

Laravel and Websocket with ReactPHP Example Chat App

Brain Socket 을 사용하면 <https://github.com/BrainBoxLabs/brain-socket>
Laravel 의 artisan command 를 사용하여 ReactPHP 기반의
Websocket 서버를 구동할 수 있습니다.

그리고 <https://github.com/BrainBoxLabs/brain-socket-js> 의 샘플 소스를 통해서
구동된 websocket 서버를 사용한 realtime chat app을 확인할 수 있습니다.

1. composer require add
composer.json require 섹션에 다음을 추가합니다.

"brainboxlabs/brain-socket": "v1.0.0"

2. app/config/app.php 의 providers 섹션에
'BrainSocket\BrainSocketServiceProvider', 을 추가해줍니다.

3. app/config/app.php 의 alias 섹션에
'BrainSocket' => 'BrainSocket\BrainSocketFacade', 을 추가해줍니다.

4. 이제 커맨드 창에서 php artisan 을 실행하면 brainsocket 관련 명령어가 추가된것을 확인할 수 있습니다.

5. php artisan brainsocket:start --port=8081 을 입력하면 8081번 포트로 reactphp
기반의 웹소켓 서버를 구동할 수 있습니다 .

6. 웹 소켓 서버만 구동해서는 아무런 대응을 할 수 없기 때문에
클라이언트에 대응하는 event 를 추가해주어야 합니다.

app/events.php 파일을 새롭게 생성하고 다음을 추가해줍니다.

```
<?php
Event::listen('generic.event',function($client_data){
    return BrainSocket::message('generic.event',array('message'=>'A message from a
generic event fired in Laravel!'));
});
```

```
Event::listen('app.success',function($client_data){
    return BrainSocket::success(array('There was a Laravel App Success Event!'));
});
```

```
Event::listen('app.error',function($client_data){
    return BrainSocket::error(array('There was a Laravel App Error!'));
});
```

7. 생성한 events.php 파일을 app/start/global.php 파일에 추가해줍니다.
filters.php 파일뒤에 추가하면 아래와 같아집니다.

```
require app_path().'/filters.php';
require app_path().'/events.php';
```

이제 클라이언트를 확인합니다.

8. <https://github.com/BrainBoxLabs/brain-socket-js> 를 클론하여
샘플 소스에서 웹소켓이 연결되는 localhost 를 설정한 서버로 잘설정되었는지
포트가 올바르게 설정되었는지 확인하고 창을 여러개 띄워서 채팅이 잘 이루어 지는지 확인합니다.

9. brainsocket 은 reactphp 기반이므로 일부 php extension 이 지원하지 않는 경우 구동되지 않을
수도 있습니다 .