

Web

타입스크립트로 슬랙 앱(봇) 만들기 - 기초편

슬랙 앱을 통해 간단한 메시지 전송 기능을 만들어보자.

#슬랙

#Slack

#Typescript

#슬랙봇

#타입스크립트

#슬랙앱

2020.05.27

목차

1. 슬랙 앱(봇) 이란?
2. 슬랙 앱 활용 사례
3. 슬랙 앱 생성하기
4. 타입스크립트로 슬랙 봇 만들기
5. 사용할 스코프 추가하기
6. 슬랙 앱 설치하기
7. 슬랙 앱으로 메시지 받아보기
8. 슬랙 앱으로 메시지 보내기
9. 아쉬운 점
10. 다음편

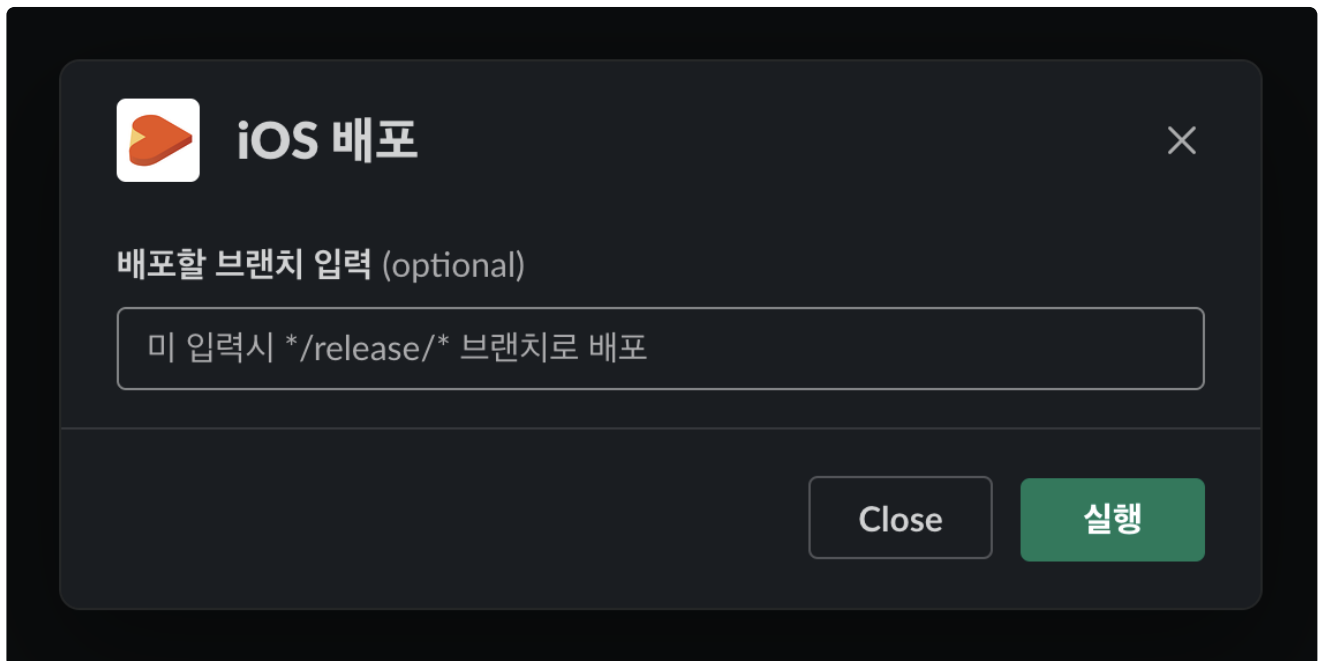
1. 슬랙 앱(봇) 이란?

국내에서도 인지도가 있는 업무용 메신저인 슬랙을 아시나요? 강남언니에서도 사내 메신저로 슬랙을 쓰고 있습니다. 슬랙의 큰 장점 중 하나인 서드파티 앱(이하 슬랙 앱)을 만들어 보려고 합니다. 슬랙 앱은 외부에서 발생한 액션을 슬랙으로 전달하거나, 슬랙 안에서의 사용자의 액션을 기반으로 외

부에 액션을 전달하는 서비스입니다. 정리 하자면, 여러 곳에서 발생한 푸시를 슬랙으로 모아 한곳에서 볼 수 있고, 슬랙의 명령어와 모달, 입력폼을 활용하여 별도의 웹 페이지 없이 액션을 할 수 있습니다.

2. 슬랙 앱 활용 사례

1. Trello [\[링크\]](#) : 사용 중인 트렐로 계정과 연결하여 보드의 상태 변경을 알려주고, 슬랙에서 카드를 등록할 수 있는 기능을 지원해줍니다.
2. Polly [\[링크\]](#) : 슬랙 안에서 투표 기능을 지원해줍니다.
3. 힐봇 : 강남언니에서 사용하고 있는 슬랙 봇입니다.



- 명령어를 통해 지라에 버그 관련 티켓 생성을 해줍니다.
- 앞으로 IoT 를 활용한 사무실 전원 관리를 목표로 업데이트 해 나가고 있습니다.
- 매일 통계자료를 알려줍니다.

- 점심 시간 1시간 여 전 같이 식사할 사람들을 매칭해줍니다.



- 메신저 안에서 특정 브랜치명을 입력하면 빌드 및 배포를 해줍니다.

3. 슬랙 앱 생성하기

슬랙 앱을 만드려면 슬랙 API 사이트에서 앱을 생성해야 합니다.

먼저 <https://api.slack.com/apps> 로 접속합니다. 접속 후 'Create New App' 을 누른 뒤 App Name 과 Development Slack Workspace 를 입력한 뒤 앱을 생성합니다

Create a Slack App

App Name

e.g. Super Service

Don't worry; you'll be able to change this later.

Development Slack Workspace

Development Slack Workspace

Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel

Create App

슬랙 앱을 생성하고 나면 사이트 좌측 메뉴(이하 전역 메뉴)에서 Settings > Basic Information 으로 접속됩니다. 해당 페이지 맨 밑에 Display Information 영역에 슬랙에서 보여질 앱의 이름과 **아** **이** **콘** **이** **미** **지**(**사이즈 w : 2000px, h : 2000px 필수**), 앱의 배경 색을 입력 합니다.

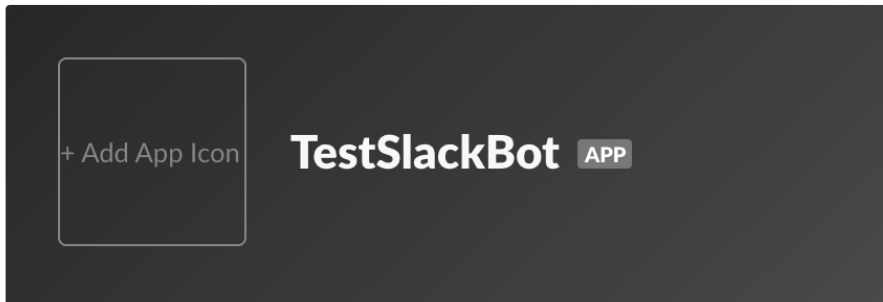
Display Information

This information will be shown in the Slack App Directory and in the Slack App
For more information, view our [App Detail Guidelines](#).

App name

Short description

App icon & Preview



Background color



4. 타입스크립트로 슬랙 봇 만들기

※ 타입스크립트로 개발하기 때문에 node.js 설치가 필수입니다.

먼저 디렉토리를 생성하고 기본적인 프로젝트 구성 및 모듈을 설치합니다. 그리고 필요한 파일들(ts config.json, src/index.ts, config/bot.json)을 생성합니다.

```
# 슬랙봇을 개발할 디렉토리 생성 후 해당 디렉토리로 이동
$ mkdir slackbot && cd slackbot

$ npm init
# package.json 생성 후 필요한 모듈 설치
# EventEmitter 관련 이슈로 인해 12 버전의 타입을 설치
$ npm install --save-dev @types/node@12.7.0
$ npm install --save-dev typescript
$ npm install --save-dev ts-node
$ npm install --save-dev express
$ npm install --save-dev @types/express
$ npm install --save @slack/events-api

# 타입스크립트 config 파일 생성
$ touch tsconfig.json
```

```
# src 디렉토리 생성 후 index.ts 파일 생성하여 슬랙봇을 구현
$ mkdir src && touch src/index.ts
#
$ mkdir config && touch config/bot.json
```

슬랙 API 사이트에서 슬랙 앱을 생성했다면, 전역 메뉴에서 Settings > Basic Information 페이지 중간에 App Credentials 영역에 있는 Signing Secret 값을 'Show' 후 복사합니다.

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID

A0

Date of App Creation

May 10, 2020

Client ID

45

Client Secret

.....

Show

Regenerate

You'll need to send this secret along with your client ID when making your [oauth.v2.access](#) request.

Signing Secret

.....

Show

Regenerate

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

tvyn

Regenerate

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.

복사한 Signing Secret 값을 config/bot.json 에 저장합니다.

```
// config/bot.json
// XXXX 에 복사한 Signing Secret 값으로 덮어쓰기 합니다.

{
  "SIGNING_SECRET": "XXXX"
}
```

tsconfig.json 파일에 아래의 코드를 복사하여 붙여넣기 합니다. (최상단 파일명의 주석은 제외)

```
// tsconfig.json

{
  "compilerOptions": {
    "resolveJsonModule": true,
    "esModuleInterop": true
  },
  "include": ["src/*"]
}

/*
    resolveJsonModule : json 파일을 import 로 받아오는 걸 허용하는 옵션
    esModuleInterop : commonJS 형태의 모듈도 import 로 받아오는 걸 허용해주는 옵션
*/
```

src/index.ts 파일에 슬랙 앱을 구현하기 위해 아래의 코드를 복사하여 붙여넣기 합니다.

```
// src/index.ts

import express from 'express';
import { createEventAdapter } from '@slack/events-api';
import { createServer } from 'http';

// 생성한 슬랙봇에 대한 키값들
import CONFIG from '../config/bot.json';
/*
{
  "SIGNING_SECRET": "XXXX"
}
*/

// 슬랙에서 슬랙봇에게 접근가능한 엔드포인트를 만들기 위해 웹서버(express)를 사용
const app = express();

const slackEvents = createEventAdapter(CONFIG.SIGNING_SECRET);
```

```
// 메시지 이벤트 구독하기
slackEvents.on('message', async (event) => {
  console.log(event);
});

// 메시지 이벤트 엔드포인트를 express 에 등록하기
app.use('/slack/events', slackEvents.requestListener());

// express 웹 서버 실행
createServer(app).listen(3000, () => {
  console.log('run slack bot');
});
```

터미널에서 슬랙 앱을 실행합니다.

```
# localhost:3000 으로 웹 서버가 실행됨.
$ npx ts-node ./src/index.ts
```

외부에서 localhost:3000 을 접속시키기 위한 여러가지 방법 중 ngrok 이라는 프로그램을 사용해 봅니다.

<https://ngrok.com/download> 에서 다운받아서 설치한 뒤 터미널에서 커맨드로 사용하면 됩니다.

참고자료 : [\[express에서 ngrok으로 외부에 서버 공개하기\]](#)

```
# 외부에서 로컬에서 실행한 웹서버를 접근하기 위해 ngrok 사용
# http 로 3000 포트를 외부로 연결시킨다는 옵션
$ ngrok http 3000

# 외부에서 5c943fb2.ngrok.io 로 접속 시 localhost:3000 으로 연결시켜줌
Forwarding      http://5c943fb2.ngrok.io -> http://localhost:3000
```

5. 사용할 스코프 추가하기

슬랙 API 사이트에서 전역 메뉴의 Features > Event Subscriptions 페이지로 접속합니다. 접속 후 Enable Events 를 Off 에서 On 으로 변경합니다.

Enable Events



Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)

'On' 으로 바뀌면 아래 처럼 URL 입력폼과 구독할 이벤트 선택 영역이 나옵니다. Request URL 에 index.ts 에서 생성한 웹 서버의 이벤트용 엔드포인트를 입력합니다.

<http://5c943fb2.ngrok.io/slack/events> (http://localhost:3000/slack/events 와 동일)

직접 만든 슬랙 앱의 엔드포인트와 잘 연결되면 Request URL 라벨 옆에 Verified 라는 문구가 표시 됩니다.

Enable Events

On

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)



Request URL **Verified** ✓

[Change](#)

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a **challenge** parameter, and your endpoint must respond with the challenge value. [Learn more.](#)

Subscribe to bot events

Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary [OAuth scope](#) for you.

Event Name	Description	Required Scope	
message.im	A message was posted in a direct message channel	im:history	
message.channels	A message was posted to a channel	channels:history	

[Add Bot User Event](#)

Request URL 입력 후 Subscribe to bot events 에서 'add Bot User Event' 를 클릭한 뒤 message.im, message.channels 를 추가합니다.

6. 슬랙 앱 설치하기

슬랙 API 사이트의 전역 메뉴에서 Settings > Install App 페이지로 접속한 뒤 'Install App to Workspace' 를 클릭하여 앱을 설치합니다.

Install App to Your Team

Install your app to your Slack workspace to test your app and generate the tokens you need to interact with the Slack API. You will be asked to authorize this app after clicking **Install App to Workspace**.

Install App to Workspace

강남언니의 회사이름인 'healingpaper' 로 생성한 Workspace 에 TestSlackBot 을 설치하였습니다.

TestSlackBot is requesting permission to access the healingpaper Slack workspace



What will TestSlackBot be able to view?



Content and info about channels & conversations



Cancel

Allow

설치 후 해당 페이지에 Bot User OAuth Access Token 값이 표시됩니다.

OAuth Tokens for Your Team

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

Bot User OAuth Access Token

xoxb-4.

Copy

Reinstall App

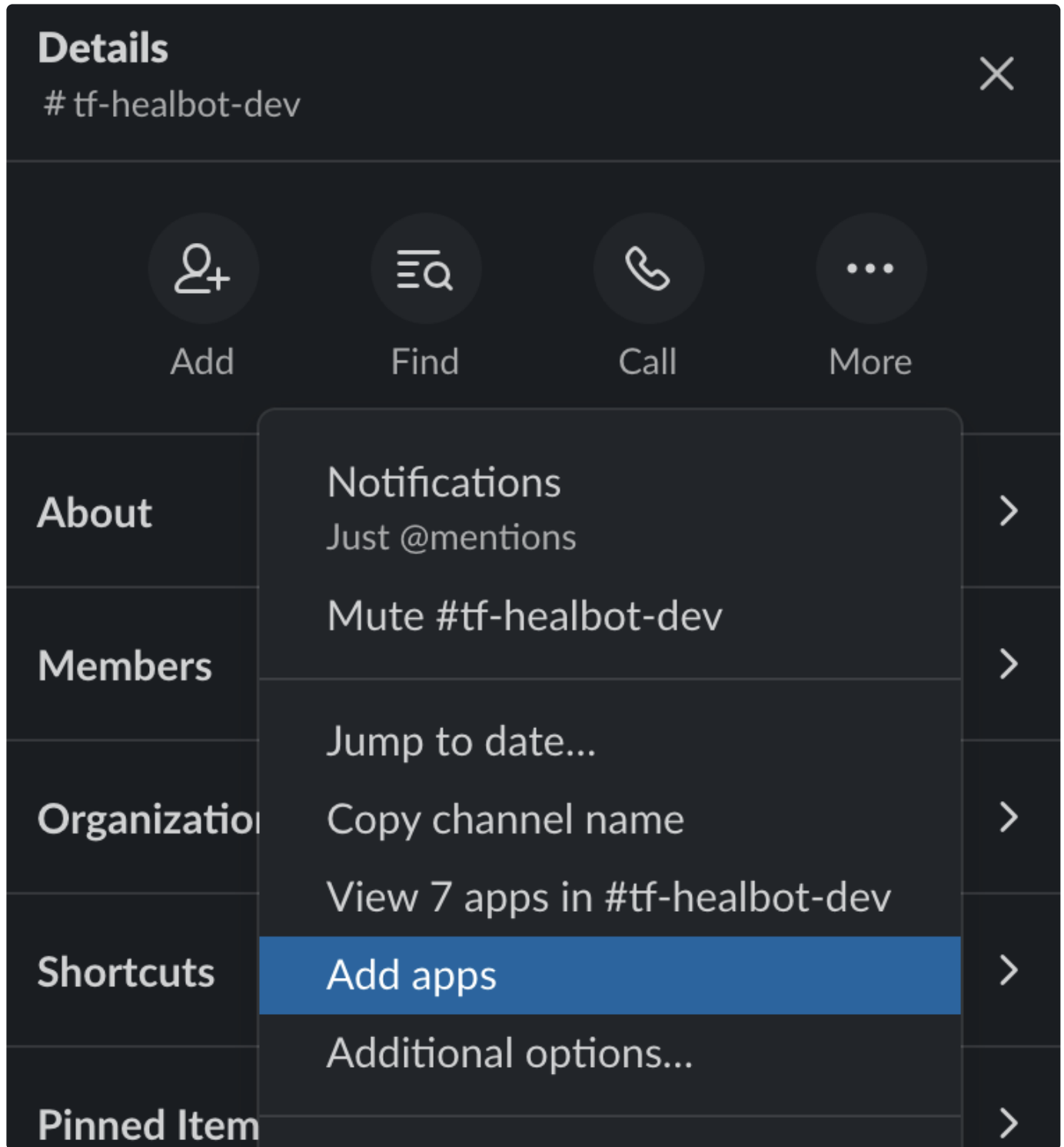
config/bot.json 파일을 열어서, Bot User OAuth Access Token 값을 복사한 뒤 아래와 같이 붙여넣기 합니다.

```
// config/bot.json
// XXXX 에 복사한 Signing Secret 값으로 덮어쓰기 합니다.

{
  "SIGNING_SECRET": "XXXX"
  "BOT_USER_OAUTH_ACCESS_TOKEN": "xoxb-XXXX"
}
```

7. 슬랙 앱으로 메시지 받아보기

슬랙을 실행한 뒤, Workspace 에 접속 후 메시지를 구독하고 싶은 채널에 슬랙 앱을 설치합니다. 채널의 Details 메뉴에서 More 를 클릭한 뒤 Add apps 를 통해서 슬랙 앱을 설치할 수 있습니다.




아래와 같이 슬랙 앱 리스트가 나와면 검색 혹은 스크롤을 통해 슬랙 앱을 Add 합니다.

Add apps to # tf-healbot-dev


View App Directory

Q TestSlackBot

In your workspace


 **TestSlackBot**

Add

 **Slack Developer Tools (Legacy)**
Learn, test, and inspect the Slack Platform

View

슬랙 앱이 해당 채널에 추가되면 추가한 유저에 의해 설치 됐다고 나옵니다.

 **testslackbot** APP 6:30 PM
was added to #tf-healbot-dev by Mark.

슬랙 앱을 재실행합니다.

```
# Ctrl + c 를 눌러 슬랙봇을 종료한 뒤, 다시 실행합니다.  
$ npx ts-node ./src/index.ts
```

슬랙 앱이 설치된 채널에서 메시지를 입력하게 되면,

 **Mark** 6:31 PM
메시지 입력 테스트

아래와 같이 터미널에 event 값이 찍히게 됩니다.

```
run slack bot
{ client_msg_id: '74d3a077-c64f-4b34-84bd-4e041342273d',
  type: 'message',
  text: '메시지 입력 테스트',
  user: '',
  ts: '1589103132.149300',
  team: '',
  blocks:
    [ { type: 'rich_text', block_id: 'kBoJv', elements: [Array] } ],
  channel: '',
  event_ts: '1589103132.149300',
  channel_type: 'channel' }
```

8. 슬랙 앱으로 메시지 보내기

원하는 채널로 메시지를 보내기 위해 web-api 모듈을 설치합니다.

```
$ npm install --save @slack/web-api
```

webClient 인스턴스를 생성하고 webClient.chat.postMessage 를 통해 메시지를 보냅니다.

message 구독 콜백 코드 안에 '?하이' 를 입력 받으면, '안녕하세요!' 라고 보내는 코드를 추가합니다.

```
// 업데이트 된 src/index.ts

import express from 'express';
import { WebClient } from '@slack/web-api';
import { createEventAdapter } from '@slack/events-api';
import { createServer } from 'http';

// 생성한 슬랙앱에 대한 키값들
import CONFIG from '../config/bot.json';
/*
{
  "SIGNING_SECRET": "XXXX",
  "BOT_USER_OAUTH_ACCESS_TOKEN": "xoxb-XXXX"
}
*/
```

```
// 슬랙에서 슬랙앱에게 접근가능한 엔드포인트를 만들기 위해 웹서버(express)를 사용
const app = express();

const slackEvents = createEventAdapter(CONFIG.SIGNING_SECRET);
const webClient = new WebClient(CONFIG.BOT_USER_OAUTH_ACCESS_TOKEN);

// 메시지 이벤트 구독하기
slackEvents.on('message', async (event) => {
  console.log(event);

  if (event.text == '?하이') {
    webClient.chat.postMessage({
      text: '안녕하세요!',
      channel: event.channel,
    });
  }
});

// 메시지 이벤트 엔드포인트를 express 에 등록하기
app.use('/slack/events', slackEvents.requestListener());

// express 웹 서버 실행
createServer(app).listen(3000, () => {
  console.log('run slack bot');
});
```

슬랙 앱을 재실행합니다.

```
# Ctrl + c 를 눌러 슬랙봇을 종료한 뒤, 다시 실행합니다.
```



강남언니 공식 블로그






다시 슬랙 API 사이트의 왼쪽 메뉴에서 Features > Event Subscriptions 페이지로 이동하여 scopes 에 chat:write 를 추가합니다.

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	
channels:history	View messages and other content in public channels that TestSlackBot has been added to	
chat:write	Send messages as @testslackbot	
im:history	View messages and other content in direct messages that TestSlackBot has been added to	



Add an OAuth Scope

Scopes 가 변경되어, 앱 설치를 다시 하라고 나올 때 다시 설치 해 줍니다.



TestSlackBot is requesting permission to access the healingpaper Slack workspace



What will TestSlackBot be able to view?

-  Content and info about channels & conversations 

What will TestSlackBot be able to do?

-  Perform actions in channels & conversations 

Cancel

Allow

앱 재설치 후 해당 채널에서 '?하이' 를 입력하면 원하는 메시지가 나오게 됩니다.



Mark 6:47 PM
?하이



testslackbot APP 6:47 PM
안녕하세요!

앱이 실행된 터미널에서는 아래와 같이 로그가 찍히게 됩니다

```

run slack bot
{ client_msg_id: 'f4fd02bc-db90-4678-a8d8-33de88577398',
  type: 'message',
  text: '?하 이 ',
  user: ' ',
  ts: '1589103667.150000',
  team: ' ',
  blocks:
    [ { type: 'rich_text', block_id: 'IXYs2', elements: [Array] } ],
  channel: ' ',
  event_ts: '1589103667.150000',
  channel_type: 'channel' }
{ bot_id: ' ',
  type: 'message',
  text: '안녕하세요!',
  user: ' ',
  ts: '1589103668.150100',
  team: ' ',
  bot_profile:
    { id: ' ',
      deleted: false,
      name: 'TestSlackBot',
      updated: 1589102819,
      app_id: ' ',
      icons:
        { image_36: 'https://a.slack-edge.com/80588/img/plugins/app/bot_36.png',
          image_48: 'https://a.slack-edge.com/80588/img/plugins/app/bot_48.png',
          image_72:
            'https://a.slack-edge.com/80588/img/plugins/app/service_72.png' },
      team_id: ' ' },
  channel: ' ',
  event_ts: '1589103668.150100',
  channel_type: 'channel' }

```

여기서 작성한 슬랙 앱 코드는 아래의 깃 저장소에 올려놨습니다.

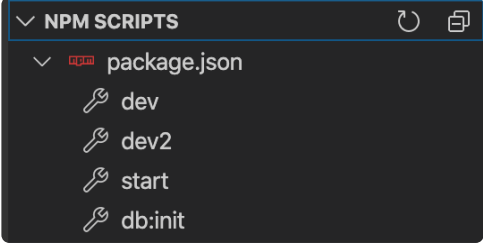
<https://github.com/hurest/typescript-slack-bot/tree/step01>

10. 아쉬운 점

슬랙이 기존 슬랙 봇에서 서드파티 앱으로 방식을 바뀌가고 있어, 설명에는 앱으로 표현하고 있습니다. 기존 슬랙은 채널에 사용자와 동등한 위치로 봇이 추가되는 방식이었으나, 지금의 슬랙은 채널에 사용자와 앱을 분리하여 채널에 앱을 설치하는 방식으로 바꿨습니다. 참고 바랍니다.

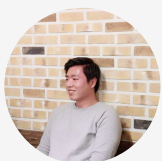
설명에는 ts-node 를 사용하였으나, ts-node-dev 를 사용하면 코드 수정 시 자동으로 앱을 리로드 시킬 수 있습니다.

package.json 파일 안에 scripts 필드 안에 명령어를 추가하면 vscode 에서 클릭 형태로 앱을 실행 할 수 있습니다.

VSCODE NPM SCRIPT	package.json
	<pre> }, "scripts": { "dev": "ts-node-dev ./src/index.ts --dev", "dev2": "ts-node-dev ./src/index.ts --dev2", "start": "ts-node-dev ./src/index.ts --prod", "db:init": "ts-node-dev ./src/db/init/index.ts --dev" }, "repository": { "type": "git", </pre>

11. 다음편

- 타입스크립트로 슬랙 봇 만들기 - 응용편
슬랙에서 제공해주는 다양한 API (모달 및 입력폼, 메시지 타입 등) 을 알아볼까 합니다.
- 타입스크립트로 슬랙 봇 만들기 - 리팩토링편
보다 간편하게 명령어를 추가할 수 있는 구조에 대해서 알아볼까 합니다.
- 타입스크립트로 슬랙 봇 만들기 - 번역편
강남언니에서 슬랙으로 외국인들과 대화하는 방법에 대해서 알아볼까 합니다.
- 타입스크립트로 슬랙 봇 만들기 - 점심팀편
강남언니는 슬랙으로 점심팀을 어떻게 구현하였는지 알아볼까 합니다.



Mark

Web Front-end Developer

웹앱으로 만들고 앱으로 느끼게 하는 거에 관심이 많은 자바스크립트 개발자입니다. 미래의 UI는 웹이라 믿고 있습니다.

< 이전 글

다음 글 >

 강남언니

©Healing Paper Co., Ltd. All rights reserved.

