

Visual Studio Code

+



Firebase

Visual Studio Code

플러그인 만들기

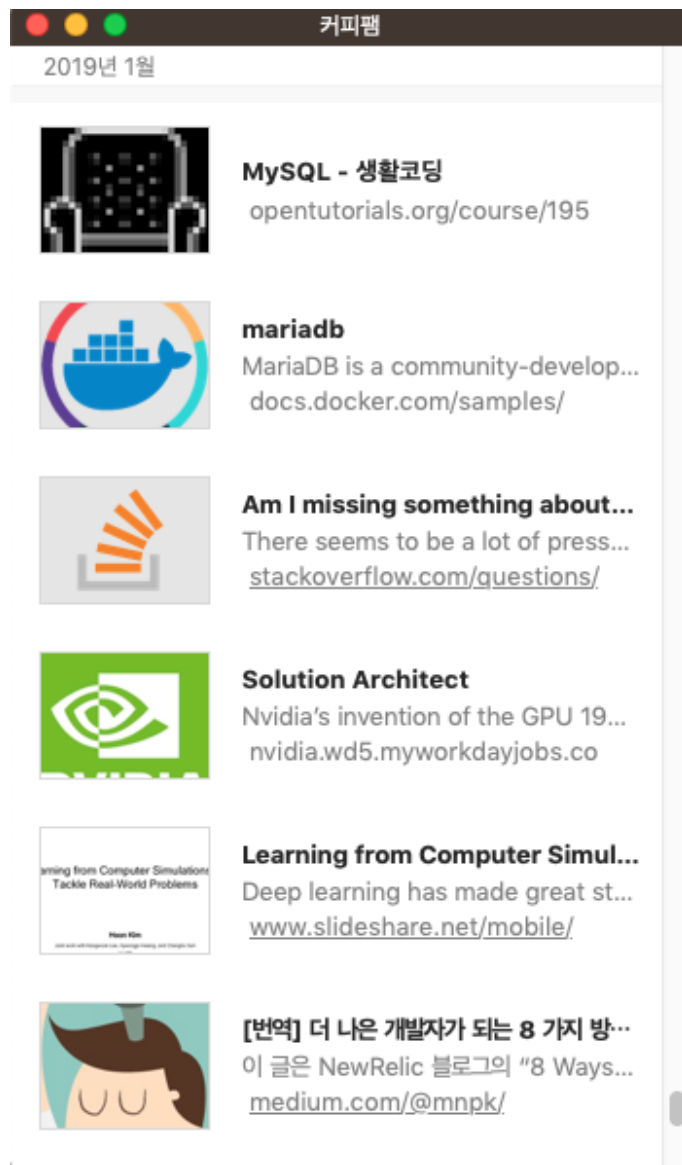
나



아직 연약한 신입
하지만 사수가 없다
코딩은 해야함
자기 개발도 해야함



뭐라도 만들자



마침 어떤 발표자료를 봄 (좋은 회사를 가려면)

1. 항상 준비가 되어 있어야함 ! (이력서, 포트폴리오 등)
2. 오픈소스 기여하면 좋아함 ! -> (본인 성장이 회사의 발전이라는 것을 암)

ex) 취업 준비 하면서 우대사항에 있는 TDD 공부한 자료 올리고 컨퍼런스에서 발표해서 취업함

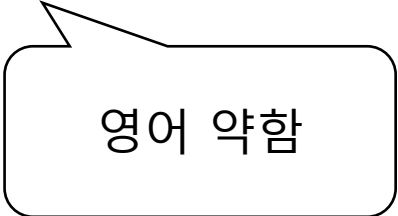
흔한 전 직장 개발자 단톡

오픈소스... 뭐가 있을까 ?

MS사 꺼를 많이 쓰는데 기여 해볼까...

맨날 구글 번역기 왔다갔다하는데

번역 하는 플러그인 만들어볼까 ?



영어 약함

VSCODE Extension 만든 삽질기

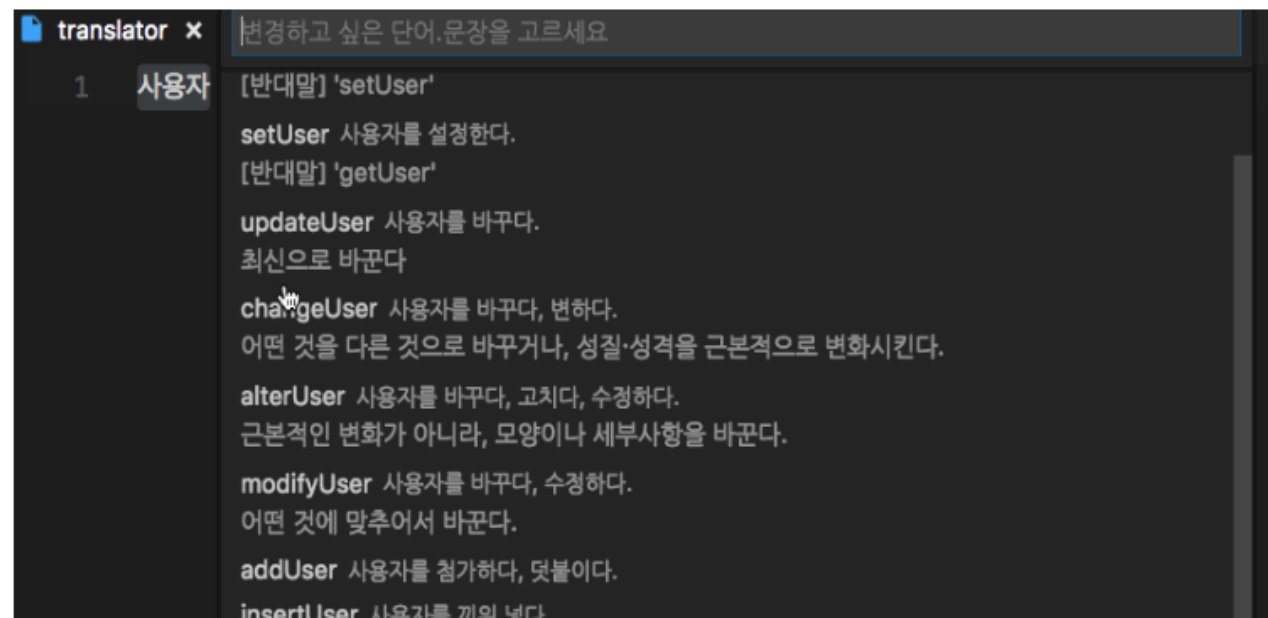
Dec 27, 2017 in Tech

MS에 기여한게 없어서 뭔가 해봐야겠다는 생각에
그동안 만들고 싶었던 vscode extensions을 만들어 봤다.



<https://marketplace.visualstudio.com/items?itemName=sculove.translator>

한글을 번역할 때는 번역된 텍스트에 적용할 수 있는 접두사가 적용된 메서드의 이름을 보여주는 기능도 넣었다.



하지만 이미 있음.. 개발자들은 역시 불편한 걸 못 참음

그러면 따라 만들면서 공부도 해보고
기능을 추가하고 내 입맛대로 만들자 !!

더 설명 잘하면 되겠지
(블로그 시작)

사전 조사

Vscode 는 Electron 으로 만들어졌다.

Electron은 Node.js 기반



Electron 이란?

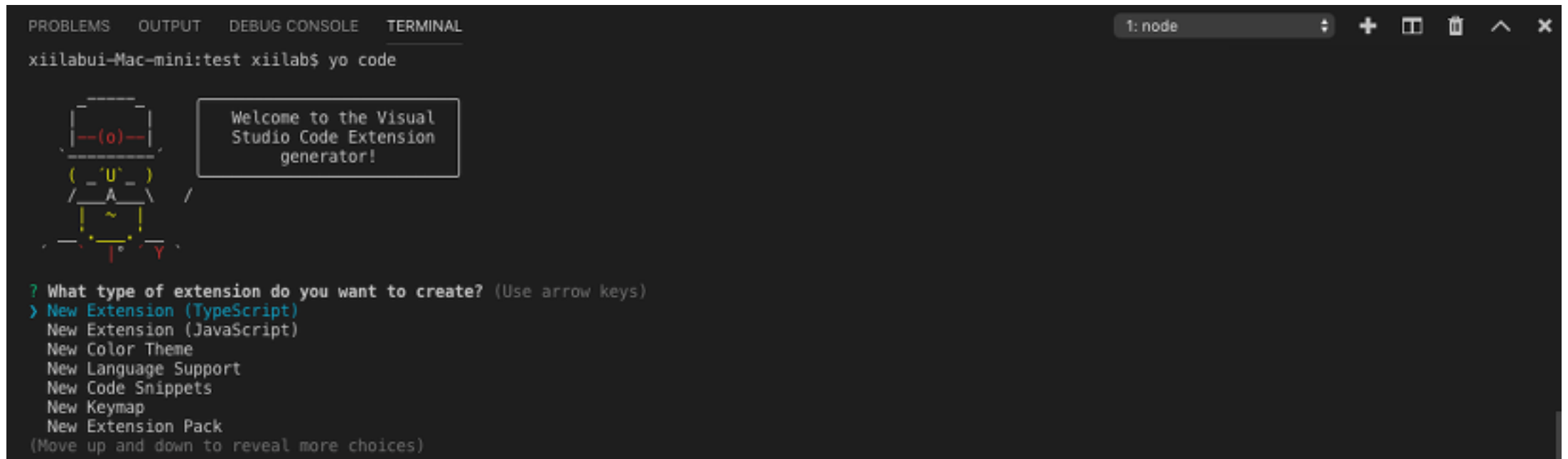
Electron 이란 NodeJS 와 Chromium (일반적으로 Chrome 의 소스 로 이해하시면 편함 : [정보](#)) 을 조합하여 웹 서버 없이도 PC 에 서 웹 서버 + 브라우저 인 것 처럼 구동하는 데스크탑 앱을 만들수 있는 툴을 말합니다.

Chromium 기반으로 하기때문에 Chromium 지원하는 OS 들을 지원하는 크로스 데스크탑 앱 플랫폼 이라고 생각하시면 편합니다.
대표적으로 Slack, Jandi, discord, Atom, Visual Studio Code 와 같은 앱들의 데스크탑 버전들이 Electron 을 이용하며 지금도 수 많 은 앱들이 개발되고 있습니다.

- 정보 : <http://electron.atom.io/apps/>

```
npm install -g yo generator-code
```

Visual Studio Code 플러그인 개발을 위한 모듈



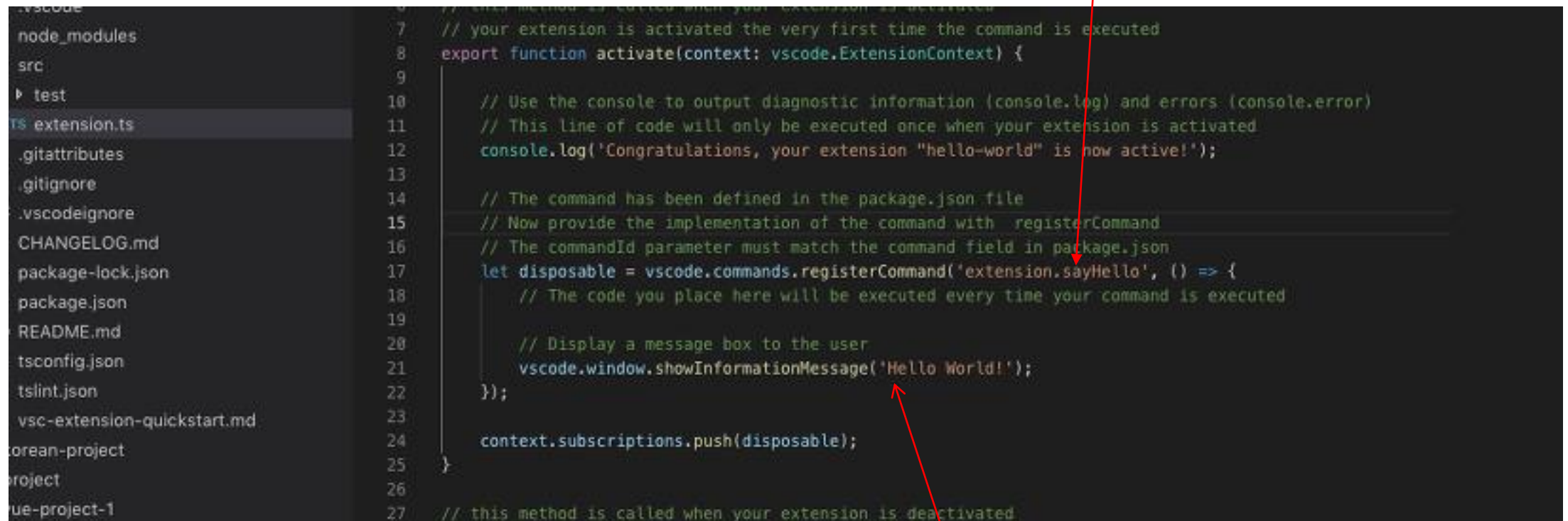
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
xiilabui-Mac-mini:test xiilab$ yo code

Welcome to the Visual
Studio Code Extension
generator!

? What type of extension do you want to create? (Use arrow keys)
> New Extension (TypeScript)
  New Extension (JavaScript)
  New Color Theme
  New Language Support
  New Code Snippets
  New Keymap
  New Extension Pack
(Move up and down to reveal more choices)
```

연습 1

Package.json 파일에서
extension.sayHello 라는 명령어 가 실행되면



The screenshot shows the file explorer on the left with 'extension.ts' selected. The editor displays the following code:

```
6 // this method is called when your extension is activated
7 // your extension is activated the very first time the command is executed
8 export function activate(context: vscode.ExtensionContext) {
9
10     // Use the console to output diagnostic information (console.log) and errors (console.error)
11     // This line of code will only be executed once when your extension is activated
12     console.log('Congratulations, your extension "hello-world" is now active!');
13
14     // The command has been defined in the package.json file
15     // Now provide the implementation of the command with registerCommand
16     // The commandId parameter must match the command field in package.json
17     let disposable = vscode.commands.registerCommand('extension.sayHello', () => {
18         // The code you place here will be executed every time your command is executed
19
20         // Display a message box to the user
21         vscode.window.showInformationMessage('Hello World!');
22     });
23
24     context.subscriptions.push(disposable);
25 }
26
27 // this method is called when your extension is deactivated
```

Two red arrows point from the text above to the code: one points to the 'extension.sayHello' command ID in line 17, and the other points to the 'vscode.window.showInformationMessage' call in line 21.

Hello World! <- 메시지창에 띄워라

package.json

You, a few seconds ago | 1 author (You)

```
1  {
2    "name": "korean-translator",
3    "displayName": "korean-translator",
4    "description": "",
5    "version": "0.0.1",
6    "engines": {
7      "vscode": "^1.29.0"
8    },
9    "categories": [
10     "Other"
11   ],
12   "activationEvents": [
13     "onCommand:extension.sayHello"
14   ],
15   "main": "./out/extension",
16   "contributes": {
17     "commands": [
18       {
19         "command": "extension.sayHello",
20         "title": "Hello"
21       }
22     ]
23   },
```

You, a few seconds ago • Uncommitted changes

extension.ts

package.json



```
1  {
2    "name": "wordcount",
3    "displayName": "WordCount",
4    "description": "",
5    "version": "0.0.1",
6    "engines": {
7      "vscode": "^1.29.0"
8    },
9    "categories": [
10     "Other"
11  ],
12  "activationEvents": [
13    extension.sayHi 로 변경
14    "onCommand:extension.sayHi"
15  ],
16  "main": "./out/extension",
17  "contributes": {
18    "commands": [
19      {
20        extension.sayHi 로 변경
21        "command": "extension.sayHi",
22        "title": "Hi"
23      }
24    ]
25  },
```

기능정의

1. 선택 영역 번역
2. 번역기는 카카오 API - 내가 카카오 가고싶으니깐..

1. 선택 영역 번역

```
'use strict';

import {
  window as vscodeWindow,
  commands,
  ExtensionContext,
  Range,
} from 'vscode';

export function activate(context: ExtensionContext) {

  console.log('Congratulations, your extension "hello-world" is now active!');

  let disposable = commands.registerCommand('extension.sayHello', () => {

    const editor = vscodeWindow.activeTextEditor;
    console.log('vscode의 글씨들', editor);
    if (!editor) {
      return;
    }
    const selections = editor.selections[0];

    console.log('select 된거 정보', selections);
    const selection_range = new Range(selections.start, selections.end);
    console.log('select 된 라인 정보', selection_range);

    const text = editor.document.getText(selection_range);
    if (!text) {
      console.log('text 선택됨...');
    }
  });
}
```

Vscode 안에 글씨 정보가 보임

어디~ 부터 어디~ 까지의 범위를 구해서 단어 획득
ex) 0번째 줄 2번째 부터 0번째 줄 5번째 까지

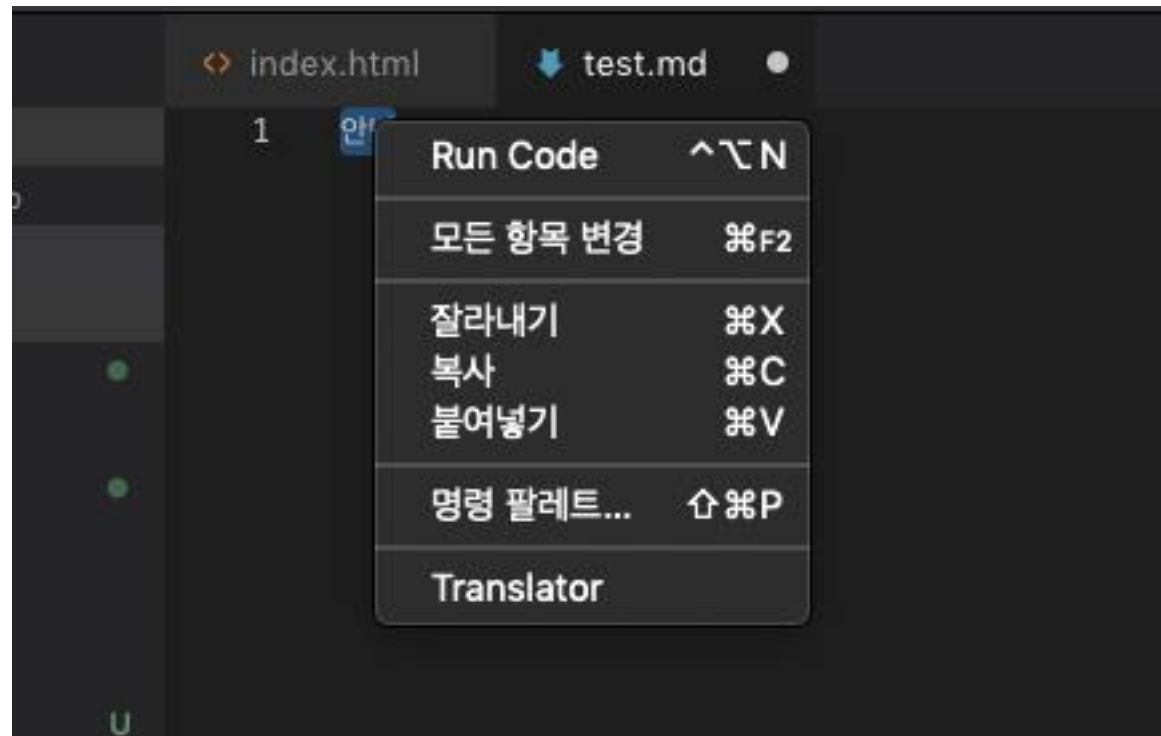
언제까지 sayHello 로 돌 순 없죠

```
let disposable = commands.registerCommand('extension.translateKorean', () => {
```

```
},
"activationEvents": [
  //커맨드 등록
  "onCommand:extension.translateKorean"
],
"main": "./out/extension",
"contributes": {
  "menus": {
    // 단어 선택 후 오른쪽 버튼 클릭하면 command translateKorean 실행
    "editor/context": [
      {
        "command": "extension.translateKorean",
        "when": "editorHasSelection"
      }
    ]
  },
  "commands": [
    {
      // 단어 클릭시 보여줄 타이틀은 Translator 이다
      "command": "extension.translateKorean",
      "title": "Translator",
      "when": "editorHasSelection"
    }
  ]
}
```

선택되었을 때

선택하면 내가 설정한 명령어 나오게끔 완성 !

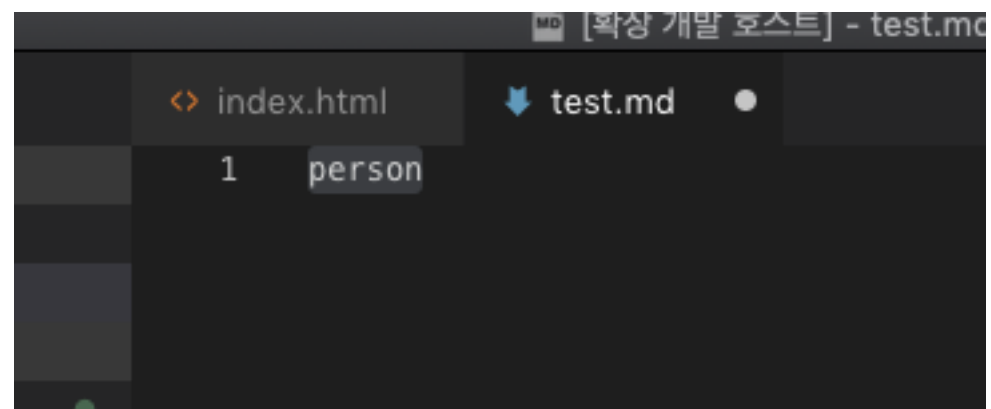
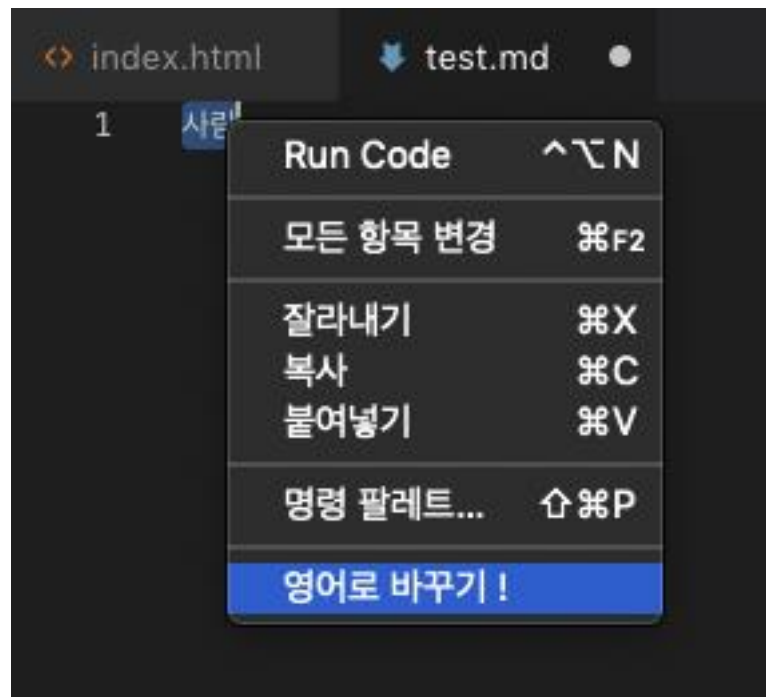


2. 번역기는 카카오 API

```
1function translationText() {
2    console.log('-----testCode-----');
3
4    const src_lang = 'kr';
5    const target_lang = 'en';
6    let query = text;
7    const headers = {
8        'Authorization': 'KakaoAK ~~~~~개인 KEY 자리 ~~~~~',
9        'Content-Type': 'application/json;charset=utf-8',
10    };
11    query = encodeURIComponent(query);
12    return fetch(
13        //번역기 사용하는 API
14        `https://kapi.kakao.com/v1/translation/translate?src_lang=${src_lang}&target_lang=${target_lang}&query=${query}`, {
15            method: 'GET',
16            headers: headers
17        }).then(
18            (res: Response) => {
19                console.log('res-----', res);
20                return res.json();
21            }
22        ).then(resJson => {
23            console.log(resJson);
24            return resJson.translated_text[0];
25        })
26        .catch(err => console.log(err));
27 }
```

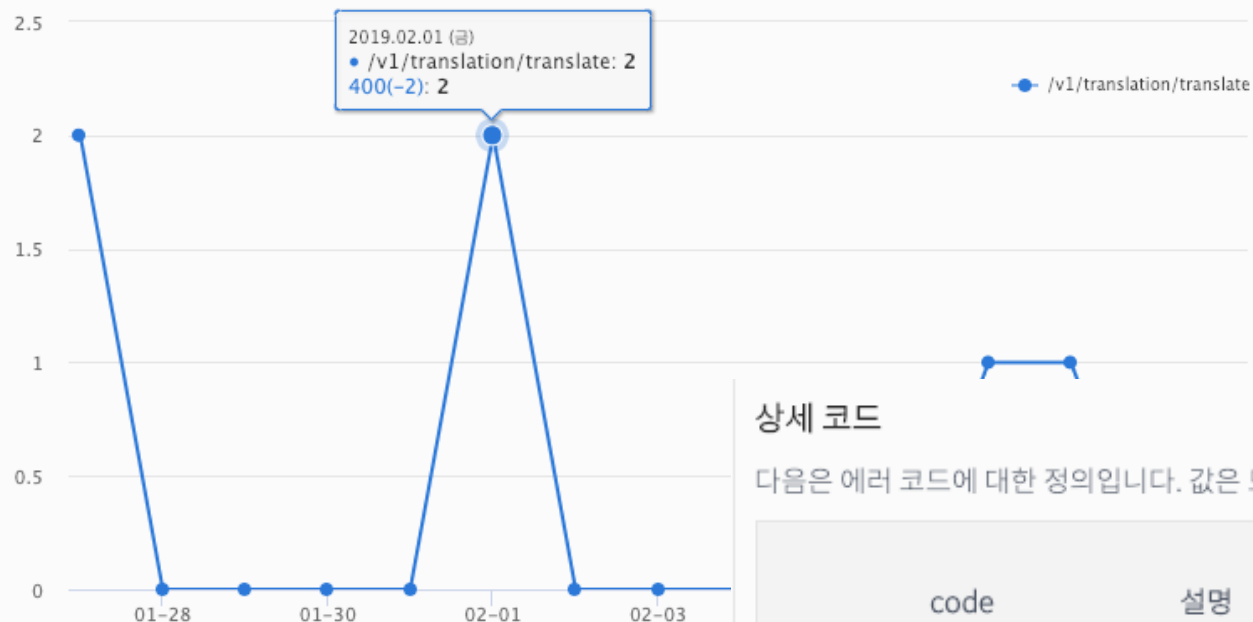
원하는 언어, 번역되어야하는 언어, 텍스트를 보냅니다.

카카오한테요 ~



API 에러 응답 상태(에러코드) 확인 ▾

CSV



상세 코드

다음은 에러 코드에 대한 정의입니다. 값은 모두 음수로 표현됩니다.

code	설명	HTTP 상태 코드
-1	내부 처리 오류. 에러 코드로 상세 분류되어 있지 않은 경우	400
-2	잘못된 파라미터. 호출 인자값이 잘못되었거나 필수 인자가 포함되지 않은 경우	400
-3	지원되지 않는 서비스 API에 대한 호출. 해당 앱의 호출된 API가 설정에서 off 되어 있을 경우	400
-4	계정 제재 또는 특정 서비스에서 해당 사용자의 제재로 인해 API 호출이 금지된 경우	400

에러가 뜨는데 이유를 알고 싶다.

왜지 ?

왜 ????

뭘 검색한거야

확인하는 방법은 ... 검색하는걸 디비에 저장하는 방법 뿐 !



그런데... 어떻게 저장하지 ?

DEVELOP



Realtime Database



Authentication



Cloud Messaging



Storage



Hosting



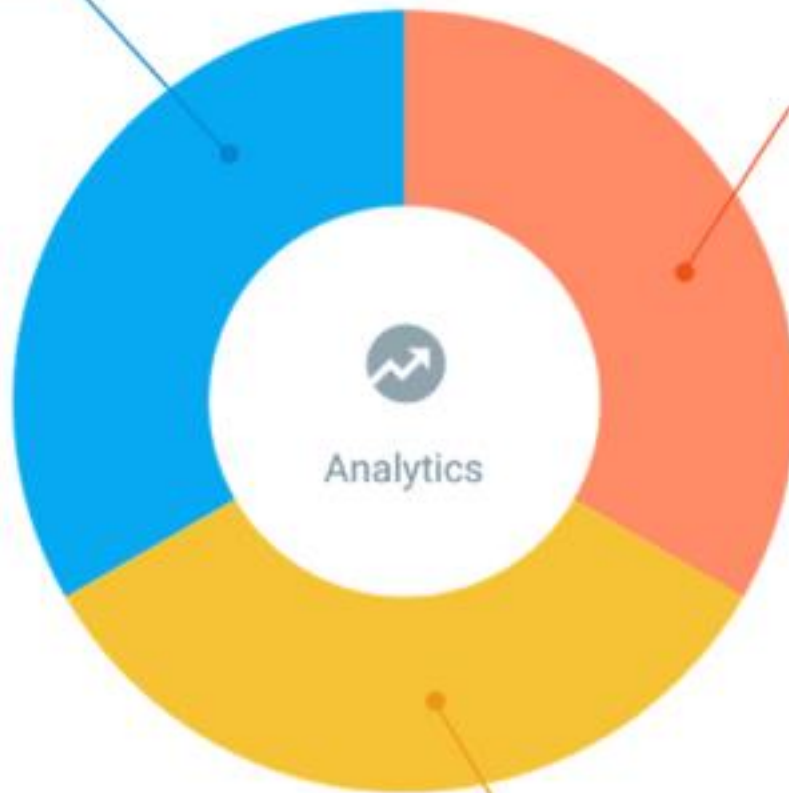
Remote Config



Test Lab



Crash Reporting



GROW



Notifications



App Indexing



Dynamic Links



Invites



AdWords

EARN



AdMob

[가이드](#)
[프로젝트 설정 및 관리](#)
[애널리틱스](#)
[개발](#)
[인증](#)
[실시간 데이터베이스](#)
[소개](#)
[데이터베이스 선택](#)
[iOS](#)
[Android](#)
[웹](#)
[시작하기](#)
[데이터 구조화](#)
[데이터 읽기 및 쓰기](#)
[데이터 목록 다루기](#)
[오프라인 기능 사용 설정](#)
[관리자](#)
[시작하기](#)
[데이터 구조화](#)
[데이터 저장](#)
[데이터 검색](#)
[REST](#)
[C++](#)
[Unity](#)
[보안 및 규칙](#)
[사용 및 성능](#)

이 문서에서는 Firebase 실시간 데이터베이스에 데이터를 쓰는 4가지 방식인 설정, set, update, push, transaction 지원에 대해 설명합니다.

데이터 저장 방법

set	정의된 경로(예: <code>messages/users/<username></code>)에 데이터를 쓰거나 대체합니다.
update	정의된 경로에서 모든 데이터를 대체하지 않고 일부 키를 업데이트합니다.
push	데이터베이스의 데이터 목록에 추가합니다. 목록에 새 노드를 푸시할 때마다 데이터베이스에서 고유 키(예: <code>messages/users/<unique-user-id>/<username></code>)를 생성합니다.
transaction	동시 업데이트에 의해 손상될 수 있는 복잡한 데이터를 다루는 경우 트랜잭션을 사용합니다.

데이터 저장

기본 데이터베이스 쓰기 작업은 지정된 데이터베이스 참조에 새 데이터를 저장하고 해당 경로의 기존 데이터를 모두 대체하는 `set` 메소드를 이해할 수 있도록 간단한 블로깅 앱을 만들어 보겠습니다. 앱의 데이터는 아래와 같은 데이터베이스 참조에 저장됩니다.

자바 **NODE.JS** PYTHON GO

```
// Import Admin SDK
var admin = require("firebase-admin");

// Get a database reference to our blog
var db = admin.database();
var ref = db.ref("server/saving-data/fireblog");
```

우선 사용자 데이터를 저장해 보겠습니다. 고유한 사용자 이름을 기준으로 각 사용자를 저장하고 전체 이름과 생일을 함께 저장합니다. 각 사용자는 고유한 사용자 이름을 갖게 되므로 별도의 키를 만들 필요가 없기 때문에 push 메소드 대신 set 메소드를 사용합니다.

```
var admin = require('firebase-admin');
var serviceAccount = require('../translator-c4119-firebase-adminsdk-ft76z-126df85c6a');
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://translator-c4119.firebaseio.com",
});
var db = admin.database();
var ref = db.ref('translate');
var successRef = ref.child('success');
var errorRef = ref.child('error');
```

```

        return res.json(),
    } else {
        console.log('실패');
        errorRef.push({
            Error: res.json(),
            Time: time
        });
        vswindow.showInformationMessage('카카오에 문제가 있습니다.');
```

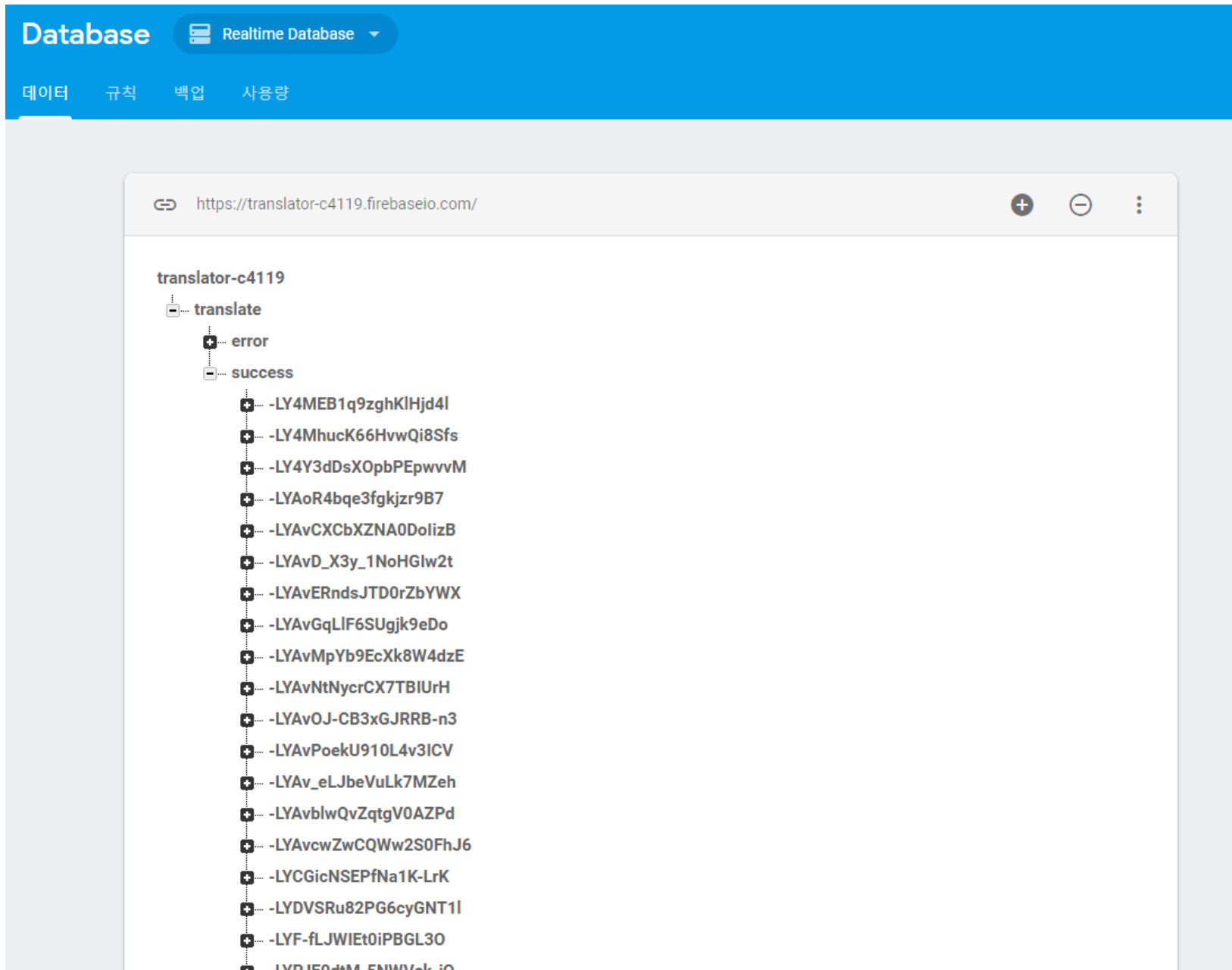
나한텐 문제 없을꺼야 ㅠ

```

    ).then(resJson => {
        console.log(resJson);
        if (resJson.translated_text[0] == '') {
            console.log('번역 불가');
            resJson.translated_text[0] = ['번역이 불가능합니다 ㅠ'];
        }
        console.log(resJson.translated_text[0][0]);
        successRef.push({
            Korean: text,
            English: resJson.translated_text[0][0],
            Time: time
        });
        editor.edit((edit: any) => edit.replace(selection_range, String(resJson.translated_text[0])));

        return resJson.translated_text[0];
    });
}
```

추후 데이터가 어느정도 모이면 시각화 하면 재미있을거 같음 !



Database



Realtime Database

데이터

규칙

백업

사용량



현재 결제 기간

2월 1일~2019년 2월 28일(GMT-8)

연결 ?

5 최대

저장소 ?

12.9KB/1GB

다운로드 ?

1.2MB/10GB

부하 ?

0% 최대



감사합니다.

더 많은 자료는 블로그, 깃에 있습니다.

(홍보)

<https://kdinner.tistory.com/>

<https://github.com/DinnerKang>