

OCA: da Oggi Contribuisco Anch'io!

Alex Comba
Agile Business Group

OCA: Odoo Community Association



The Odoo Community Association, or OCA, is a nonprofit organization whose mission is to support the collaborative development of Odoo features and promote its widespread use.

<http://odoo-community.org/>

OCA: organizzazione (1/3)

■ **Teams**

- **Community Backport Team** (OCB) Odoo additional and early bugfixes
- **Business Topics Teams** (18) Accounting, Sales, Logistics, ...
- **Localization Teams** (16) By country
- **Vertical Interest Teams** (8) Hotel, Construction, Medical, ...
- **Connector Teams** (7) Integrate Odoo with Magento, Sage, LIMS, ...
- **Community Maintainers Tools Team**

<http://odoo-community.org/page/List>

OCA: organizzazione (2/3)

Ogni team può essere composto da uno o più progetti.

CRM & Sales Maintainers

Related Projects

Find our work



Sale Workflow

This project aim to deal with modules related to manage sale and their related workflow.

[On Github](#)



CRM

This project aim to deal with advanced CRM features, like stage monitoring, letters, ...

[Github](#)



Partner & Contact Management

This project aim to deal with modules related to manage contact and partner

[On Github](#)



Sale Financial Controlling

This project aim to deal with modules related to manage financial aspect of Sales.

[On Github](#)



Sales Reports

This project aim to deal with modules related to manage sale reports.

[Github](#)




Commission Management

This project aim to deal with modules to ease the Incentive Compensation Management.

[On Github](#)

OCA: organizzazione (3/3)

Ad ogni progetto corrisponde un repo su github.com/OCA

 Odoo Community Association


CRM & Sales Maintainers


24 MEMBERS	6 REPOSITORIES
------------	----------------


Leave


This team grants **Write** access: members can read from and push to the team's repositories.


MembersRepositories


 [oca/commission](#)

 [oca/crm](#)

 [oca/partner-contact](#)

 [oca/sale-financial](#)


 [oca/sale-reporting](#)


 [oca/sale-workflow](#)


Repositories 97


Find a repository...


AllForksSourcesPrivatePublic


 [OCA/account-financial-tools](#)


 [OCA/OCB](#)


 [OCA/vertical-community](#)


 [OCA/hr](#)


 [OCA/bank-statement-reconcile](#)

 [OCA/server-tools](#)

 [OCA/l10n-spain](#)

 [OCA/l10n-italy](#)

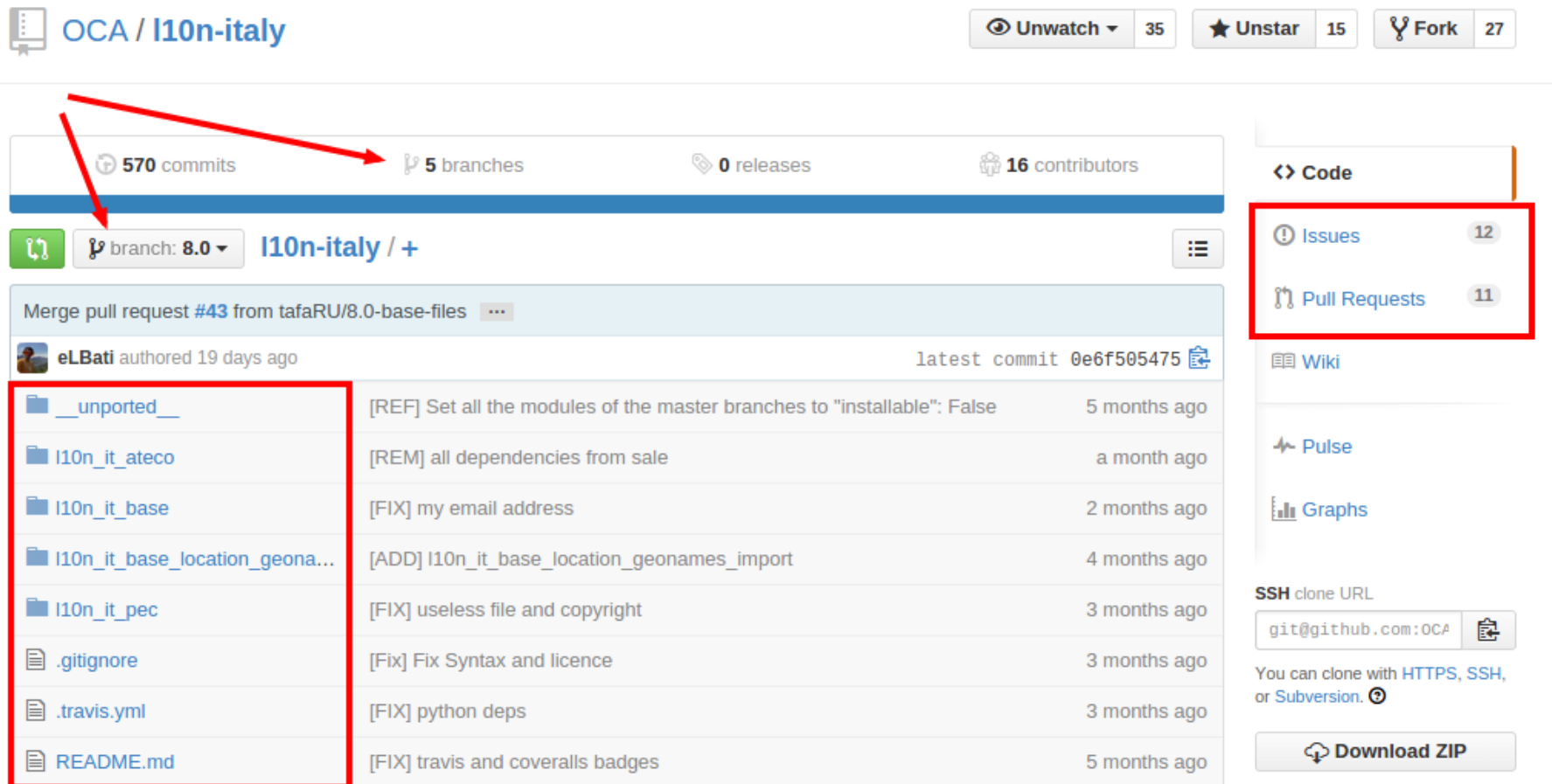
 [OCA/bank-payment](#)

 [OCA/vertical-ngo](#)

Show 87 more repositories...

Struttura di un repo su github/OCA

Per ogni branch: [moduli] + .gitignore + .travis.yml + README.md



The screenshot shows the GitHub repository page for 'OCA / l10n-italy'. The repository has 570 commits, 5 branches, 0 releases, and 16 contributors. The current branch is '8.0'. The file structure is listed on the left, and the commit history is shown on the right. A red box highlights the file structure, and a red arrow points to the '5 branches' link.

Repository Statistics:

- 570 commits
- 5 branches
- 0 releases
- 16 contributors

File Structure (Highlighted):

- __unported__
- l10n_it_ateco
- l10n_it_base
- l10n_it_base_location_geona...
- l10n_it_pec
- .gitignore
- .travis.yml
- README.md

Commit History:

Commit Message	Time Ago
[REF] Set all the modules of the master branches to "installable": False	5 months ago
[REM] all dependencies from sale	a month ago
[FIX] my email address	2 months ago
[ADD] l10n_it_base_location_geonames_import	4 months ago
[FIX] useless file and copyright	3 months ago
[Fix] Fix Syntax and licence	3 months ago
[FIX] python deps	3 months ago
[FIX] travis and coveralls badges	5 months ago

Repository Actions:

- Issues: 12
- Pull Requests: 11
- Wiki
- Pulse
- Graphs

SSH clone URL: git@github.com:OCA

Download ZIP

QA: Quality Assurance (1/3)

Per assicurare la qualità del codice rilasciato, OCA si avvale di alcuni tools QA:

- ***<https://travis-ci.org/>***
- ***<https://coveralls.io/>***

Travis CI rileva automaticamente quando un nuovo commit viene “pushato” sul repository, e ogni volta che questo accade, si occupa di creare una nuova build del progetto, eseguire i test e segnalarne il risultato positivo con una luce **verde** o negativo con una luce **rossa**.

Coveralls determina quale percentuale del codice del progetto viene effettivamente coperta dai test.

QA: Quality Assurance (2/3)

All'interno del progetto OCA/maintainer-quality-tools (MQT) sono contenuti gli strumenti QA utilizzati durante le build su Travis CI.

- ***sample_files/.travis.yml*** (file template di configurazione di Travis CI per ciascun progetto OCA)
- ***travis/travis_run_tests***
 - ***travis/test_flake8***
 - ***travis/cfg/travis_run_flake8.cfg***
 - ***travis/cfg/travis_run_flake8__init__.cfg***
 - ***travis/test_pylint***
 - ***travis/cfg/travis_run_pylint.cfg***
 - ***travis/test_server***

QA: Quality Assurance (3/3)

*travis_run_tests exited with 0 -> **LUCE VERDE!!!***

```
+=====
| Tests summary:
|-----
| test_flake8           Success
| test_pylint           Success
| test_server           Success
+=====
```

```
The command "travis_run_tests" exited with 0.
```

```
$ coveralls
```

```
Done. Your build exited with 0.
```

README.md progetto

OCA/maintainer-quality-tools/blob/master/README.md

📖 README.md

build passing coverage 100%

Odoo Italia Modules

README modulo (1/2)

A partire da ec0b770[1] la chiave 'description' può essere impostata anche in ['README.rst', 'README.md', 'README.txt']

reST e markdown rendono la visualizzazione della descrizione del modulo più accattivante

Al suo interno si possono aggiungere le sezioni:

- ***Authors***
- ***Contributors***

In github, le issues, i commenti, e le descrizioni delle PR utilizzano un dialetto di markdown^[2]

[1] <https://github.com/odoo/odoo/commit/ec0b770ed3c4f4fbdf0f24930aaa05bcc5cb0b53>

[2] <https://help.github.com/articles/github-flavored-markdown/>

README modulo (2/2)

Maggior visibilità per Authors e Contributors

README.rst

Account Move Template - Templates for Journal Entries

The user can configure journal entries templates, useful for recurring entries. The amount of each template line can be computed (through python code) or kept as user input.

If user input, when using the template, user has to fill the amount of every input lines.

The journal entry form allows to load, through a wizard, the template to use and the amounts to fill.

Credits

Authors:

- Davide Corio <davide.corio@agilebg.com>
- Lorenzo Battistini <lorenzo.battistini@agilebg.com>
- Paolo Chiara <p.chiara@isa.it>
- Franco Tampieri <franco.tampieri@agilebg.com>

Contributors:

- Alex Comba <alex.comba@agilebg.com> (Port to V8)



Account Move Template

By Agile Business Group

Upgrade

Uninstall

Information

Technical Data

Installed Features

Website

<http://www.agilebg.com>

Category

Accounting

Summary

Templates for recurring Journal Entries

Technical Name

account_move_template

License

Affero GPL-3

Latest Version

8.0.1.0

Account Move Template - Templates for Journal Entries

The user can configure journal entries templates, useful for recurring entries. The amount of each template line can be computed (through python code) or kept as user input.

If user input, when using the template, user has to fill the amount of every input lines.

The journal entry form allows to load, through a wizard, the template to use and the amounts to fill.

Credits

Authors:

- Davide Corio <davide.corio@agilebg.com>
- Lorenzo Battistini <lorenzo.battistini@agilebg.com>
- Paolo Chiara <p.chiara@isa.it>
- Franco Tampieri <franco.tampieri@agilebg.com>

Contributors:

- Alex Comba <alex.comba@agilebg.com> (Port to V8)

Come contribuire su github

I principali strumenti messi a disposizione da github per contribuire sono:

- ***Pull Request (PR)***
- ***Issue***

Una PR è il modo per comunicare che abbiamo apportato delle modifiche che vorremmo venissero incluse nel progetto. La tipologia utilizzata per OCA è il fork and pull.

Le Issues vengono utilizzate per segnalare task , bug o feature di un progetto. Queste andrebbero sempre controllate ed utilizzate, soprattutto quando ci prepariamo ad inviare una PR. Potrebbe infatti darsi che il bug che stiamo correggendo sia noto e già segnalato da qualcun'altro.

hub: come sfruttare al meglio github

 hub

MAKES GIT BETTER WITH GITHUB

documentation: [man hub](#)

hub is a command-line wrapper for git that makes you better at GitHub.

<https://hub.github.com/>

Per estendere git con i comandi di hub è sufficiente aggiungere a .bashrc o .bash_profile:

eval "\$(hub alias -s)"

Workflow per PR fork and pull (1/3)

Supponiamo di aver scovato un bug in un modulo di OCA/l10n-italy branch 8.0 e di voler proporre la sua fix.

\$ git clone OCA/l10n-italy

\$ cd l10n-italy

\$ git browse -- issues (apre <https://github.com/OCA/l10n-italy/issues>)

\$ git checkout -b 8.0-fix-issue1

(scrivo le modifiche...)

\$ git add -p

\$ git commit -m "Close issue #1"

Un messaggio di commit scritto bene rende il processo di review più veloce.

<https://help.github.com/articles/closing-issues-via-commit-messages/>

Workflow per PR fork and pull (2/3)


Non potendo scrivere direttamente sul progetto, devo crearne una copia, ovvero un fork.

\$ git fork

- **(fork su github: tafaRU/l10n-italy)**
- **(git remote add tafaRU git@github.com:tafaRU/l10n-italy.git)**

\$ git push tafaRU

Your recently pushed branches:

 **tafaRU:8.0-fix-issue1** (less than a minute ago)

 **Compare & pull request**

In alternativa:

\$ git pull-request


(apre il text editor per modificare il titolo, aprire il link della PR appena creata e modificare descrizione)

Workflow per PR fork and pull (3/3)

Ora che la PR è stata aperta inizia il processo di review, una vera è propria conversazione in stile social in cui potremmo discutere insieme ai vari reviewer. Dovremmo essere pronti a sostenere discussioni, applicare ulteriori fix o ad accettare un eventuale rifiuto.


L'accettazione della PR e quindi il suo merge dipende esclusivamente da due fattori:

- **> 2** 👍 [1][2]

- 

✓ **All is well** — The Travis CI build passed [Details](#)

This pull request can be automatically merged.
You can also merge branches on the [command line](#).

 **Merge pull request**

[1] <http://odoo-community.org/page/how-to> (con 3 approve, merge quasi immediato)

[2] emojis - :+1:

Come muoversi tra i vari branches

Vediamo cosa dovrete fare per contribuire allo sviluppo di 8.0-fix-issue1:

\$ git remote add -p tafaRU

controllo i remotes

\$ git remote -v

\$ git fetch tafaRU - \$ git fetch -all

\$ git checkout 8.0-fix-issue1

\$ git checkout -b 8.0-imp-issue1

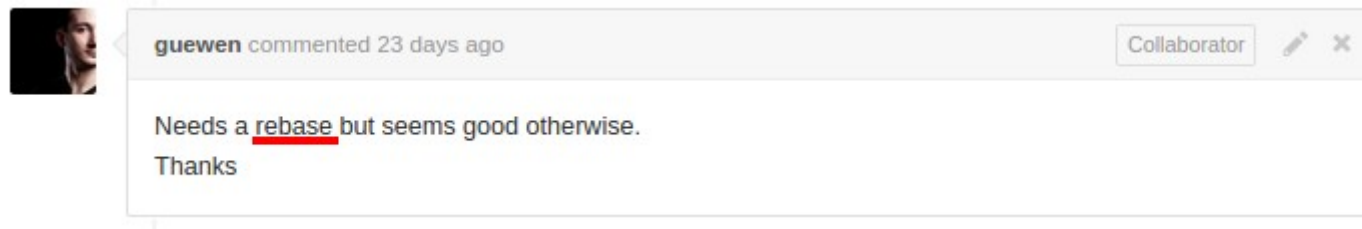
aggiungo le modifiche e le committo

\$ git push your_account

faccio PR con base tafaRU/l10n-italy e branch 8.0-fix-issue1

Rebase una PR

Una delle richieste più ricorrenti durante una PR:



Questo ha come risultato:

- ***integrare nel vostro branch le modifiche approdate sul branch principale***
- ***rilanciare la build di Travis CI***

A partire dal vostro branch 8.0-fix-issue1:

\$ git rebase origin/8.0 (opzione -i per pulire commit)
\$ git push -f (riscrivo la history!!!)

<https://github.com/edx/edx-platform/wiki/How-to-Rebase-a-Pull-Request>

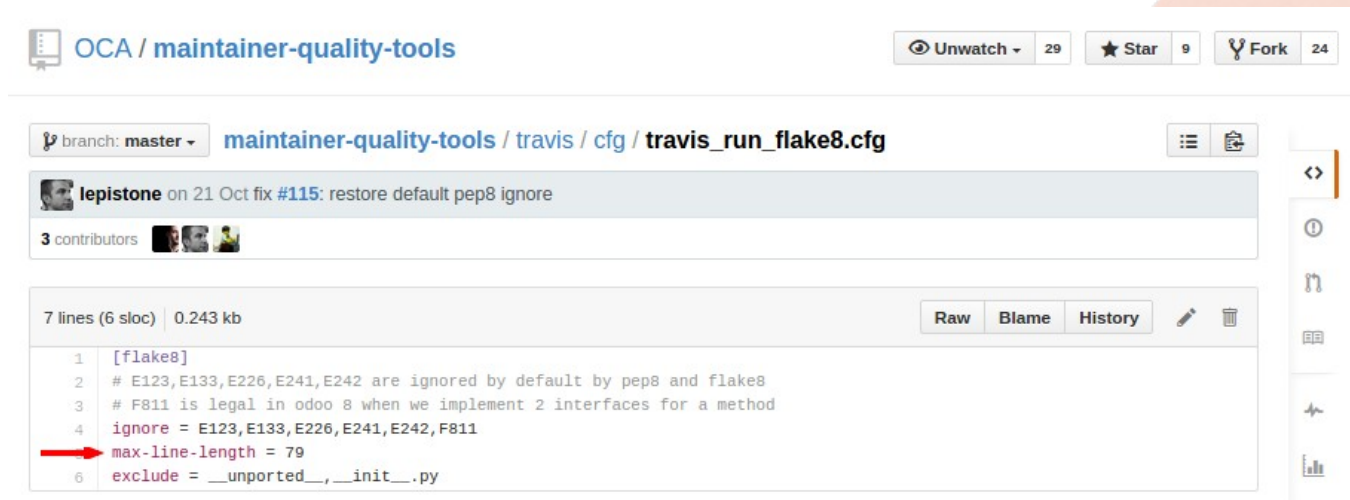
Convenzioni e linee guida

Quelle presenti sono ancora incomplete o poco aggiornate:

[*http://odoo-community.org/page/how-to*](http://odoo-community.org/page/how-to)

Quello che fa fede al momento sono i tools presenti nell' MQT repo, per tale motivo vi consiglio di impostare il watching^[1], per essere notificati su tutte le nuove PR e nuove issue.

Ad es. <https://github.com/OCA/maintainer-quality-tools/issues/46>



```
[flake8]
# E123,E133,E226,E241,E242 are ignored by default by pep8 and flake8
# F811 is legal in odoo 8 when we implement 2 interfaces for a method
ignore = E123,E133,E226,E241,E242,F811
max-line-length = 79
exclude = __unported__,__init__.py
```

[1] <https://help.github.com/articles/watching-repositories/>

Come contribuire: review di PR

- **Review di PR**
 - *Code review*
 - *Manual testing*
 - *Functional review*

Ogni venerdì si tiene l'OCA Review Day:

- *<https://waffle.io/oca/maintainers-tools>*
- *<https://github.com/pulls?user=OCA>*

Canali di comunicazione

- **Mailing lists**
 - <https://www.odoo.com/page/odoo-community>
- **IRC (freenode.net)**
 - **#openobject**
 - <https://botbot.me/freenode/openobject/>
 - **#OCA** (discutere le PR e vedere aggiornamenti PR)
- **Twitter**
 - **@OdooCommunity**
 - **@OCA_Bot** (merge di PR)

OCA: Ovvero, Con un'Aforisma

"Given enough eyeballs, all bugs are shallow"

Eric Steven Raymond(The Cathedral and the Bazaar)

GRAZIE!

- ***GitHub: github.com/tafaRU***
- ***Twitter: [@tafaRU](https://twitter.com/tafaRU)***
- ***IRC: [tafaRU](#)***
- ***LinkedIn: it.linkedin.com/in/tafaru***
- ***Gmail: alex.comba@agilebg.com***