
pypusher Documentation

Release 0.1.0

Igor 'idle sign' Starikov

April 26, 2013

CONTENTS

1	Requirements	3
2	Contents	5
2.1	Getting started with pypusher	5
2.2	Apple Push Notifications	6
2.3	Exceptions	6
3	Get involved into pypusher	7
4	The tip	9

pypusher pushes stuff from Python

This module allows you to send Apple Push Notifications to various Apple devices from your Python code.

It relies upon *libcapn* to feature *Apple Push Notification Service* support.

REQUIREMENTS

1. Python 3
2. capn library - <http://libcapn.org/>

CONTENTS

2.1 Getting started with pypusher

Note: Please install *libcapn* <http://libcapn.org/> to use pypusher.

```
from pypusher.pypusher import *

# Check whether capn library version is supported by pypusher.
if lib_capn_version_satisfied():

    # Create a notification for device 'DEVICE_01_TOKEN_HEXSTRING'.
    notification = AppleNotification('This notification is brought to you by pypusher.', ['DEVICE_01_TOKEN_HEXSTRING'])

    # Add another recipient.
    notification.add_recipient('DEVICE_02_TOKEN_HEXSTRING')
    # Set badge number to show 10.
    notification.set_badge(10)
    # Set expiration timestamp to 2014-10-10 10:10:10
    notification.set_expires_at(1412921410)

    # Send the notification using the connection data. And handle possible exceptions.
    try:
        # Set connection parameters: connection certificate and private key.
        with ApplePushConnection('/home/idle/my_cert.pem', '/home/idle/my_key.pem') as connection:
            # Let's use Apple sandbox servers for tests.
            connection.use_sandbox()
            connection.send_notification(notification)
    except AppleException as e:
        print('Pusher failed his mission: %s' % e)

    # Let's hunt stale device tokens if any.
    try:
        with AppleFeedbackConnection('/home/idle/my_cert.pem', '/home/idle/my_key.pem') as connection:
            stale_tokens = connection.get_stale_recipients()
            print(stale_tokens)
    except AppleException as e:
        print('Pusher failed on feedback: %s' % e)
```

2.2 Apple Push Notifications

pypusher relies upon two classes to send Apple push notifications: `ApplePushConnection` and `AppleNotification`.

`AppleFeedbackConnection` class allows to get feedback information on stale device tokens from Apple.

2.2.1 Utilities

Warning: When troubleshooting please use `lib_capn_version_satisfied()` to check whether C library is found and has an appropriate version.

2.2.2 ApplePushConnection

2.2.3 AppleFeedbackConnection

2.2.4 AppleNotification

2.3 Exceptions

pypusher features the following exceptions to tackle with errors related to Apple Pushes.

2.3.1 AppleException

2.3.2 CapnLibraryCallException

GET INVOLVED INTO PYPUSHER

Submit issues. If you spotted something weird in library behavior or want to propose a feature you can do that at <https://github.com/idlesign/pypusher/issues>

Write code. If you are eager to participate in library development, fork it at <https://github.com/idlesign/pusher>, write some code, whether it should be a bugfix or a feature implementation, and make a pull request right from the forked project page.

Spread the word. If you have some tips and tricks or any other words in mind that you think might be of interest for the others — publish it.

THE TIP

Sick and tired of apples? Have some bananas.