JSON-RPC

Let’s make a small demo

Proteus is cool

- It uses cool python features
- You can write tests and scripts with it
- Auto-completion in advanced python shells
- `pip install proteus`!