Django-Payzen Documentation

Release 0.9

Bertrand Svetchine

November 13, 2014

Contents

CHAPTER 1

django-payzen

Django app for payzen Electronic Payment Terminal.

Read extended documentation provided at Read the Docs .

Contents:

1.1 Installation

1. Install django-payzen package.

```
pip install https://github.com/bsvetchine/django-payzen/archive/master.zip
```

2. Add django_payzen in your INSTALLED_APPS

Edit your django settings.py and add django_payzen in your INSTALLED_APPS.

2. Configure your *django-payzen settings* .

In your settings.py file, add the settings specific to django-payzen.

3. Include "django_payzen.urls" in your urls.py

```
url(r'^payment/', include("django_payzen.urls")),
```

1.2 Settings

1.2.1 VADS_SITE_ID

Mandatory setting - your payzen client ID. You will find it in your payzen dashboard .

1.2.2 VADS_CERTIFICATE

Mandatory setting - the payzen certificate to use. You can generate a certificate in your payzen dashboard. There are 2 certificates ; one for testing purposes and one for production.

1.2.3 VADS_CTX_MODE

Mandatory setting - 'TEST' or 'PRODUCTION'. Remark : be careful to set the *VADS_CERTIFICATE* value accordingly to the *VADS_CTX_MODE*.

1.2.4 VADS_CURRENCY

Default currency used if the currency is not explicitely set in the payment request. vads_currency value should be the ISO 4217 code of the currency. By default vads_currency = 978, which corresponds to €. You can have look at the *django_payzen.constants.VADS_CURRENCY_CHOICES* variable to see the supported currencies.

1.2.5 VADS_ACTION_MODE

Default vads_action_mode used in the payment request. vads_action_mode can be set to :

- 'INTERACTIVE' : if the card information retrieval is done by payzen (default).
- 'SILENT' : if all card information retrieval is done by merchant site.

1.3 PaymentRequest

Have a look at the Payment Request model and at the Payzen documentation to see all the parameters that are supported by payzen.

Be careful about the amount parameter ; its value is always a positive integer repesenting the amount in cents.

```
from django_payzen import models as pz_models

payment_request = pz_models.PaymentRequest(
    amount=1000  # corresponds to 10 € if no vads_currency supplied
)
payment_request.save()

payment_request = pz_models.PaymentRequest(
    amount=1000,
    vads_url_success='http://www.loadingdata-band.com/',
    vads_url_refused='http://www.loadingdata-band.com/',
    vads_url_cancel='http://www.loadingdata-band.com/',
    order_id=1,
    order_info="Loading Data c'est d'la boulette",
    order_info2="puis Blues Pills aussi",
    vads_trans_id="000001"
)
payment_request.save()
```

1.3.1 Special PaymentRequest fields

vads_trans_id

vads_trans_id is a value composed by 6 numeric characters representing the id of the transaction. There is a unicity constraint between *vads_trans_date* and *vads_site_id*. *vads_trans_id* field is mandatory. If not set by user, django-payzen will generate randomly its value, considering the probability of having 2 generated values identical for the same day as null.

The following fields should not be set manually. Their value are computed during the save() or update() methods. Their value are computed from all other parmeters. Thus, there are some precations to take :

- if you create PaymentRequest objects using django model create function or in bulk_create, do not forget to call PaymentRequest update() method to compute the following fields.
- if you update parameters relative to the payment (could be PaymentRequest fields or adding a CustomPaymentConfig object to the PaymentRequest object, do not forget to call the update() method to update all computed fields.

signature

signature field is security parameter sent to payzen with all other payment data. The signature is computed from payment request parameters and hashed using sha1.

vads_payment_config

vads_payment_config is a string parameter allowing to set up a multiple payments. To set up multiple payments that you have 2 possibilities :

- edit a MultiPaymentConfig object. This is the simplest method hower not highly customizable.
- add one or several CustomPaymentConfig objects. This allows you to select the payment date and amount for every payment process.

vads_theme_config

vads_theme_config is a string parameter allowing to customize payzen payment pages. vads_theme_config string is computed from the ThemeConfig object linked. If no ThemeConfig is set, *vads_theme_config* is null.

1.4 *payzen_form* tag

Once you have a PaymentRequest object, you can proceed with the payment following theses steps :

1. Set up a django view that provide the PaymentRequest object to a template
2. In this template, load payzen template tags and call payzen_form tag

```
{% load payzen_extras %}

{% payzen_form object %}
```

payzen_form tag renders a hidden form including all PaymentRequest data. The form is directly submitted to payzen url and the user is redirected to payzen card registration process.

By default *payzen_form* will not autosubmit. You can do that calling :

```
{% payzen_form object auto_submit=True %}
```

1.5 Signals

You may need to automate actions once a payment is received or rejected or if django_payzen cannot read/analyze payzen responses. The following signals allow you to do that.

1.5.1 payment_success

signal sent once a payment sucessful response is received and saved. The PaymentResponse object is sent in argument.

1.5.2 payment_failure

signal sent once a payment unsuccessful response is received and saved. You can analyze data in PaymentResponse to check what wend wrong. The PaymentResponse object is sent in argument.

1.5.3 payment_error

signal sent if django_payzen cannot read or analyze the request sent by payzen.

1.6 Testing

1. Install the requirements.

```
pip install -r test-requirements.txt
```

2. Find a public interface.

If you want to test django-payzen locally and you don't have a public IP you need to expose your localhost over internet to be able to received the payzen payment response request.

You can use a tool like ngrok . Ngrok will give you a public url from wich you can access to your localhost server.

Then you need either to edit your serveur notifications urls or to specify explicitely the notification url in your payment request (recommended).

3. Launch the testing serveur

You need to specify the liverserveur url to 0.0.0.0 to make the testing serveur visible from the network and specify the port accordingly to your ngrok setup.

```
python manage.py test django_payzen --liveserver 0.0.0.0:8000
```