

ARTIGO: 271

Push Server - Enviando notificações

Intrudução

O Mobile Mind Push Server oferece um serviço para envio de notificações push para as seguintes plataformas:

- Android FCM
- IOS

O sistema encapsula toda a complexidade que envolve envio de mensagens para cada uma dessas plataformas através de uma unica API.

Nesse artigo veremos como:

- Recuperar todos os usuários
- Enviar notificações

**** Todas as chamadas requerem autenticação basic

Recupernado os usuários da minha aplicação

Ao acessar um aplicativo devidamente configurado, um token é gerado e enviado para o servidor Push Server juntamente com informações sobre o usuário. Para recuperar esses usuários usamos a API da seguinte forma:

```
GET http://push.mobilemind.com.br/apps/myapp-channel
```

exemplo de retorno

```
{
  "fulano@email.com": [
    {
      "subscriber_id": "ERyVTeIuyP",
      "name": "Fulano de tal",
      "email": "fulano@email.com",
      "production": true,
      "ios": false,
      "android": true
    },
    {
      "subscriber_id": "PoiHgFFt",
      "name": "Fulano de tal",
      "email": "fulano@email.com",
      "production": true,
      "ios": true,
      "android": false
    }
  ],
  "sicrano@email.com": [
    {
      "subscriber_id": "UHdsuUYuY",
      "name": "Sicrano e tal",
      "email": "sicrano@email.com",
      "production": true,
      "ios": false,
      "android": true
    }
  ]
}
```

Para cada email retornará uma lista de dispositivos registrador, cada qual com seu identificador (subscribe_id)

Enviando notificações:

```
POST http://push.mobilemind.com.br/event/myapp
```

conteúdo da mensagem

```
data = {  
  'msg': "Mensagem da notificação",  
  'title': 'Título da notificação',  
  'data.user_id': subscriber_id  
}
```

Código exemplo

python

```
# -*- coding: utf-8 -*-  
  
import requests  
import json  
from requests.auth import HTTPBasicAuth  
  
class NotificationService(object):  
  
    def __init__(self):  
        self.push_server = 'http://push.mobilemind.com.br'  
        self.basic_auth = HTTPBasicAuth('user-name', 'user-password')  
        self.channel = "app-channer"  
  
    def list_users(self):  
        action = 'apps/%s' % self.channel  
        r = requests.get("%s/%s" % (self.push_server, action), auth=self.basic_auth)  
        return r.json()  
  
    def do_notify_user(self, notification, username):  
  
        notification_users = self.list_users()  
        user_push = None  
  
        for username in notification_users:  
  
            if username != username:  
                continue  
  
            user_push = notification_users[username]  
            break  
  
        if not user_push:  
            raise Exception("user %s not found" % username)  
  
        for it in user_push:  
            self.do_notify(notification, it['subscribe_id'])  
  
    def do_notify(self, note, subscribe_id):  
        action = 'event/%s' % self.channel  
  
        nt = {  
            'msg': note.message,  
            'title': note.title,  
            'data.user_id': subscribe_id  
        }  
        requests.post("%s/%s" % (self.push_server, action), data=nt, auth=self.basic_auth)
```