
niteoweb.jvzoo Documentation

Release 0.3

December 03, 2012

CONTENTS

A Plone add-on that integrates [JVZoo](#) digital products retailer system with [Plone](#) to enable paid memberships on your site.

- [Source code @ GitHub](#)
- [Releases @ PyPI](#)
- [Documentation @ ReadTheDocs](#)
- [Continuous Integration @ Travis-CI](#)

HOW IT WORKS

1. Visitor comes to `yoursite.com/order` (or similar) and clicks *Order* link.
2. Visitor is sent to JVZoo's order form (on `http://jvzoo.com`), where he enters his personal information and performs payment.
3. JVZoo calls-back a special view on your plone site (`/@@jvzoo`), which reads POST data from JVZoo, verifies it against your `Secret Key` and creates a new member.
4. The following information is stored in member data for later use:
 - product_id** JVZoo's *Product ID* of the purchased item.
 - product_name** JVZoo's *Product Name* of the purchased item.
 - affiliate** Affiliate who referred the buyer.
 - last_purchase_id** JVZoo's *Receipt ID* of the last purchase. This field gets updated on every recurring payment.
 - last_purchase_timestamp** Exact timestamp of the last purchase. This field gets updated on every recurring payment.
5. Upon creating a new member, Plone sends an email with login password.
6. An `IMemberCreateEvent` is emitted upon creating a new member.
7. The new member can now login and use the site.
8. It is possible to create a `product_id` to `group_name` mapping in Plone Control Panel. This means that if a member purchased a product which is listed in this mapping, the member will also be added to a group mapped to this product.

Note: If a member already exists in Plone, then the `/@@jvzoo` view simply updates `last_purchase_id` and `last_purchase_timestamp` member fields. The member will also be added to the new product group, but also kept in the old.

DEMO

You can see this product in action at [BigContentSearch](#).

INSTALLATION

To install `niteoweb.jvzoo` you simply add `niteoweb.jvzoo` to the list of eggs in your buildout, run buildout and restart Plone. Then, install *niteoweb.jvzoo* using the Add-ons control panel.

CONFIGURATION

4.1 JVZoo

Go to JVZoo.com and use Sellers -> Add a Product to add a new *Product*.

Then check the option External Program Integration. For *URL* set `http://yoursite.com/@@jvzoo`. Under the My Account page set the JVZIPN Secret Key.

4.2 Plone

Go to Site Setup -> jvzoo control panel form and configure the Secret Key field by pasting in the *Secret Key* you defined above.

You can also configure the `product_id` to `group_name` mapping. This comes in effect when member purchases a product which is listed in this mapping, the member will also be added to a group mapped to this product.

For example, imagine you have the following in your mapping:

```
1|basic-members
2|premium-members
```

Members purchasing the product with id 1 will be added to the `basic-members` group, those purchasing 2 will be added to the `premium-members` group. For others, nothing will be done.

When switching products, an updated member will be added to new product group, but also kept in the old group. No information is removed/deleted.

TEST IT

You are now ready to do a test buy! Go back to `Sellers` and click `Test Purchases`. Select a product, click `Create Test Purchase Code` and finish by clicking the link in `Buy / Link` column in the table below. In order for the purchase link to work, the product needs to be activated in `Sellers Dashboard` (select a product and check `Allow Sales`).

Before you finish the transaction, you of course need to set up your Plone site to receive JVZoo server notifications.

Confirm by logging-in to <http://jvzoo.com> and checking to see if there were any purchases (on `Sellers` tab). Also check if your received an email with username and password for accessing your site and try to login with them.

KNOWN ISSUES

- If members stop paying for monthly or yearly subscriptions, you have to manually delete them from your Plone site.
- The same as above goes for any chargebacks or refunds. You have to manage them manually.

DEVELOPER DOCUMENTATION

7.1 Tips & Tricks

7.1.1 JVZoo IPN API documentation

Available at <http://support.jvzoo.com/Knowledgebase/Article/View/17/0/jvzipn>.

7.1.2 Mocked request

If you want to mock a request from JVZoo in your local development environment, run something along these lines:

```
$ curl -d "ccustname=JohnSmith&ccuststate=&ccustcc=&ccustemail=test@niteoweb.com&cproditem=1&cproditi
```

7.2 Conventions

7.2.1 Line length

All Python code in this package should be PEP8 valid. However, we don't enforce the 80-char line length rule. It is encouraged to have your code formatted in 80-char lines, but somewhere it's just more readable to break this rule for a few characters. Long and descriptive test method names are a good example of this.

Note: Configuring your editor to display a line at 80th column helps a lot here and saves time.

Note: The line length rules also applies to non-python source files, such as documentation .rst files.

7.2.2 About imports

1. Don't use `*` to import *everything* from a module.
2. Don't use commas to import multiple stuff on a single line.
3. Don't use relative paths.

```
from collective.table.local import add_row
from collective.table.local import delete_rows
from collective.table.local import update_cell
```

instead of

```
from collective.table.local import *
from collective.table.local import add_row, delete_rows
from .local import update_cell
```

7.2.3 Sort imports

As another imports stylistic guide: Imports of code from other modules should always be alphabetically sorted with no empty lines between imports. The only exception to this rule is to keep one empty line between a group of `from x import y` and a group of `import y` imports.

```
from collective.table.tests.base import TableIntegrationTestCase
from plone.app.testing import login
```

```
import os
```

instead of

```
import os

from plone.app.testing import login
from collective.table.tests.base import TableIntegrationTestCase
```

7.2.4 Commit checklist

Before every commit you should:

- Run *Unit tests*.
- Run *Syntax validation*.
- Add an entry to *Changelog* (if applicable).
- Add/modify *Sphinx documentation* (if applicable).

Note: All syntax checks and all tests can be run with a single command:

```
$ ./pre-commit-check.sh
```

7.2.5 Unit tests

Un-tested code is broken code.

For every feature you add to the codebase you must also add tests for it. Also write a test for every bug you fix to ensure it doesn't crop up again in the future.

You run tests like this:

```
$ bin/test -s niteoweb.jvzoo
```

7.2.6 Syntax validation

All Python source code should be *PEP-8* valid and checked for syntax errors. Tools for checking this are *pep8* and *pyflakes*.

To validate your source code, run the following two commands:

```
$ bin/pyflakes src/niteoweb/jvzoo
$ bin/pep8 --ignore=E501 src/niteoweb/jvzoo
$ for pt in `find src/niteoweb/jvzoo/ -name "*.pt"` ; do bin/zptlint $pt; done
```

Note: It pays off to invest a little time to make your editor run *pep8* and *pyflakes* on a file every time you save that file. Saves lots of time in the long run.

7.2.7 Changelog

Feature-level changes to code are tracked inside `docs/HISTORY.txt`. Examples:

- added feature X
- removed Y
- fixed bug Z

Add an entry every time you add/remove a feature, fix a bug, etc.

7.2.8 Sphinx documentation

Un-documented code is broken code.

For every feature you add to the codebase you should also add documentation for it to `docs/`.

After adding/modifying documentation, re-build *Sphinx* and check how it is displayed:

```
$ bin/sphinxbuilder
$ open docs/html/index.html
```

Documentation is automatically generated from these source files every time you push your code to GitHub. The post-commit hook is handled by ReadTheDocs and the results are visible at <http://readthedocs.org/docs/niteowebjvzoo>.

7.3 Releasing a new version

Releasing a new version of *niteoweb.jvzoo* involves the following steps:

1. Create a git tag for the release.
2. Push the git tag upstream to GitHub.
3. Generate a distribution file for the package.
4. Upload the generated package to Python Package Index (PyPI).

7.3.1 Checklist

Before every release make sure that:

1. You have documented your changes in the `HISTORY.rst` file.
2. You have modified the version identifier in the `setup.py` to reflect the new release.
3. You have confirmed that the package description (generated from `README.rst` and others) renders correctly by running `bin/longtest`.
4. You have committed all changes to the git repository and pushed them upstream.
5. You have the working directory checked out at the revision you wish to release.

7.3.2 Actions

For help with releasing we use `jarn.mkreleaser`. It's listed as a dependency in `setup.py` and should already be installed in your local bin:

```
$ bin/mkrelease -d pypi -pq ./
```

Note: In order to push packages to PyPI you need to have the appropriate access rights to the package on PyPI and you need to configure your PyPI credentials in the `~/.pypirc` file, e.g.:

```
[distutils]
index-servers =
  pypi

[pypi]
username = fred
password = secret
```

7.3.3 Example

In the following example we are releasing version 0.1 of `niteoweb.jvzoo`. The package has been prepared so that `version.txt` contains the version 0.1, this change has been committed to git and all changes have been pushed upstream to GitHub:

```
# Check that package description is rendered correctly
$ bin/longtest

# Make a release and upload it to PyPI
$ bin/mkrelease -d pypi -pq ./
Releasing niteoweb.jvzoo 0.1
Tagging niteoweb.jvzoo 0.1
To git@github.com:niteoweb/niteoweb.jvzoo.git
* [new tag]      0.1 -> 0.1
running egg_info
running sdist
warning: sdist: standard file not found: should have one of README, README.txt
running register
Server response (200): OK
running upload
warning: sdist: standard file not found: should have one of README, README.txt
```

Server response (200): OK
done

Note: Please ignore the sdist warning about README file above. PyPI does not depend on it and it's just a bug in setupools (reported and waiting to be fixed).

7.4 Source documentation

7.4.1 Source

Interfaces, Events and Exceptions

interface niteoweb.jvzoo.interfaces.**IJVZooSettings**

Definition of fields for niteoweb.jvzoo configuration form.

secretkey

JVZoo Secret Key

help_secretkey

mapping

Product ID to Group mapping

help_secretkey

exception niteoweb.jvzoo.interfaces.**JVZooError**

Bases: exceptions.Exception

Base class for niteoweb.jvzoo exception.

exception niteoweb.jvzoo.interfaces.**SecretKeyNotSet**

Bases: niteoweb.jvzoo.interfaces.JVZooError

Exception thrown when secret-key for jvzoo is not set.

interface niteoweb.jvzoo.interfaces.**IJVZooEvent**

Base class for niteoweb.jvzoo events.

interface niteoweb.jvzoo.interfaces.**IMemberCreatedEvent**

Extends: niteoweb.jvzoo.interfaces.IJVZooEvent

Interface for MemberCreatedEvent.

context

A member was created by @@jvzoo.

class niteoweb.jvzoo.interfaces.**MemberCreatedEvent** (*context, username*)

Bases: object

Emmited when a new member is created by jvzoo post-back service calling @@jvzoo.

interface niteoweb.jvzoo.interfaces.**IJVZooSettings**

Definition of fields for niteoweb.jvzoo configuration form.

7.4.2 Tests

CHANGELOG

8.1 0.3 (2012-12-03)

- When a member is cancelled, send notification email to site administrator. [matejc]
- If “ctranstime” is empty, set current time. [matejc]
- Support for handling ‘RFND’ requests from JVZoo. [matejc]

8.2 0.2 (2012-11-06)

- Add functionality for auto cancelation and tests. [matejc]

8.3 0.1 (2012-10-16)

- Initial release. [matejc, zupo]

LICENSE (3-CLAUSE BSD)

Copyright (c) 2012, NiteoWeb Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of NiteoWeb Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NITEOWEB LTD. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

n

niteoweb.jvzoo.interfaces, ??