
Zoook e-Sale Documentation

Release 6.1.1.0

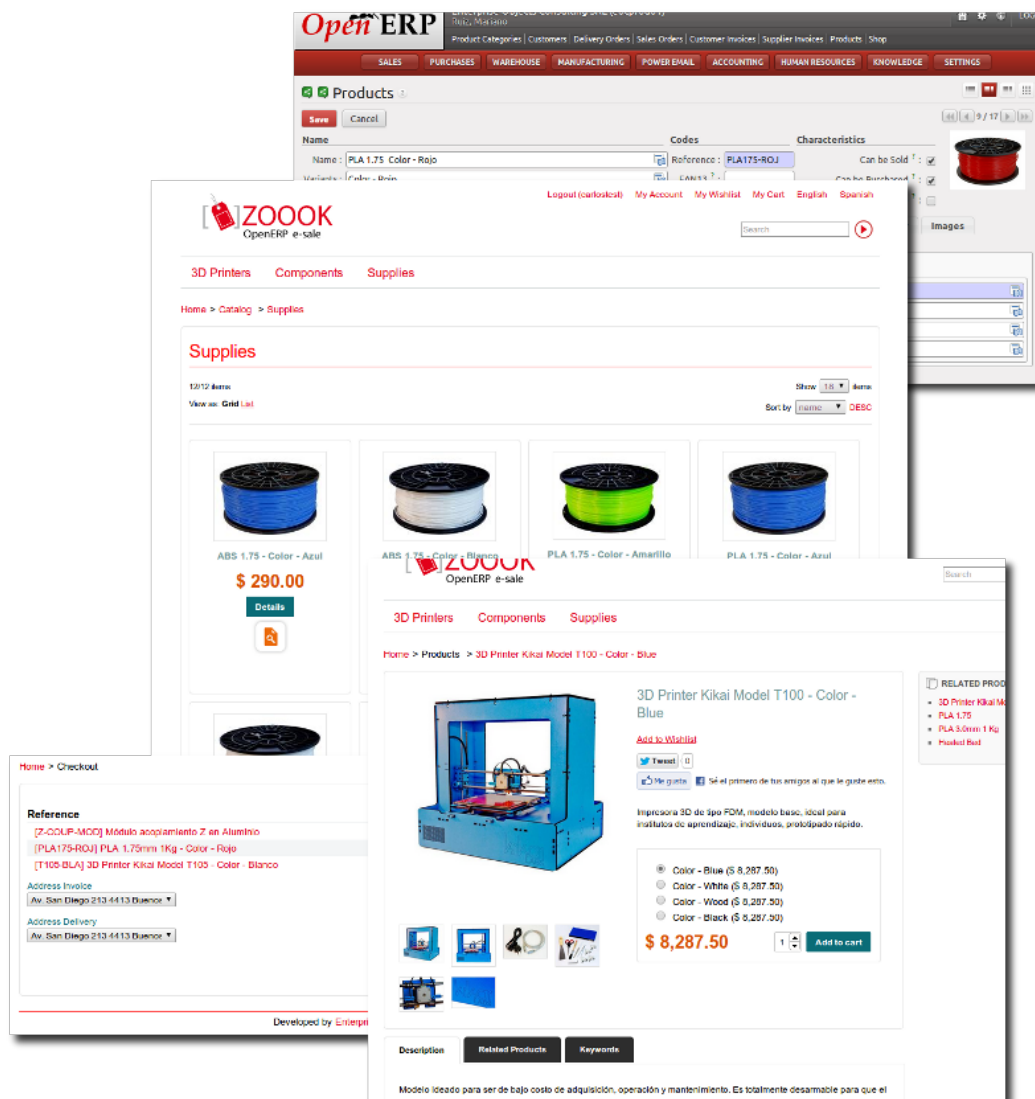
Enterprise Objects Consulting

Jul 31, 2017

Contents

1	Indices and tables	3
1.1	Introduction	3
1.2	Installation	4
1.3	Screenshots	17
1.4	About Zoook e-Sale	25

OpenERP e-commerce system



Zook e-Sale modules, and Django-Zook: *e-commerce* system 100% designed and integrated for OpenERP.

This documentation covers version 6.1.1.0 of Zook.

Note: **Zook e-Sale** was originally developed by *Zikzakmedia*, and currently is unmaintained. This is a **fork** developed by *Enterprise Objects Consulting*, fully functional, with active development, free and open source, and with commercial support too.

Introduction

Zook is e-commerce [OpenERP](#) App **without connector**. Forget connectors and errors.

Zook is a **web store**, developed in **Python** with [Django](#) framework, and is 100% compatible and designed to work with OpenERP.

The orders, the stock, postage and prices (to list just a few features) come straight from OpenERP.

One feature not available a lot of e-commerce is price rules are not same at ERP that e-commerce. You need to create new rules prices at e-commerce app and maintenance two or more rate prices. Same at stock, delivery, promotions... Zook provides and integration e-sale and relation at client (CRM) or any model OpenERP available at web.

Remember, Django is SEO application and products and categories add new features to SEO. Also available cross sells and related products features.

Note: This documentation and the project is work in progress. *Please contribute!*

Architecture

The application is divided into two parts: the *Django Web App*, and the *OpenERP modules* part.

The *Django-Zook* Application part is the webstore itself, and the *modules in OpenERP* is the part allows you to configure countries, payments and delivery available to each sale shop.

You can design payments related with Picking Policy, Order Policy or Invoiced Quantity (it isn't same cash on delivery payment or Paypal payment).

Also the Zook module in OpenERP is used to synchronize the *catalog cache*.

Connection with OpenERP

Zook uses Pyro or XML-RPC protocol and OOP (OpenObject on Python) and Paramiko libraries to connect with OpenERP.

`Client browser <---HTTP---> Zoook e-Sale <---XML/RPC---> OpenERP`

For example, when a customer reserves a product in the shopping cart, Zoook creates in real time a standard *Sale Order* object in *Draft* state in OpenERP. And when the user confirms the order, Zoook moves the state of this order to *In Progress*, and the workflow will be launched a *Packing* to manages the delivery of the goods to the customer.

Catalog cache

Zoook only synchronize as **cache** the catalog information, but all the operations are attended by OpenERP in real time.

The reason than Zoook are synchronized with OpenERP is because if your ERP is off connection, your e-commerce will be still available.

Also, a lot of communication is catalog search (users or search engines), this is an example from an e-commerce store:

Period	Visits	Real Orders
Mar	1744	9
Feb	2227	7
Jun	2167	8

Zoook as Framework

Zoook is not only a shopping cart system, can be used as **development framework** to construct any web application integrated with OpenERP.

The source code has everything you need to start a new project:

- Integrated with OpenERP with webservises (customers, sales, pickings, etc).
- Users in Django App are mapped as partners (customers) in OpenERP.
- Developed with the most popular web framework in Python: [Django](#).
- Easy to modify and extend, compact and legible code.

Installation

This chapter contains the general installation procedure for deploying Zoook e-Sale, and configure an installed OpenERP instance to work with this system.

Requirements

This version of Zoook only works with **OpenERP version 6.1**.

The installation procedure for Zoook under **Linux** is explained in this chapter. The procedure is well tested on *Ubuntu 12.04 LTS*.

Environment tools

The followings tools and packages will be used to install Zoook:

- OpenERP 6.1
- python (2.6+)
- python-setuptools

- python-pip
- python-dev
- libjpeg8-dev
- zlib1g-dev
- bzip
- postgresql
- postgresql-client
- openssh-server

You have to install with `apt-get` tool, or any other you use.

Django-Zoook App sources

To install the Django-Zoook application, you must download them from the repository:
`lp:~zoook-community/zoook-esale/django-zoook-6.1.`

Then install the following dependencies.

Python Packages dependencies

This is the list of Python packages dependencies, they can be installed with `apt-get` tool too, or with `pip` command:

- django == 1.4.x
- django-localeurl
- django-south
- django-pagination
- pycopg2
- python-magic
- paramiko
- django-transmeta
- pillow
- django-inplaceedit
- django-maintenancemode
- Pyro
- OOP (<https://github.com/eoconsulting/ooop>)

In Linux packages systems, the Python packages starts with `python-` prefix, so by example, the `django` package, is named `python-django`.

Optionals:

- recaptcha-client
- django-filebrowser >= 3.5.2
- django-grappelli
- staticfiles_ignoredebug (<https://github.com/eoconsulting/django-staticfiles-ignoredebug>)

OpenERP modules dependency

You have to install the `zoook` module, it's available in the main repository of Zoook e-Sale Project in Launchpad:
`lp:~zoook-community/zoook-esale/zoook-6.1`

Depends modules

The module `zoook` depends of:

- `account_payment_extension`
- `base`
- `base_external_mapping`
- `base_vat`
- `currency_numeric_code`
- `delivery`
- `django`
- `poweremail`
- `product_m2mcategories`
- `product_images_olbs`
- `product_attributes`
- `product_variant_name`
- `sale_multi_shop`
- `sale_payment`
- `sale_promotions`
- `web_wysiwyg`
- `web_display_html`
- `partner_title_pricelist`

Optional:

- `nan_product_pack`

All this modules are in the dependencies branch repository of Zoook e-Sale Project in Launchpad:
`lp:~zoook-community/zoook-esale/zoook-extra-addons-6.1`

Configuration

In this section will be describe how to configure Zoook and OpenERP.

Having finished the dependencies installation in OpenERP and Django, we will proceed with their parameterization and the connection between the boot components.

Django-Zoook Configuration

In this section will be configure the web shop part of Zoook.

Note: Zoook web shop is a *Django* project. Read first about Django framework. The official docs is a great documentation: <https://docs.djangoproject.com/en/1.4>

Checkout and install requeriments

In the *Requirements guide* are listed what you need, you must be install all, following the standard steps for each type of packages.

Installing the Zoook sources

To download the latest version of *Django-Zoook*, go to any folder you want to save the repositories, and checkout it with *Bazaar*:

```
$ bzr branch lp:~zoook-community/zoook-esale/django-zoook-6.1 django-zoook
```

This will download all the source code in a new folder *django-zoook*.

Create Zoook Database

To manage users and databases in **PostgreSQL** will be use **PSQL** command line tool, but you can use other tools like the **PgAdmin** GUI, or the web tool **phpPgAdmin**.

To execute PSQL interpreter as `postgres` user (the **root** user in PostgreSQL):

```
$ sudo -u postgres psql
```

The system will ask your password, and then you will be at the PostgreSQL command line prompt:

```
postgres=#
```

To exit the psql:

```
postgres=# \q
```

Create PostgreSQL user

At first create a PostgreSQL database for Zoook. In a development environment you can use the user `openerp`. This way you don't have to create a new one. In production its recommended to make sure that every user has an exclusive access to the database.

The command to create a PostgreSQL user:

```
$ sudo -u postgres createuser --createdb --no-superuser --no-createrole --pwprompt_
↪ zoook
```

Create the BBDD

This is mandatory, the Zoook database stores all the catalog cache, the registered users (mapped as client in OpenERP), and others configuration tables.

First enter with `psql` command, and then create the database with:

```
postgres=# CREATE DATABASE dj_zoook OWNER zoook;
```

To verify the correct creation of the database, exit with `\q` command, and then enter with:

```
$ sudo -u postgres psql dj_zoook -U zoook
Password for user zoook:
psql (9.1.9)
Type "help" for help.
dj_zoook=>
```

Parameterization Django-Zoook

Django-Zoook is a Django application, so many parameters are standard of this framework.

Configuration file: config.py

Edit the file and adjust the parameters of the connection to OpenERP. You'll have to configure all passages. The following ones are especially important for the interconnection with the OpenERP server.

DATABASES: Defines where and how Zoook finds its database. If you are installing the environment in your local computer, verify the NAME, USER, and PASSWORD parameters:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'dj_zoook',
        'USER': 'zoook',
        'PASSWORD': 'postgres',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

OERP_CONF: Defines the connection to the XML-RPC service of OpenERP. In a local environment, check the first three parameters.

```
OERP_CONF = {
    'username': 'admin',
    'password': 'admin',
    'dbname': 'openerp',
    'protocol': 'xmlrpc', #xmlrpc
    'uri': 'http://localhost', #xmlrpc
    'port': 8069, #xmlrpc
    # 'protocol': 'pyro', #pyro
    # 'uri': 'localhost', #pyro
    # 'port': 8071, #pyro
}
```

Locale: Defines the languages of your web shop. They have to be the same as in OpenERP:

```
LANGUAGE_CODE = 'es'
LANGUAGES = (
    ('en', ugettext('English')),
    ('es', ugettext('Spanish')),
)
DEFAULT_LANGUAGE = 1
LOCALE_URI = True
LOCALEURL_USE_ACCEPT_LANGUAGE = True
LOCALES = {
    'en': 'en_US',
    'es': 'es_ES',
}
```

OERP_SALE: Indicates which OpenERP shop must be used in Zoook.

```
OERP_SALE = 2 #Sale Shop. All price, orders, ... use this Sale Shop ID.  
OERP_SALES = [2] #Sale Shops. Orders by Sale Shops
```

Email: Emails notification are parsed by *Poweremail* in OpenERP, but are sent by Zoook with the Django framework support. This is an example with a Gmail connection:

```
EMAIL_USE_TLS = True  
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_HOST_USER = 'myemailuserXXX'  
EMAIL_HOST_PASSWORD = 'mypasswordXXX'  
EMAIL_PORT = 587
```

Log configuration file: logconfig.py

Zoook uses the standard **logging** library in Python. If you want to verify the log setup, check the `logconfig.py` file.

By default the log output are located in `../django-zoook/log/zoook.log` file. Depending of the user's system and file system configuration, you will need to create manually the container directory ("**log**" folder), and with the necessary permissions to get read/write access.

Note: When Zoook starts if you get a `IOError` in the output console related with the `zoook.log` file, and the process is aborted, then you don't have the permissions to access the file, or the file cannot be created because the container "**log**" folder don't exist.

Create Zoook's database tables

To generate the database structure, go to the `django-zoook/django_zoook` folder with the command line. Then run:

```
$ python manage.py syncdb
```

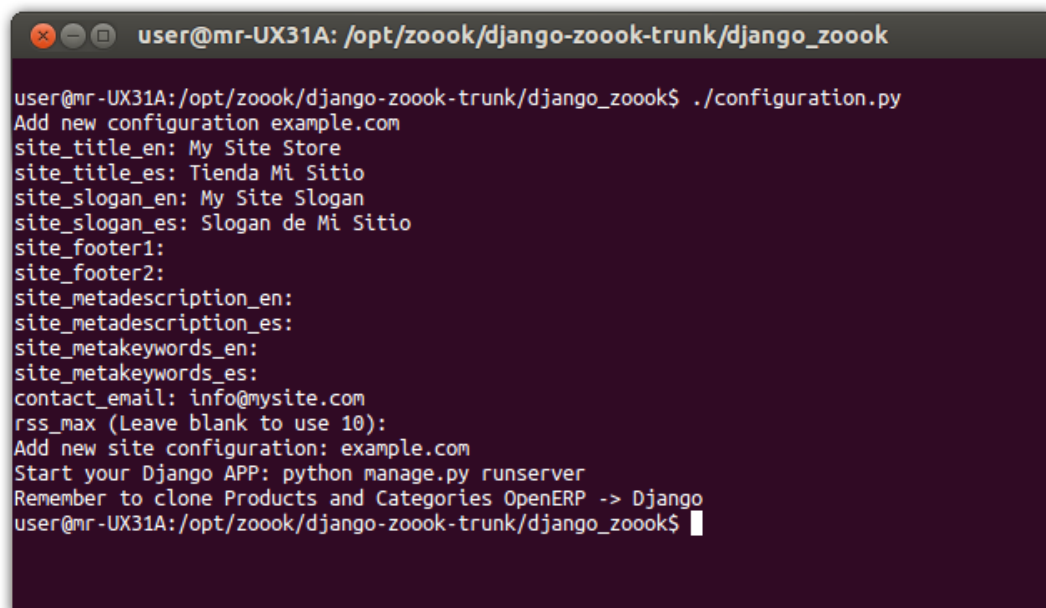
The script will ask if you want to create a new super-user. Affirm and pass it the needed data for the account you'll later want to administer Zoook with.

Create Zoook's configuration

To generate the initial configuration, go to the `django-zoook/django_zoook` folder with the command line. Then run:

```
$ ./configuration.py
```

The script will ask you some basic information, like the store name, slogan, etc.

A terminal window with a dark purple background and a title bar that reads "user@mr-UX31A: /opt/zoook/django-zoook-trunk/django_zoook". The terminal shows the execution of a configuration script. The prompt is "user@mr-UX31A:/opt/zoook/django-zoook-trunk/django_zoook\$". The first command is "./configuration.py". The script then prompts for various site settings, which are entered as follows: "Add new configuration example.com", "site_title_en: My Site Store", "site_title_es: Tienda Mi Sitio", "site_slogan_en: My Site Slogan", "site_slogan_es: Slogan de Mi Sitio", "site_footer1:", "site_footer2:", "site_metadescription_en:", "site_metadescription_es:", "site_metakeywords_en:", "site_metakeywords_es:", "contact_email: info@mysite.com", and "rss_max (Leave blank to use 10):". After these prompts, the script outputs "Add new site configuration: example.com" and "Start your Django APP: python manage.py runserver". It then displays a reminder: "Remember to clone Products and Categories OpenERP -> Django". Finally, the prompt returns to "user@mr-UX31A:/opt/zoook/django-zoook-trunk/django_zoook\$".

```
user@mr-UX31A: /opt/zoook/django-zoook-trunk/django_zoook
user@mr-UX31A:/opt/zoook/django-zoook-trunk/django_zoook$ ./configuration.py
Add new configuration example.com
site_title_en: My Site Store
site_title_es: Tienda Mi Sitio
site_slogan_en: My Site Slogan
site_slogan_es: Slogan de Mi Sitio
site_footer1:
site_footer2:
site_metadescription_en:
site_metadescription_es:
site_metakeywords_en:
site_metakeywords_es:
contact_email: info@mysite.com
rss_max (Leave blank to use 10):
Add new site configuration: example.com
Start your Django APP: python manage.py runserver
Remember to clone Products and Categories OpenERP -> Django
user@mr-UX31A:/opt/zoook/django-zoook-trunk/django_zoook$
```

Run !


Now you can run Zoook. Use the usual way in a normal Django App:

```
$ python manage.py runserver
```


The application will run in the port number 8000, and will accept only local connections. To get more options of how to start a Django application go to the official documentation.

To view the page open a web browser and type <http://localhost:8000> in the URL bar and press enter.

The application will look like this:



[Entrar](#)
[Inglés](#)
[Español](#)



Welcome Zoook - OpenERP e-sale

You are at home of *Django Zoook*.

OpenERP Configuration

Edit **config.py**. Configure OpenERP connections (XML-RPC or PyRO).

Home Page

This is the **doc** home page. Change to the **Index** page (with "Featured Products") in `django_zoook/urls.py`.
Edit tpl index for customization frontend and custom your template.

Django configuration

1. Add in **config.py** your configuration (database, OpenERP connection, sale,).
2. py files in sync dir: change execute permissions.
3. Edit **config_path.py**. Add you Django path.
4. Add same languages OpenERP in **settings.py** (LANGUAGES).
5. Create tables: `python manage.py syncdb`
6. Important! Run **configuration.py**. Add new values site django (extra fields).
7. Run Server: `python manage.py runserver`

OpenERP configuration

1. *Sale/Administration/Sales*. Configure your Sale Shop and paramiko connection (ssh).
2. Add Countries to Check Vat. This option is required because is same countries available shop.
3. Sale Order or Invoices sequence not use / (for example, SO/001). URL navigator get order number and / it's a path url and broken.

Catalog and Product Sync

Sales, partners, ... are creating in real time e-sale to OpenERP. Only need sync categories and products. Go to sale shop OpenERP for sync. Remember Django Sync dir need execute permissions. Also, remember see OpenERP logs and Django logs.

Attention! Manager catalog section is only check values sync (product, categories,...) Not change this values with manager tool. Use Sync Tools.

Django Template

See more documentation about Django template at [Django Templates Doc](#).
Add new modules in your template: `{% module [position]%}`

Active Payments

Default configuration not available payments active. How to active payments?

1. Go to OpenERP *Sale->Shop*. In Payment tab, add payments active in this shop.
2. Edit **settings.py**. Add payment APP available (see comment app example)
3. Edit **payment/urlsPayment.py**. Add urls payment type available (see comment urls example)
4. Edit **config.py** if you payment type need configuration. There are some examples configuration params *Sermepa* and *Paypal*.

Paypal or Served payments, need install in your server [django-paypal](#) and [django-sermepa](#).

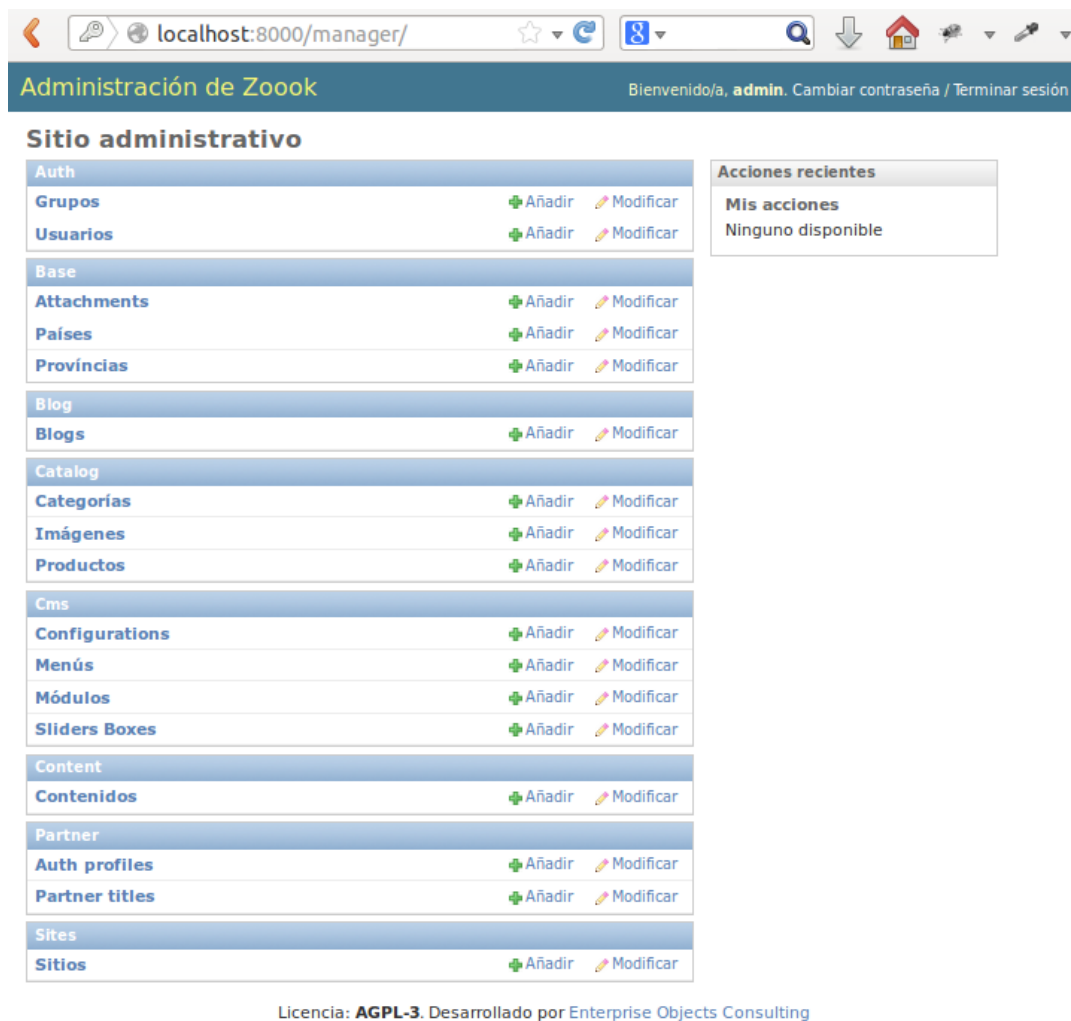
Catalog Admin

Super Users are available access to Catalog Admin. This section is only check catalog database sync OpenERP -> Django. Create your catalog in OpenERP.

[Go to Admin](#)
[Go to documentation and code](#)

Desarrollado por [Enterprise Objects Consulting](#)

Also Zoook has an Administrator page (*Django Admin Site* module), you can enter with the `/manager` URL, then the username and password of the *super-user* will asked to access the page:



Sitio administrativo

Auth	
Grupos	Añadir Modificar
Usuarios	Añadir Modificar

Base	
Attachments	Añadir Modificar
Países	Añadir Modificar
Provincias	Añadir Modificar

Blog	
Blogs	Añadir Modificar

Catalog	
Categorías	Añadir Modificar
Imágenes	Añadir Modificar
Productos	Añadir Modificar

Cms	
Configurations	Añadir Modificar
Menús	Añadir Modificar
Módulos	Añadir Modificar
Sliders Boxes	Añadir Modificar

Content	
Contenidos	Añadir Modificar

Partner	
Auth profiles	Añadir Modificar
Partner titles	Añadir Modificar

Sites	
Sitios	Añadir Modificar

Acciones recientes

Mis acciones

Ninguno disponible

Licencia: **AGPL-3**. Desarrollado por Enterprise Objects Consulting

Zoook OpenERP module Configuration

In this section will configure the module of Zoook in our OpenERP instance.

Checkout and install requirements

In the *Requirements guide* are listed what you need, you must be install all following the standard steps for each type of packages.

Installing the Zoook's sources

To checkout the OpenERP modules and depends, go to any folder you want to store the repositories, and execute this:

```
$ bzr branch lp:~zoook-community/zoook-esale/zoook-6.1 zoook
$ bzr branch lp:~zoook-community/zoook-esale/zoook-extra-addons-6.1 zoook-extra-
↪ addons
```

You must be copy all the modules in these folders into the addons folder of OpenERP, or add the paths of these branches into the OpenERP file configuration. Go to [OpenERP Documentation](#) to get more information about the configuration file.

Installing the Zoook's modules in OpenERP

Go to *Settings > Modules > Update Modules List* after copy or linking the modules into the addons path.

Then go to *Settings > Modules > Modules*, and search and install the module **zoook**: will be installed with all dependencies.

Create a Shop

Zoook is multi-shop, but not multitennancy: that means than you can have more than one shop configured in OpenERP, each with its own products, categories, and all shop in the same OpenERP database (and therefore the same instance), but in the Django Apps side, you must be have one instance by shop, each instance with its own Zoook database, configuration and Python installation (and each running in a different process and HTTP port).

To create a shop go to *Sales > Configuration > Sales > Shop*, by default one shop exist with the name “*Shop 1*”, you can use this, or create another. The shops objects are not only related to Zoook, you can configure several shops to reflect your business infrastructure. Then when you sale something in the *Sale Order* you can select in which shop you are. And when a customer buys something from Zoook, the shop configured for this Zoook instance will be appear in the order.

So to setup a Zoook shop in an existing shop or a new one, in the edit form check the “*OpenERP e-Sale*” field. There will be new sections in the form, related to Zoook.

Note: When you edit a register in OpenERP in most cases the ID in the database is not visible, but if you see in the URL, the parameter “*id*” has this ID. You have to get this shop ID to configure the **OERP_SALE** and **OERP_SALES** parameters of the *Django-Zoook configuration*.

Configuring the Shop

You must be fill all the mandatory fields before save the shop. These are the important fields:

Root product Category: The root category of all the sub-categories will be displayed in the store. You must be create or select one (also with the “*OpenERP e-Sale*” check activated). Zoook support n-levels of sub-categories.

Email Sale Order: The *Poweremail* template will be used to render the confirmation email of a Sale Order. To create the template, you can use the *example* in the source code of zoook module.

Languages: Zoook is multi-language, you can configure the same languages of your OpenERP database in Zoook, and show your products in the store in multiples locales.

Delivery: Deliveries methods. You must be have one at least. If you don't have delivery service, you can configure a “*nodelivery*” method, with a related delivery pricelist with zero cost price.

Payment: Supported customer payment methods. Can be virtual (like *PayPal*), or not virtual like the “*cashon-delivery*” method. You must be have one at least. Each payment method has a related application to handle the payment transaction. Check the supported methods by the Web Apps in the `django_zoook.payment` package.

SSH Connection: The SSH parameters to connect with Django-Zoook application. Fill this with the standard parameters needed in this type of connection. Check the following section about the SSH server and the necessary steps to get a successfully connection.

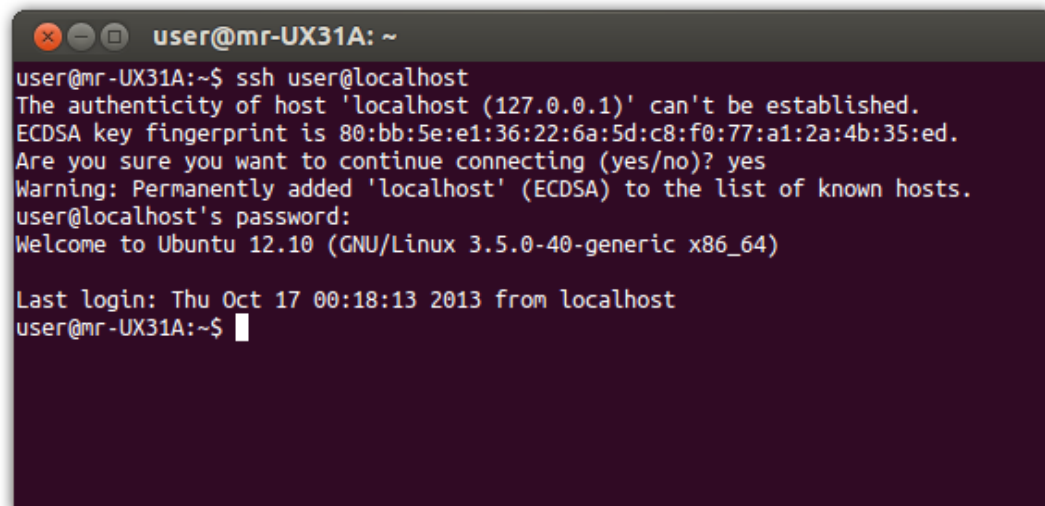
Configuring SSH Connection

First, you must be installed and running a SSH server (like *OpenSSH*) in the same host where Django-Zoook are. If it's not the same host than the OpenERP server, you must be procure the access to this SSH server in Django-Zoook to OpenERP server (check the firewall configuration).

Also you must be import the SSH certificate in the “*know hosts*” list of consumer host, this can be done when you try to connect from the console: if the SSH client don’t recognize the certificate (it’s not a public key), will be ask you to import the key, or abort the connection.

The import of the certificate must be achieved even if the Django-Zoook daemon and the OpenERP daemon reside in the same host.

This is an example of connecting to your *localhost* through SSH command line tool, and import the certificate:

A terminal window titled 'user@mr-UX31A: ~' with a dark background and light text. The user enters 'ssh user@localhost'. The terminal shows the SSH warning about the authenticity of the host 'localhost (127.0.0.1)' and the ECDSA key fingerprint. The user responds 'yes' to continue. A warning message states 'Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.' The user is then prompted for a password, which is masked with dots. After the password is entered, the terminal displays 'Welcome to Ubuntu 12.10 (GNU/Linux 3.5.0-40-generic x86_64)' and the last login information: 'Last login: Thu Oct 17 00:18:13 2013 from localhost'. The prompt returns to 'user@mr-UX31A:~\$'.

If you don’t perform this step, then Zoook module won’t connect with Django-Zoook. After these steps, in OpenERP go to the shop configuration, and in the “*SSH Connection*” click the “*Test Connection*” button to check if the configuration are OK.

Product Configurations

After configure the Shop, you have to configure some categories (in the three of the root category), and configure the products. Each product and category must be checked in the “*OpenERP e-Sale*” field to be exported. In the product form, the categories must be configured in the “*Extra Categories*” tab (the “*Category*” field is not used by Zoook).


Also **remember**, the shop where the product will be exported must be configured in the section “*Website*” of the tab “*e-Sale*”, remember than Zoook is multi-shop and the shop must be specified, or will not be exported.

Note: You can attach many images for each product, but for the moment Zoook only supports *Links* pictures. The pics stored in the database (uploaded) will be exported incorrectly. The link’s images are exported well, and resized and cached by Zoook in the `django_zoook/static/catalog` folder.

Export the catalog

Once finished the initial catalog setup, you have to export all the data to Zoook webshop. To do the task go to *Sales > Configuration > Sales > Shop* menu, and enter to the shop to synchronize. In “*Actions*” tab in the bottom section, click in the buttons to export, in the same order are showed, depending of the data you want to export. If it’s the first export, click in the “*Global configuration*” before the other export tasks.

After export the catalog, a category page will look like this:


Logout (carlorest) My Account My Wishlist My Cart English Spanish

[3D Printers](#)
[Components](#)
[Supplies](#)

[Home](#) > [Catalog](#) > [Supplies](#)

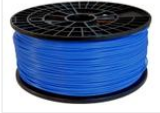











Supplies

12/12 items

View as: [Grid](#) [List](#)

Show items

Sort by

 <p>ABS 1.75 - Color - Azul</p> <p>\$ 290.00</p> <p>Details</p> 	 <p>ABS 1.75 - Color - Blanco</p> <p>\$ 290.00</p> <p>Details</p> 	 <p>PLA 1.75 - Color - Amarillo</p> <p>\$ 290.00</p> <p>Details</p> 	 <p>PLA 1.75 - Color - Azul</p> <p>\$ 290.00</p> <p>Details</p> 
			

Note: When you export data, only the newest information are exported according to the last exportation date. If the process fails, the date will be updated anyway, and you won't export the information again, at least you first edit, or modify manually the last exportation field.

Also the delete operations are not exported, so if you delete a product, you must be delete from the *Django Admin Site* of Zoook (/manager URL), or you can edit the "Active" field or the "Visibility" field of the product to deactivate.

Configuring Poweremail

This is the OpenERP module responsible to send emails when a client confirms an order, or when a user is registered or changes their password.

A good guide of this module is [here](#).

Note: Install **poweremail** module only from `lp:~zoook-community/zoook-esale/zoook-extra-addons-6.1`, this version has some important fixes to work with Zoook.

Email accounts

Go to *Poweremail > Configuration > All accounts*, and create a new account, the following example is a SMTP Outgoing account in Gmail:

- **Name:** A name or brief description
- **Server:** smtp.gmail.com
- **SMTP Port:** 587

- **Use SSL/TLS:** False
- **Use TLS:** True
- **Email ID:** `xxx@gmail.com` (*Gmail account id*)
- **User Name:** `xxx@gmail.com`
- **Password:** *Gmail password*
- **Related Name:** *The OpenERP user owner of this account, the same used in Zoook*
- **Company Mail A/c:** No

This account will be used to send email notifications when you add a Zoook user to an existing client in OpenERP (Django tab in partner form), or when you reset also from OpenERP the Zoook user password.

Email configuration in Zoook

The emails sent by the shop are processed by Poweremail, but dispatched by Zoook with the Django API. So, you have to configure the email account in Django Zoook App too, in the [config.py](#) file.

Templates

The process of defining templates is used to configure email for a specific model, like the *Sale Order* object, or the *Reset Password* wizard object.

To create a new template go to *Poweremail > Templates > Email Templates*, and click in *Create* button.

In the folder [util/poweremail-templates-examples](#) of **zoook** module you will find the necessary templates for each type of object where Zoook launches emails, you can use a custom text/HTML for this emails, or send to the clipboard the content of this examples and paste into the OpenERP form, in the *Body (HTML)* field. This is an example of template for **Sales Order**:

- **Body (HTML):** *The content of sale.order-email-en.html example template*
- **Model:** *Sales Order (sale.order)*
- **Receipient (To):** `${object.partner_id.dj_email or object.partner_order_id.email}`
- **Default BCC:** *A email to receive the Sale Order notifications, like myemail@company.com (optional)*
- **Default Subject:** `My Shop - Sale Order Ref ${object.name} - ${object.shop_id.name or object.company_id.name}`
- **Language:** `${object.partner_id.lang}`
- **Templating Language:** *Mako Templates*

After create this template, you have to setup in the [shop configuration field](#).

Example for **Create User** wizard:

- **Body (HTML):** *The content of zoook.create.user.wizard-email-en.html example template*
- **Model:** *zoook.create.user.wizard*
- **Receipient (To):** `${object.email}`
- **Default Subject:** *My Shop - New User*
- **Templating Language:** *Mako Templates*

Example for **Reset User (reset password create)** wizard:

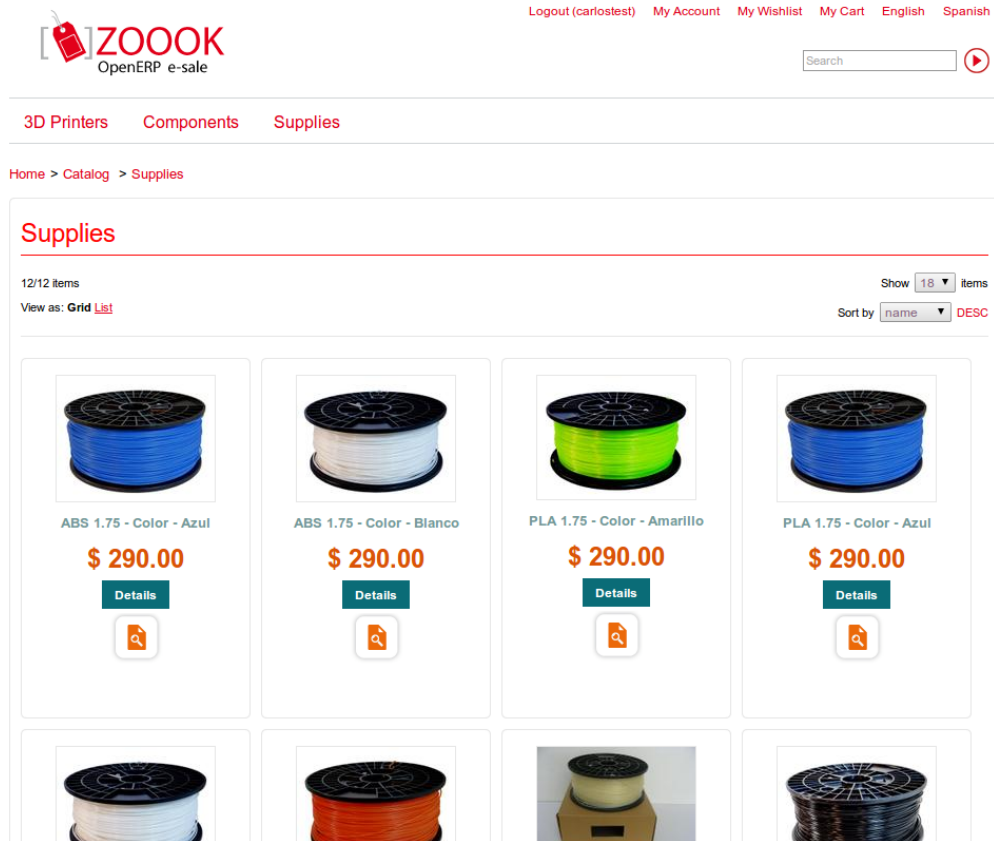
- **Body (HTML):** *The content of zoook.reset.user.wizard-email-en.html example template*
- **Model:** *zoook.reset.user.wizard*

- **Receipient (To):** `${object.partner_id.dj_email}`
- **Default Subject:** My Shop - Reset User Password
- **Templating Language:** *Mako Templates*

Screenshots


Catalog list

Catalog list, viewing a category in grid mode.




Product details

Product information and pictures.

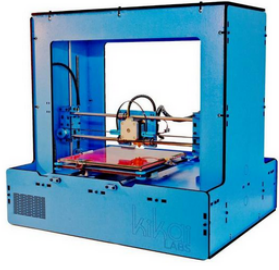


[Logout \(carlorest\)](#)
[My Account](#)
[My Wishlist](#)
[My Cart](#)
[English](#)
[Spanish](#)




[3D Printers](#)
[Components](#)
[Supplies](#)



[Home](#) > [Products](#) > [3D Printer Kikai Model T100 - Color - Blue](#)



3D Printer Kikai Model T100 - Color - Blue

[Add to Wishlist](#)

 [Tweet](#) 0







 [Me gusta](#)
 Sé el primero de tus amigos al que le guste esto.

Impresora 3D de tipo FDM, modelo base, ideal para institutos de aprendizaje, individuos, prototipado rápido.

☒ Color - Blue (\$ 8,287.50)
 ☐ Color - White (\$ 8,287.50)
 ☐ Color - Wood (\$ 8,287.50)
 ☐ Color - Black (\$ 8,287.50)

\$ 8,287.50

[Add to cart](#)

[Description](#)
[Related Products](#)
[Keywords](#)


Modelo ideado para ser de bajo costo de adquisición, operación y mantenimiento. Es totalmente desarmable para que el usuario pueda efectuar reparaciones por su cuenta. Carcasa de madera MDF. Actualizable a modelo con doble extrusor.

RELATED PRODUCTS


- 3D Printer Kikai Model T105
- PLA 1.75
- PLA 3.0mm 1 Kg
- Heated Bed

Product gallery

Product gallery plugin.

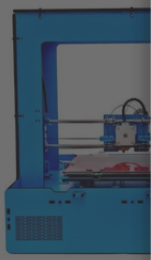


[Logout \(carlorest\)](#)
[My Account](#)
[My Wishlist](#)
[My Cart](#)
[English](#)
[Spanish](#)




[3D Printers](#)
[Components](#)
[Supplies](#)



[Home](#) > [Products](#) > [3D Printer Kikai Model T100 - Color - Blue](#)



3D Printer Kikai Model T100 - Color - Blue

[Add to Wishlist](#)

 [Tweet](#) 0







 [Me gusta](#)
 Sé el primero de tus amigos al que le guste esto.

Impresora 3D de tipo FDM, modelo base, ideal para institutos de aprendizaje, individuos, prototipado rápido.

☒ Color - Blue (\$ 8,287.50)
 ☐ Color - White (\$ 8,287.50)
 ☐ Color - Wood (\$ 8,287.50)
 ☐ Color - Black (\$ 8,287.50)

\$ 8,287.50

[Add to cart](#)

[Description](#)
[Related Products](#)
[Keywords](#)

Modelo ideado para ser de bajo costo de adquisición, operación y mantenimiento. Es totalmente desarmable para que el usuario pueda efectuar reparaciones por su cuenta. Carcasa de madera MDF. Actualizable a modelo con doble extrusor.

RELATED PRODUCTS

- 3D Printer Kikai Model T105
- PLA 1.75
- PLA 3.0mm 1 Kg
- Heated Bed




Kit de herramientas incluido

Pictures of this product are provided as a guide only. Colour and aspect may vary from the actual stock.

Wishlist



Customer products like.



[3D Printers](#)
[Components](#)
[Supplies](#)

[Home](#) > [Wishlist](#)


[Orders](#)
[Invoices](#)
[Your account](#)
[My Wishlist](#)
[Change password](#)
[Logout](#)

Product Name	Description	Price
 Fuente PC ATX 500W 24 PINS SA	Fuente de alimentación ATX 500W SATA B 24P.	\$ 120.00
 LM8UU Holder	Ruleman lineal LM8UU	\$ 28.00

Developed by [Enterprise Objects Consulting](#)

Header menu

Multi-level menu.



[My Account](#)
[My Cart](#)
[Identificarse](#)

[Electrodomésticos](#)
[Cocina](#)
[Aire acondicionado y calefacción](#)
[Descalcificadores y osmosis](#)
[Vinos](#)

Submenú llarg	Submenú 1	Submenú 2	Submenú 3	Submenú 4
> Lorem ipsum dolor sit	> Menú opció 1	> Menú opció 2	> Menú opció 1	> Menú opció 1
> Aliquam	> Menú opció 2	> Menú opció 3	> Menú opció 2	> Menú opció 2
> Morbi	> Menú opció 3	> Menú opció 4	> Menú opció 3	
> Praesent	> Menú opció 4	> Menú opció 5		Submenú 5
> Pellentesque	> Menú opció 5	> Sumsubmenú 1		> Menú opció 1
		> Menú opció 1		> Menú opció 2
		> Menú opció 2		> Menú opció 3
		> Menú opció 3		
		> Menú opció 2	Submenú 7	
		> Menú opció 3	> Menú opció 1	
		> Menú opció 2	> Menú opció 2	
		> Menú opció 3		

Checkout

Shopping cart confirmation.

[Home](#) > [Checkout](#)

Reference	Qty	Discount	Unit Price	Subtotal	Delete
[Z-COUP-MOD] Módulo acoplamiento Z en Aluminio	2.0	0.0	55.0	99.55	
[PLA175-ROJ] PLA 1.75mm 1Kg - Color - Rojo	1.0	0.0	290.0	239.67	
[T105-BLA] 3D Printer Kikai Model T105 - Color - Blanco	1.0	0.0	9,668.75	8,750.0	

Address Invoice
Av. San Diego 213 4413 Buenos

Address Delivery
Av. San Diego 213 4413 Buenos

Total: \$ 10,068.75
Delivery: Retira en local (+0.0)
Payment: Efectivo
Code Promotion:

Confirm

Developed by [Enterprise Objects Consulting](#)

OpenERP product setup

Configuring the product SEO.

The screenshot shows the OpenERP product setup interface. The top navigation bar includes links for Product Categories, Customers, Delivery Orders, Sales Orders, Customer Invoices, Supplier Invoices, Products, and Shop. The main menu has tabs for SALES, PURCHASES, WAREHOUSE, MANUFACTURING, POWER EMAIL, ACCOUNTING, HUMAN RESOURCES, KNOWLEDGE, and SETTINGS. The 'Products' section is active, showing a list of products. The 'e-Sale' tab is selected, and the 'SEO' sub-tab is active. The product 'PLA 1.75 Color - Rojo' is selected, and its SEO configuration is displayed. The 'General' tab is active, showing the 'Slug' as 'pla-175', 'Meta Description' as 'Rollo PLA para impresoras RepRap', 'Meta Keyword' as 'PLA', and 'Meta Title' as an empty field. The 'Codes' section shows the 'Reference' as 'PLA175-ROJ' and the 'EAN13' as an empty field. The 'Characteristics' section shows 'Can be Sold' as checked, 'Can be Purchased' as checked, and 'Can Constitute an Expense' as unchecked. A small image of a red spool of PLA filament is shown on the right.

OpenERP shop options

Shop export and setup options.

The screenshot shows the OpenERP e-sale shop options interface. The top navigation bar includes links for Actions, Configuration, Languages, Delivery, Payment, Partner Titles, and Country. The 'SSH Connection' tab is active, showing a list of export options. The 'Export Categories' button is highlighted, and the 'Last Export Categories' is shown as '08/08/2013 17:00:49'. The 'Export Products' button is highlighted, and the 'Last Export Products' is shown as '09/30/2013 14:36:16'. The 'Export Images' button is highlighted, and the 'Last Export Images' is shown as '09/28/2013 17:24:07'. The 'Global Configuration' button is also visible.


Zoook login

User can login or create an account.


The screenshot shows the Zoook login page. At the top, there are links for 3D Printers, Components, Supplies, and Accessories. Below these links is a section titled 'Log in or create an account.' which contains two main options: 'New customers' and 'Registered customers'. The 'New customers' section includes a message: 'If you create an account in our store, you could buy and access the orders quickly.' and a 'Create new account' button. The 'Registered customers' section includes a message: 'If you have an account with us, please log in.' and a login form with fields for 'Username' and 'Password', a 'Login' button, and a 'Remember Password' checkbox. A note at the bottom indicates that '*' denotes required fields.

Zoook Customer Orders

Users can see their history orders.




[Logout \(educatest1\)](#)
[My Account](#)
[My Wishlist](#)
[My Cart](#)
[English](#)
[Spanish](#)



[3D Printers](#)
[Components](#)
[Supplies](#)
[Accessories](#)

[Home](#) > [Orders](#)

[Orders](#)
[Invoices](#)
[Your account](#)
[My Wishlist](#)
[Change password](#)
[Logout](#)

Reference	Date	Date Confirm	Total	State	
SO201	Oct. 4, 2013	Pending	31.5	Draft	
SO200	Oct. 4, 2013	Oct. 4, 2013	270.0	Progress	
SO199	Oct. 4, 2013	Oct. 4, 2013	13,455.0	Progress	
SO198	Oct. 4, 2013	Oct. 4, 2013	144.0	Done	
SO197	Oct. 4, 2013	Oct. 4, 2013	3,721.5	Progress	

Developed by [Enterprise Objects Consulting](#)

Zoook Customer Order

Users can see the details of their history orders.

[Home](#) > [Orders](#) > Order SO197

[Orders](#)
[Invoices](#)
[Your account](#)
[My Wishlist](#)
[Change password](#)
[Logout](#)

Date: Oct. 4, 2013
Date Confirm: Oct. 4, 2013
State: Progress
Delivery: Retira en local

Address Invoice
 Dir Establecimiento 1123
 5555 CABA

Address Delivery
 Dir Establecimiento 1123
 5555 CABA

Reference	Qty	Unit Price	Discount %	Subtotal
[SKUP-PRO] SketchUp Pro 8	1.0	3,300.0	10.0	2,454.55
[TOOLS-MAK] Kit de Herramientas Maker	3.0	85.0	10.0	189.67
> Cartuchera para herramientas	3.0	0.0		0.0
> Espátula	3.0	0.0		0.0
> Tijera	3.0	0.0		0.0
> Pincel	3.0	0.0		0.0
> Lima	3.0	0.0		0.0
> Pinza de Bruselas	3.0	0.0		0.0
> Llave Allen para M3	3.0	0.0		0.0
> Destornillador de cerámica	3.0	0.0		0.0
> Clip Metálico	12.0	0.0		0.0
[PLA-BLA] PLA 1.75mm 1Kg - Color - Blanco	2.0	290.0	10.0	431.4
Total: 3,721.5				

Product Price Update

Users can see the price of the products according with their price list.



modelos 3D, que luego se pueden materializar con las impresoras 3D de Kikai

Code: SKUP-PRO

~~\$ 3,300.00~~

\$ 2,970.00

1

Add to cart

Product Variants

Products can have variants like color, size, etc.

presora 3D Modelo T125 - Color - Madera



Impresora 3D Modelo T125 - Color - Madera

Me gusta

Sé el primero de tus amigos al que le guste esto.

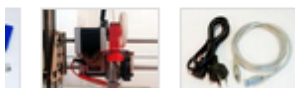
Impresora 3D de tipo FDM, con un extrusor de 1.75mm, boquilla de 0,4mm y plataforma calefaccionada MK2.

- ☐ Color - Negro (\$ 14.950,00)
- ☐ Color - Azul (\$ 14.950,00)
- ☒ Color - Madera (\$ 14.950,00)
- ☐ Color - Blanco (\$ 14.950,00)

\$ 14.950,00

1

Agregar al carrito



Registration form

Users can register in the web shop.



[3D Printers](#) [Components](#) [Supplies](#) [Accessories](#)

Create an account.

New customers

User Information

Username *	<input type="text"/>	(az09)
Email *	<input type="text"/>	
Password *	<input type="password"/>	
Password confirmation *	<input type="password"/>	

Customer Information

Full Name *	<input type="text"/>
Country *	<input type="text" value="Argentina"/>
You are ?	<input type="text"/>
Vat	<input type="text"/>

Invoice Address

Address *	<input type="text"/>
ZIP *	<input type="text"/>
City *	<input type="text"/>
Phone	<input type="text"/>
Same address for delivery	<input checked="" type="checkbox"/>

Create new account

* Required Fields

User Profile

Users can edit their profile information.



[3D Printers](#) [Components](#) [Supplies](#) [Accessories](#)

[Orders](#) [Invoices](#) [Your account](#) [My Wishlist](#) [Change password](#) [Logout](#)

User Profile

Customer Information

Full Name	Educ Test 1
Email	eductest1@mail.com
Vat	
You are ?	Establecimiento Educativo

Invoice & Delivery Addresses

Address *	<input type="text" value="Av Santa Fe"/>
ZIP *	<input type="text" value="5555"/>
City *	<input type="text" value="Buenos Aires"/>
Phone	<input type="text" value="011 444 3333"/>
Type	<input type="text" value="Invoice Address"/>

Remove address

Address *	<input type="text" value="Av. San Martín"/>
ZIP *	<input type="text" value="1111"/>
City *	<input type="text" value="Vicente López"/>
Phone	<input type="text"/>
Type	<input type="text" value="Delivery Address"/>

Remove address

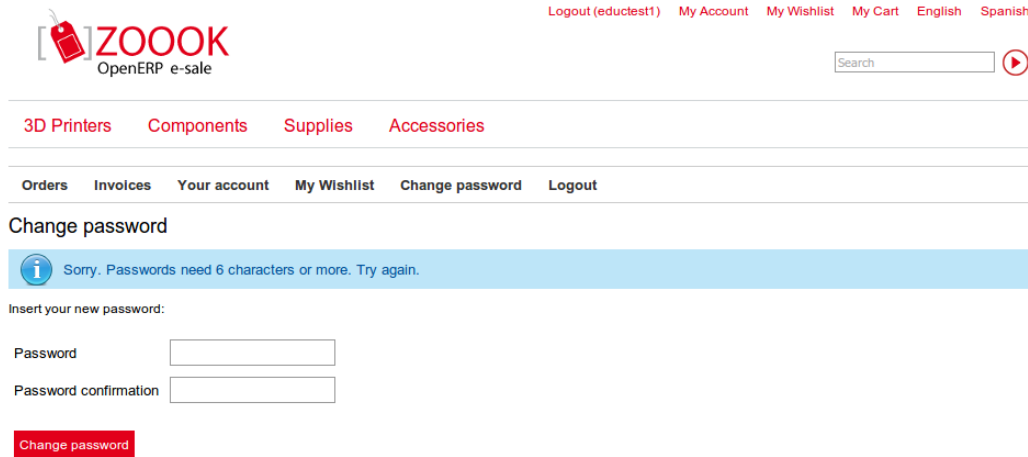
Add new address

Save

* Required Fields

Zoook Change password

Validation are applying.



The screenshot shows the Zoook e-Sale interface. At the top, there's a navigation bar with links: Logout (eductest1), My Account, My Wishlist, My Cart, English, and Spanish. Below this is a search bar. A secondary navigation bar lists categories: 3D Printers, Components, Supplies, and Accessories. A third bar lists user actions: Orders, Invoices, Your account, My Wishlist, Change password, and Logout. The main content area is titled 'Change password'. A blue error message box states: 'Sorry. Passwords need 6 characters or more. Try again.' Below this, there's a label 'Insert your new password:' followed by two input fields: 'Password' and 'Password confirmation'. At the bottom of the form is a red 'Change password' button.

About Zoook e-Sale

The Project

Zoook e-Sale was originally developed by [Zikzakmedia](#), and currently is unmaintained. This is a **fork** developed by [Enterprise Objects Consulting](#), fully functional, with active development, free and open source, and with commercial support too.

The **home page** of this project are in [Launchpad](#).

Who are using Zoook?

These are a few sites using Zoook:

- [Kikai Labs](#): Online store for sale 3D printers.
- [AvanzOsc E-Shop](#): Online shop to sale consulting services.
- [Comomio.com](#): Online community to sale pet services, this site uses Zoook as base framework.

If you are using Zoook as E-commerce or as framework tool, please send an email to the mailing list in [Launchpad](#) to be added to this list.

Versioning

This is the version 6.1.1.0 of Zoook e-Sale.

The number version is composed of two parts, separated by dots:

- The **major version**: corresponding with the OpenERP targeted. This version works with OpenERP 6.1, so, the major version of this release is 6.1.
- The **minor version**: this is the stable build release, now the 1.0.

Authors

- Mariano Ruiz <mrsarm@gmail.com> (Enterprise Objects Consulting)
- Raimon Esteve (Zikzakmedia)

Source Code

The source code of Zoook modules, Django Zoook, this documentation and some of the dependencies are in the [Launchpad Project](#).

License

This software and documentation are distributed under the [AGPL-3](#) license.