
Academica Documentation

Release v0.3

Luis Zarate, Solvo

September 09, 2015

1	Features	3
1.1	Features in stable branch	3
1.2	Features in gache branch	3
2	Installation	5
3	Run in development mode	7
4	Run in production mode	9
5	Building new modules	11
5.1	Registering in menu	11
6	Indices and tables	13

Académica is a free enrollment system made with python3 and Django, it is basic, easy to setup and easy to use.

Donation is always welcome by [Paypal](#) donate here or contac us in [info\[at\]solvosoft.com](mailto:info@solvosoft.com) o by github.

For demo go to: <https://academica.solvosoft.com/>

Features

1.1 Features in stable branch

- Social authentication for students
- Classic authentication by email with password recovery
- Course management
- Groups management
- Pre-enrollment
- Enrollment
- Student list (Pre-enrolled, enrolled)
- Paypal payment System (optional, see `matricula.contrib.bills`)
- Multilingual (spanish and english, translators are welcome)
- Page content
- Multilevel Menu and admin menu editor

1.2 Features in gache branch

Warning: They are unstable and untested.

All stable features are here too.

- Profesor management
- Classroom management (different type of classroom)
- Group, Profesor, Classroom asignation management

Installation

To install this program you need python 3 (tested with python 3.4)

```
$ git clone https://github.com/luisza/academica.git
$ cd academica/
$ pip install -r matricula/requirements/matricula.txt
```

To install billing system (Installing this is high recommended)

```
$ pip install -r matricula/requirements/bills.txt
```

To install documentation

```
$ pip install -r matricula/requirements/docs.txt
```

Configure the database

```
$ python manage.py migrate
$ python manage.py createsuperuser
```

Run in development mode

Using two shells, in the first shell run

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

In the second shell run a debug email server

```
$ python -m smtpd -n -c DebuggingServer localhost:1025
```

If you want to use billing system you will need Paypal account <https://developer.paypal.com/>

Note: For paypal payment you will need a public access point see <https://ngrok.com/> for secure tunnels

Run in production mode

This is a configuration basic modification, you probably need to adjust to your institutional security policy

Take a look django deploy documentation. <https://docs.djangoproject.com/en/1.8/howto/deployment/>

Change in academia/settings.py all variables that you want but especially this variables

Warning: SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = '*'

SECURITY WARNING: don't run with debug turned on in production!

Set debug to False

```
DEBUG = False
ALLOWED_HOSTS = ["your domain", "your subdomain"]
```

Put your email account correctly

```
EMAIL_HOST = "yourhost"
EMAIL_PORT = 25
EMAIL_USE_TLS = True
DEFAULT_FROM_EMAIL = 'youruser@yourhost'
```

see Django email documentation. <https://docs.djangoproject.com/en/1.8/topics/email/>

Change paypal account

```
PAYPAL_TEST = False
PAYPAL_RECEIVER_EMAIL = "user-buyer@example.com"
MY_PAYPAL_HOST = "https://miserver.com"
```

We recommends to deploy in https for security reason

see this howto <http://michal.karzynski.pl/blog/2013/06/09/django-nginx-gunicorn-virtualenv-supervisor/> if you never deploy django.

```
# https
SECURE_PROXY_SSL_HEADER=('HTTP_X_FORWARDED_PROTO', 'https')
SESSION_COOKIE_SECURE=True
CSRF_COOKIE_SECURE=True
```

Building new modules

New modules are django apps that can communicate with academica, for example bills, create new apps with

```
$ python manage.py startapp mymodule
```

Inside you can add all functionalities that you want.

5.1 Registering in menu

For register a menu you could make something like this in admin.py **I am used _() for multilingual support**

```
from matricula.menues import add_main_menu
add_main_menu( _("Bills"), 'bills', True, 3, True) )
```

The format can be a list of tuples or a tuple like:

(Display text, link, require authentication, order, use reverse function to extract link)

for example

```
("Bill", "/bills/", True, 3, False)
```

It is valid too

```
[ (..), ("Course", "courses", False, 0, True), (...) ]
```

Do not forget register your urls in urls.py

Indices and tables

- `genindex`
- `modindex`
- `search`