

---

# Django WalletOne Documentation

*Release 1.0.0*

**Stanislav Otovchits**

April 28, 2016



<b>1</b>	<b>Install</b>	<b>3</b>
<b>2</b>	<b>Using</b>	<b>5</b>
2.1	Payment form . . . . .	5
2.2	Getting paid results . . . . .	6
2.3	Signals . . . . .	6
<b>3</b>	<b>Tests</b>	<b>9</b>
<b>4</b>	<b>Update the database</b>	<b>11</b>
<b>5</b>	<b>Release notes</b>	<b>13</b>
5.1	Version 1.0.0a5 . . . . .	13
5.2	Version 1.0.0a1 . . . . .	13
<b>6</b>	<b>Indices and tables</b>	<b>15</b>



Django WalletOne is a pluggable application that integrates with WalletOne online payments provider.

Before any using this application it is strongly recommended to go through the [official Wallet One documentation](#).

Contents:



---

## Install

---

Install using pip:

```
pip install django-walletone
```

Add application to INSTALLED\_APPS:

```
INSTALLED_APPS = [  
    # ...  
    'walletone.apps.DjangoWalletoneConfig',  
    # ...  
]
```

Settings:

```
# Online store ID (account number)  
# received at registration  
DJANGO_W1_MERCHANT_ID = '123456789012'  
  
# EDS creation method  
DJANGO_W1_SIGN_METHOD = 'md5'  
  
# Secret key from your W1 account  
DJANGO_W1_SECRET_KEY = 'sekret key'  
  
# Online store web-pages addresses (URL),  
# where buyer will be directed after successful  
# or unsuccessful payment.  
DJANGO_W1_SUCCESS_URL = 'https://your.domain/payment/success/'  
DJANGO_W1_FAIL_URL = 'https://your.domain/payment/fail/'  
  
# Currency ID (ISO 4217)  
# 643 -- Russian Rubles  
# 840 -- US Dollar  
# 978 -- Euro  
# ...  
DJANGO_W1_CURRENCY_DEFAULT = '643'
```

Settings your root urls:

```
urlpatterns = [  
    # ...  
    url(r'^w1/', include('walletone.urls')),  
    # ...  
]
```

Update your database:

```
./manage.py migrate
```



## 2.1 Payment form

In order to simplify building payment form django-walletone has the `WalletOnePaymentForm`. It is necessary to simplify rendering information in the templates and signature field calculation.

For example:

```
from walletone.models import WalletOneSuccessPayment

def payment_form_example(request):
    form = WalletOnePaymentForm(initial={
        'WMI_PAYMENT_AMOUNT': '99.00',
        'WMI_DESCRIPTION': 'Order fro what?',
        'WMI_PAYMENT_NO': '1',
        'EXTRA_FIELD': 'value',
    })

    return render(
        request,
        'payment_form.html',
        {'form': form}
    )
```

**Note:** You can use here any form field from [W1 docs](#) and any extra field just typing it in form's initial.

**Warning:**

Notice that for now you can't use multiple values for form field like `WMI_PTENABLED` and `WMI_PTDISABLED`.

You can set only one value for `WMI_PTENABLED` and only one value for `WMI_PTDISABLED`.

It's alpha version. Sorry :)

The corresponding template `payment_form.html`:

```
<form action="{{ form.action_url }}" method="POST" accept-charset="UTF-8">
  <p>{{ form.as_p }}</p>
  <p><input type="submit" value="Buy"></p>
</form>
```

### Note:

The form is rendered as a set of input hidden tags.

The form has a `action_url` attribute that contains the correct W1 processing URL.

Notice `accept-charset="UTF-8"`. It is better to **always** set.

Note that `{% csrf_token %}` no needed here.

---

## 2.2 Getting paid results

From official docs:

*Once the buyer completes the payment order, Wallet One Checkout performs POST-request to the “Data to send the results of transaction”, indicated the in the online store settings. This request contains parameters of the payment form, information about the result of payment and some additional parameters.*

So if you set your urls like:

```
urlpatterns = [
    url(r'^w1/', include('walletone.urls')),
]
```

you need to set “Data to send the results of transaction”:

```
https://your.domain/w1/confirm/
```

After POST request from W1 app saves information about success payment into the database (see model `WalletOneSuccessPayment`) and generates signal (see below *Signals*).

What if I send to W1 some extra fields? For example “EXTRA\_FIELD1” and “EXTRA\_FIELD2”. How can I retrieve it from my database? Very simple. Each `WalletOneSuccessPayment` instance has the `extra_attrs` attribute. For instance to get above attr just access it like any other python attribute:

```
payment.extra_attrs.EXTRA_FIELD1
payment.extra_attrs.EXTRA_FIELD2
```

## 2.3 Signals

`django-walletone` sends `payment_received` signal after success confirmation from W1 and after saving information to a database. `payment_received` signal provides `payment` arg contains all information about payment.

Example:

```
from walletone.signals import payment_received
from walletone.models import WalletOneSuccessPayment

def receiver(**kwargs):
    payment = kwargs['payment']
    assert payment ==
        WalletOneSuccessPayment.objects \
            .get(WMI_ORDER_ID=payment.WMI_ORDER_ID)
```

```
payment_received.connect(receiver)
```



---

## Tests

---

To run the django-walletone tests:

- Download the [source from GitHub](#) or your fork.
- Install tox:

```
pip install tox
```

- Run tox:

```
tox
```

This will run all the tests on all supported combinations of Django/Python.



---

## Update the database

---

django-walletone uses the built in Django migrations framework.

To update your database run:

```
./manage.py migrate
```





---

## Release notes

---

### 5.1 Version 1.0.0a5

- Fix home page url in setup.py
- README enhancements

### 5.2 Version 1.0.0a1

- Alpha version
- No support multiple values for WMI\_PTENABLED and WMI\_PTDISABLED fields



---

## Indices and tables

---

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