

---

# **CakePHP-Push Documentation**

***Release 0.0.1***

**Romain Monteil**

**Apr 23, 2017**



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



### CakePHP Push

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

### Requirements

- PHP  $\geq$  5.6
- CakePHP  $\geq$  3.4.\*



## 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": "~2.0"
    }
}
```

## Enable plugin

You can load the plugin by typing the following command in a terminal:

```
bin/cake plugin load ker0x/Push -b
```

or by adding the following line inside your `config/bootstrap.php` file:

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





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

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



---

## Firestore Cloud Messaging Adapter

---

### Introduction

It currently only supports HTTP protocol for :

- sending a downstream message to one or multiple devices

### Configuration

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.

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',
                ],
            ],
        ],
    ],
];
```

## Usage

```
use Kerox\Push\Adapter\Fcm;

$adapter = new Fcm();
$adapter
    ->setTokens($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)
- `$paramaters` is an array of parameters for the notification. (optional)

## Example

```
use Kerox\Push\Adapter\Fcm;
use Kerox\Push\Push;

$adapter = new Fcm();
$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();
```

## Appendice

[FCM HTTP Protocol reference](#)

## CHAPTER 5

---

### Indices and tables

---

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