
ideos Documentation

Release 0.1.dev6

Eldarion

Sep 27, 2017

Contents

1	Getting started	3
1.1	Prerequisites	3
1.2	Installation	3
1.3	Usage	4
2	Reference	5
2.1	Settings	5
2.1.1	<code>AUTH_PROFILE_MODULE</code>	5
2.2	Modules	5
2.2.1	<code>idios.urls</code>	5
2.2.2	<code>idios.utils</code>	5
2.2.3	<code>idios.views</code>	5
	Python Module Index	7

idios is an extensible user profile app built by Eldarion. It was originally designed to improve upon Pinax profile features and has now been open-sourced.

CHAPTER 1

Getting started

This document is designed to get you up and running with idios. This app is designed for use with [Django](#), however, can be used with [Pinax](#). Pinax can provide some extra components which will enhance the functionality of idios. Those extra components are designed for Django too and Pinax simply brings them together.

Prerequisites

These are the requirements to run idios:

- Python **2.4+** (Python 3.x is **not** supported yet)
- Django **1.2+**

Installation

To install idios use [pip](#):

```
pip install idios
```

Add idios to your `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    # ...  
    "idios",  
]
```

Hook up idios to your `URLconf`:

```
urlpatterns = patterns("",  
    # ...  
    url(r"^profiles/", include("idios.urls"))  
)
```

This is largely all you need to do to get idios setup in your project. Continue reading on to learn more about how to use idios.

Usage

To get Django to link `User` instances to their profile pages you can add this bit to `settings.py`:

```
ABSOLUTE_URL_OVERRIDES = {
    "auth.user": lambda o: "/profiles/profile/%s/" % o.username,
}
```

Now when you call `get_absolute_url` on any `User` instance it will return the URL to that user's profile as configured above.

idios is designed to give you full control of the profile model. idios provides a `ProfileBase` for your profile model to extend. To hook in your profile model add this to your `settings.py`:

```
AUTH_PROFILE_MODULE = "myapp.Profile"
```

This will additionally setup your profile model as the one Django's auth app will use.

Where you put your profile model is entirely up to you. A common practice we've used is either having a project specific app named `profiles` or named after the project (we name all projects with `_project` suffix leaving the first bit open for an app name).

Here is what how your profile model might look like:

```
from django.db import models

from idios.models import ProfileBase


class Profile(ProfileBase):

    name = models.CharField(max_length=100)
    phone_number = models.CharField(max_length=20)
```


This document covers various components of idios.

Settings

`AUTH_PROFILE_MODULE`

The site-specific user profile model used by this site. This setting is used by both Django and idios for similar purposes. Example:

```
AUTH_PROFILE_MODULE = "myapp.Profile"
```

The first part, `myapp`, is an app on your `sys.path`. The second part, `Profile`, is a model defined in `models.py` of `myapp`.

Modules

`idios.urls`

`idios.utils`

`idios.views`

i

- `idios.utils`, 5
- `idios.views`, 5

I

`idios.utils` (module), 5
`idios.views` (module), 5