
CakePHP-Push Documentation

Release 0.0.1

Romain Monteil

September 25, 2016

1	Introduction	1
1.1	CakePHP Push	1
1.2	Requirements	1
2	Installation	3
2.1	Using Composer	3
2.2	Enable plugin	3
3	Configuration	5
4	Firestore Cloud Messaging Adapter	7
4.1	Introduction	7
4.2	Configuration	7
4.3	Usage	7
4.4	Example	8
4.5	Appendice	8
5	Indices and tables	9

Introduction

1.1 CakePHP Push

Push is a plugin for CakePHP to send downstream message through services like Firebase Cloud Messaging.

1.2 Requirements

- PHP $\geq$ 5.6
- CakePHP $\geq$ 3.3

Installation

2.1 Using Composer

To install this plugin, run `composer require ker0x/cakephp-push` or add this snippet in your project's `composer.json`.

```
{
    "require": {
        "ker0x/cakephp-push": "dev-master"
    }
}
```

2.2 Enable plugin

You need to enable the plugin in your `config/bootstrap.php` file:

```
Plugin::load('ker0x/Push', ['bootstrap' => true]);
```

Configuration

Make a config file `config/push.php`:

```
return [
    'Push' => [
        'adapters' => [
            'Fcm' => [
                'api' => [
                    'key' => '<api-key>',
                    'url' => 'https://fcm.googleapis.com/fcm/send',
                ],
            ],
        ],
    ],
];
```

Firebase Cloud Messaging Adapter

4.1 Introduction

First, you have to get an API key. Go to <https://console.firebase.google.com/>, create a project then in your project's settings you will see your Web API Key.

4.2 Configuration

Update your `push.php` config file to set Fcm to the adapters's array:

```
return [
    'Push' => [
        'adapters' => [
            'Fcm' => [
                'api' => [
                    'key' => 'your_web_API_key',
                    'url' => 'https://fcm.googleapis.com/fcm/send',
                ],
            ],
        ],
    ],
];
```

4.3 Usage

```
use ker0x\Push\Adapter\FcmAdapter;

$adapter = new FcmAdapter();
$adapter
    ->setTokens($tokens)
    ->setNotification($notification)
    ->setDatas($datas)
    ->setParameters($parameters);
```

where:

- `$tokens` is an array of device's token. (required)
- `$notification` is an array containing the notification. (optional)

- `datas` is an array with some datas that will be passed. (optional)
- `$parameters` is an array of parameters for the notification. (optional)

4.4 Example

```
use ker0x\Push\Adapter\FcmAdapter;
use ker0x\Push\Push;

$adapter = new FcmAdapter();
$adapter
    ->setTokens(['1', '2', '3', '4'])
    ->setNotification([
        'title' => 'Hello World',
        'body' => 'My awesome Hello World!'
    ])
    ->setDatas([
        'data-1' => 'Lorem ipsum',
        'data-2' => 1234,
        'data-3' => true
    ])
    ->setParameters([
        'dry_run' => true
    ]);

$push = new Push($adapter);

// Make the push
$result = $push->send();

// Get the response
$response = $push->response();
```

4.5 Appendice

[FCM HTTP Protocol reference](#)

Indices and tables

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