
csv2oerp Documentation

Release

Stéphane Mangin

January 29, 2013

CONTENTS

INTRODUCTION

csv2oerp is a convenient library to import data from CSV format to an *OpenERP* instance.

csv2oerp is a simple and fast method to perform an import, through OpenERP model's fields bound to CSV's columns. You can easily ordered your data before coding any lines as the manner of *OpenERP*. You can perform post-import field treatment, omit or modify it. You can also according to criterias, skip a line or do not decide to create an object being processed.

QUICK START

Import csv2oerp and some callbacks:

```
>>> from csv2oerp import Model, Field, Openerp, Session, show_stats
>>> import csv2oerp.callbacks as cb
```

Configure OpenERP connection:

```
>>> oerp = Openerp('198.168.0.1', 8069, 'admin', 'admin', 'database', 'fr_FR')
```

Create a new importation instance:

```
>>> example = Session('Example name for logger', 'example_file.csv',
    delimiter=u';', quotechar=u'"', encoding=u'utf-8',
    offset=1, limit=10)
```

Define a custom callback(Field's value pre-treatment):

```
>>> def country_code(session, model, field, value, line_num):
...     """Return the first two uppered characters from current column value
...     """
...     return value[:2].upper()
```

Define your mapping to link both csv and OpenERP:

```
>>> res_partner = Model('res.partner', fields=[

...     Field('name', columns=[1]),
...     Field('siren', columns=[2]),
...     Field('website', columns=[16]),
...     Field('comment', columns=[56]),

...     ], update=False, search=['siren']) # Unique by siren and no update

>>> res_country = Model('res.country', fields=[

...     # A default can be used only if NULL value from CSV's column
...     Field('code', columns=[13], callbacks=(country_code), default='FR'),
...     Field('name', columns=[13], default='FRANCE'),

...     ], update=False, search=['code', 'name'])

>>> res_partner_address = Model('res.partner.address', fields=[

...     # Custom field's value without link to a CSV's column
...     Field('type', custom='default'),
```

```

...     # Simple fields
...     Field('zip', columns=[9], default='35000'),
...     Field('city', columns=[10], default='RENNES'),
...     Field('phone', column=[14]),
...     Field('fax', columns=[15]),
...     Field('email', columns=[17], unique=True), # Unique email
...     Field('cedex', columns=[68]),

...     # Mixing columns (concatenation)
...     Field('street', columns=[7, 6], method='concatenate'),
...     Field('street2', columns=[8, 5], method='concatenate'),

...     # Model's relation with dynamic insertion from OpenERP database
...     # Not native object from OpenERP framework
...     Field('region_id', columns=[11],
...           callbacks=(cb.get_id('res.region', ['name'])))
...     Field('dep_id', columns=[12],
...           callbacks=(cb.get_id('res.dep', ['name'])))

...     # Model's relations
...     Field('country_id', relation=res_country),

...     # Model's relations with no NULL value allowed (otherwise,
...     # skip current Model creation)
...     Field('partner_id', relation=res_partner, required=True),

... ], search=['type', 'partner_id'])

```

Finally join objects to the session which starts the import process:

```

>>> # No need to inject relationnals models, only the first level is needed
>>> example.bind(oerp, models=[res_partner_address])

```

And show statistics of objects's activities during the importation process:

```

>>> show_stats()

```

DOWNLOAD AND INSTALL

See *download-install* section.

DOCUMENTATION

See *content* section.

SUPPORTED DESKTOP VERSIONS

All architectures.

SUPPORTED PYTHON VERSIONS

csv2oerp support Python versions 2.6 and 2.7.

LICENSE

This software is made available under the LGPLv3 license.

BUGS OR SUGGESTIONS

Please, feel free to report bugs or suggestions in the [Bug Tracker](#)!