

**FOR PYTHON
WEB APPLICATIONS**

Raphael Michel / @_rami_



*WordPress is used by 59.6% of all the websites whose content management system we know. This is **28.7% of all websites.***

Source: w3techs.com

Plugins

Extend your WordPress experience with 52,401
plugins.





- Open Source event ticket shop
- Python/Django stack
- Key design goal: Be extensible. Don't make people patch or fork it.

IDEAS FOR PLUGINS

- Payment methods
- Export formats, ticket layouts
- Additional features

OUR PLAN

1. Establish a way that plugins can hook into your application
2. Provide many of these hooks
3. Document them well
4. There is no step four!

LET'S WRITE SOME PYTHON!

SIMPLE SIGNAL SYSTEM

```
class Signal:
    def __init__(self):
        self.receivers = []

    def register(self, func):
        self.receivers.append(func)
        return func

    def send(self, *args, **kwargs)
        return [
            func(*args, **kwargs)
            for func in self.receivers
        ]
```

```
user_created = Signal()
```

```
@user_created.register
def plugin_listener():
    send_mail()
```

DJANGO.DISPATCH.SIGNAL

- Similar API
- Handles thread-safety for you
- Caching, weak references, ...

→ If you're on Django, use this one.

1 PLUGIN = 1 DJANGO APP

- Plugins can have their own models
- Plugins can have their own templates
- Plugins can have their own static files
- ...

HOW TO INSTALL A PLUGIN

- Like a django app!
- Install package with source code
- Add to `INSTALLED_APPS` in `settings.py`
- Add url include to `urls.py`

Too many steps!

And we don't want to touch code...

ALL WE WANT IST

```
$ pip install pretix-xyz
```

(+ migrations, maybe)

PLUGIN: __INIT__.PY

```
from django.apps import AppConfig

class PluginApp(AppConfig):
    name = 'xyz'
    verbose_name = 'XYZ plugin'

    class PretixPluginMeta:
        name = 'XYZ plugin'

    def ready(self):
        from . import signals

default_app_config = 'pretix_xyz.PluginApp'
```

PLUGIN: SETUP.PY

```
setup(  
    name='pretix-xyz',  
    install_requires=[],  
    packages=find_packages(exclude=['tests', 'tests.*']),  
    include_package_data=True,  
  
    entry_points={  
        'pretix.plugin': [  
            'pretix_xyz=pretix_xyz:PretixPluginMeta'  
        ]  
    }  
  
)
```

APP: SETTINGS.PY

```
from pkg_resources import iter_entry_points

for entry_point in iter_entry_points(
    group='pretix.plugin', name=None):

    INSTALLED_APPS.append(entry_point.module_name)
```


PLUGIN: URLS.PY

```
urlpatterns = [  
    url(...),  
    ...  
]
```

APP: URLS.PY

```
plugin_patterns = []
for app in apps.get_app_configs():
    if hasattr(app, 'PretixPluginMeta'):
        if importlib.util.find_spec(app.name + '.urls'):
            urlmod = importlib.import_module(
                app.name + '.urls')
            plugin_patterns.append(
                url('', include((singlurlmod.urlpatterns,
                                app.label))))
        )

urlpatterns = [ ...,
    url('', include((plugin_patterns, 'plugins'))),
]
```

```
{% url "plugins:pretix_xyz:my.url.name" %}
```

THAT'S OUR PLUGIN SYSTEM!

BONUS: PLUGINS PER CLIENT/USER

- Custom `Signal()` subclass
- Store list of enabled plugins per client
- More URL magic (if wanted)

Installed plugins

Bank transfer

Disable

Version 1.8.0.dev0 by *the pretix team*

This plugin allows you to receive payments via bank transfer

Stripe

Enable

Version 1.8.0.dev0 by *the pretix team*

This plugin allows you to receive credit card payments via Stripe

PayPal

Enable

Version 1.8.0.dev0 by *the pretix team*

This plugin allows you to receive payments via PayPal

PDF ticket output

Disable

Version 1.8.0.dev0 by *the pretix team*

This plugin allows you to print out tickets as PDF files

MAKE IT EASY

```
{% load signal %}  
...  
{% signal "pretix.control.signals.order_info" order=order %}
```

```
@register.simple_tag  
def signal(signame: str, **kwargs):  
    sigmodule, signame = signame.rsplit('.', 1)  
    sigmodule = importlib.import_module(sigmodule)  
    signal = getattr(sigmodule, signame)  
    html_result = []  
    for receiver, response in signal.send(event, **kwargs):  
        if response:  
            html_result.append(response)  
    return mark_safe("".join(html_result))
```

WRITE DOCUMENTATION

No, seriously.

PROVIDE A COOKIECUTTER TEMPLATE

```
$ cookiecutter \  
https://github.com/pretix/pretix-  
plugin-cookiecutter
```

AUTO-INSTALL

Naah, better don't.

THANK YOU! ANY QUESTIONS?



Raphael Michel

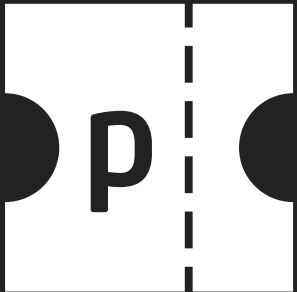
mail@raphaelmichel.de

 @_rami_  raphaelm

pretix.eu

support@pretix.eu

 @pretixeu  pretix



django **con**

EUROPE 2018

MAY 23-27TH, 2018

HEIDELBERG, GERMANY

2018.DJANGOCON.EU