

[| 자유 게시판 | >](#)

[스압] 사람들은 node-kakao를 사용하고 싶어하면서 왜 못 하는가

윤 챗봇 입문자 1:1 채팅

2020.07.10. 14:55 조회 517

[댓글 31](#) [URL 복사](#) [⋮](#)

안녕하세요. 윤입니다.

요즘 글에 댓글을 쓰려면 자꾸 등급 제한되어 눈팅만 하다가 드디어 글을 씁니다.

사실 이 주제는, node-kakao 가 퍼블릭이 되기 전 부터 쓸까말까 했던 주제입니다.

당시 카페에서 되게 민감한 문제였으니까요.

근데 이제 레포가 퍼블릭으로 되면서 오픈 채팅방을 보는데 어떻게 보는 것만으로도 사람의 화를 돋구는지, 정말 신기할 따름이라 이곳에 적습니다.

제가 화나는 것은 다음과 같습니다.





"후웅... 노드 카카오가 요즘 그렇게 유명하던데, 뭣?! Node JS 로 만들 수 있다고? 알림을 읽을 뿐인 다른 것과 달리 LOCO 를 사용하잖아? 나도 사용해야지."

좋습니다. 새로운 방법에 대한 탐구는 개발자로서 매우 환영하는 일입니다.
하지만 코뿔소에서 돌아가는 JS 도 제대로 사용하지 못 하는 분들이 태반이더군요. 그런데 이런 분들이 V8 에서 돌아가는 정식 JS 를 사용한다?

NPM 과 노드의 기본 프로젝트 구성도 모르는 분들이 사용한다? 당연히 어렵습니다.
기어다니는 얘기가 달릴 수 없는 법과 같죠.

- "너무 어려운 거시네요. 저에게 LOCO 를 알려주시와요."
- "노드 카카오 어떻게 사용하나요?"
- "저도 로코가 하고 싶어요. 알려주세요."

라고 질문을 많이 합니다. 물론, "제가 ~~ 했는데 여기서 ~~ 하면서 안 돼요. ~~ 하려면 어떻게 해야할까요?" 라고 정중하고 자세히 묻는다면 당연히 아는 선에서 대답해드립니다.

근데 TML (Teach Me Loco) 빌런들은 일단 입을 벌리고 누군가 본인의 입에 로코를 쑤셔넣어주기 전 까지 기다립니다.



저는 제 화를 다스리지 못 하여, 이곳에 글을 쓰게 됩니다.
이 글에 대한 제제는 카페 스텝분들께서 판단하에 잘 처분해 주세요.

서론이 끝났습니다. 목차를 알아봅시다.

- □ Node.JS 와 NPM
- □ Github Package Registry
- □ Javascript 그리고 Typescript

목차는 3개지만 이 내용들은 전부 앞으로 여러분들이 개발자로 지내면서 살아가는 한, 그리고 특히 Javascript 를 사용하는 개발자인 이상 평생 도움이 될만한 내용들입니다.

□ Node.JS 와 NPM



HugeServer

- Node JS

원래 자바스크립트라는 것은 웹브라우저에서, 웹 페이지를 동적으로 보여주기 위해 고안된 언어입니다. 웹 개발 이외엔 쓸모도 없던 언어가 이 NodeJS 가 탄생하면서 언어의 위용을 뒤집어버립니다.

NodeJS 는 브라우저에서 자바스크립트가 런타임 중 실행되는 V8 엔진과 같이 사용하여, 브라우저가 아닌 환경에서 실행되는 런타임 프레임워크입니다.

많은 분들이 "프로그래밍 언어 무엇을 사용하시나요?" 라고 물으면 "노드를 사용합니다." 대답하시는데, 이는 잘못된 대답입니다. 노드는 언어가 아니라 런타임 프레임워크이니까요!

- NPM

NPM 은 Node Package Manager 의 약자입니다. Python 의 PIP와 비슷하거나 같은 존재입니다. 이 존재가, Node JS 의 위상을 더 높히는 데 엄청난 몫을 하였습니다.

원하는 기능이 있으면 일단 npm 에서 검색해 봅니다. 있으면 패키지를 설치합니다. 추가해서 라이브러리 사용하듯 하면 끝입니다.

이러니 개발 속도는 엄청나게 빨라질 수 있었고, 덕분에 사람들은 노드를 사랑하게 되었죠.

하지만 node_modules 폴더에 집중적으로 모여지는 패키지 관리는 썩 좋지 못 했습니다. 그렇게 만들어지게 된 것이 deno 입니다.

뭐, 옛날 얘기는 이쯤 하고 npm 을 사용하는 방법에 대해서 넘어가도록 하죠.

사실 노드 프로젝트를 시작하는 것은 어렵지 않습니다.

폴더를 하나 만들고, 해당 폴더에서 "npm init -y" 만 입력하면 프로젝트 세팅이 끝납니다.

NPM 의 단점 중 하나가 node_modules 이듯, 종속성이 걸린 패키지가 많아지면 많아질수록 용량이 커지게 됩니다. 그럼, 실행 가능한 프로젝트를 다른 사람과 공유하기 위해선 이 node_modules 폴더까지 공유해야 할까요? 용량이 XXMB 가 되 스드 이느 거요?

만약 그랬다면 오늘날의 노드는 존재하지 않았겠죠. 노드 프로젝트를 관리하는 `package.json` 에 담긴 정보를 가지고, 종속성이 걸린 패키지를 설치합니다.

우리는 `node-kakao` 에 대한 얘기중이었으니, 해당 프로젝트를 기준으로 설명하겠습니다.

```
"devDependencies": {
  "@types/bson": "^4.0.0",
  "@types/chai": "^4.2.3",
  "@types/mocha": "^5.2.7",
  "@types/node": "^12.7.11",
  "@types/node-forge": "^0.8.6",
  "@types/request-promise": "^4.1.44",
  "chai": "^4.2.0",
  "mocha": "^6.2.1",
  "ts-node": "^8.4.1",
  "typescript": "^3.6.3"
},
"dependencies": {
  "bson": "^4.0.2",
  "json-proxy-mapper": "^1.1.1",
  "lossless-json": "^1.0.3",
  "node-forge": "^0.9.1",
  "request": "^2.88.0",
  "request-promise": "^4.2.4"
}
```

다른건 내버려두고 이 부분만 봅시다.

이 내용은, 해당 프로젝트를 이용하기 위해선 위의 모듈들이 필요하다는 얘기입니다.

그러면 모듈을 설치해야겠죠?

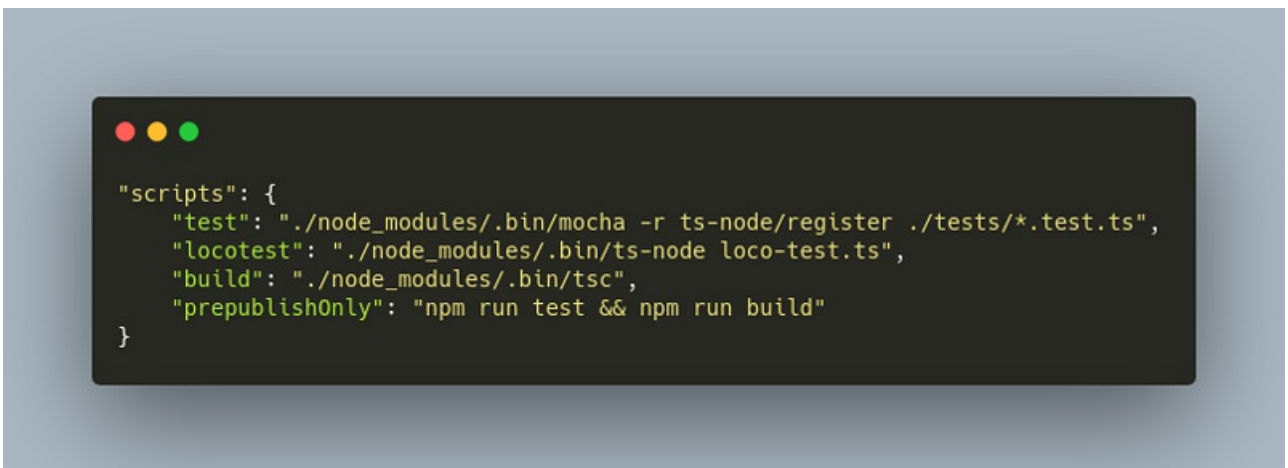
모듈을 설치도 안 하고 무작정 끌어와서 쓰겠다니 제대로 동작 안 하겠죠? 설치부터 합시다.



허무할 만큼 간단합니다. 근데 이 간단한 것을 못 하는 사람들이 많습니다.
이해 할게요. 모르면 그럴 수 있으니까.

여러 종속성이 걸린 패키지를 설치했습니다. 그러면 이제 무엇을 하나요?
그저 깃헙에서 클론한 프로젝트를 가지고 src 폴더를 모듈로 불러올까요?

이런 멍청한! 잘못된 발상입니다. 일단, 무엇을 할 수 있는지에 대한 스크립트도 package.json에 있습니다.



package.json 에 scripts 객체는, npm run [Key] 로 실행시킬 수 있는 스크립트입니다.
자, 대충 보면 우리는 어떤 것을 실행시켜야 할까요?

test: ./tests 폴더에 .test.ts 확장자가 붙은 스크립트를 실행하는 것 같습니다. 모카가 뭔지, ts-node/register 가 뭔지는 중요하지 않습니다. 일단 내용을 보면 알듯, 테스트하는 것 같습니다. 우리 코드를 테스트할 게 아니라 사용할 것이니 필요하지 않습니다.

locotest: loco-test.ts 스크립트를 실행시킨다고 합니다. 근데 정작 loco-test.ts 파일은 존재하지 않습니다.

build: tsc를 실행시킨다고 합니다. 저는 이게 뭐하는지 알고 있지만, 모르는 입장에선 감도 잡히지 않습니다. 하지만 key에서 볼 수 있듯 코드를 빌드하는 것 같습니다.

prepublishOnly: test 와 build 를 실행시킵니다. 우리 아직까지 테스트엔 관심 없습니다.


그러면, 코드를 빌드하기 위해선 우린 `npm run build` 를 실행시켜야 한다는 사실을 알았습니다. 실행시켜보면 프로젝트 폴더에 `dist` 폴더가 새로 생깁니다.

우린 이제 이 모듈을 사용하면 됩니다.


□ Github Package Registry

사실 일련의 과정은 전부 소스 제작자인 `storycraft` 님께서 전부 해두고, 패키지를 인스톨하면 바로 사용할 수 있도록 해주셨습니다. 단지 그것이 기존에 있던 `npm` 서버에 있지 않을 뿐이죠.

node-kakao 3.0.3 Latest version

 Install from the command line: [Learn more](#)

```
$ npm install @storycraft/node-kakao@3.0.3
```

 Install via package.json:

```
"@storycraft/node-kakao": "3.0.3"
```

이미 깃헙에 이런식으로 잘 설치하라고 되어있습니다.

하지만 단순히 사진만 믿고 `npm install @storycraft/node-kakao` 를 입력했다간, 뒤통수를 맞으며 배신감을 느끼게 될 것입니다.

"패키지를 찾을 수 없습니다."



아퍼라고 시발

기본적으로 npm 은 본인들의 서버인 npmjs 에서 패키지를 검색하게 되어있습니다. 하지만 노드 카카오 모듈이 존재하는 장소는 깃헙 서버입니다. 일단, npm 이 github 에 접근할 수 있게 본인의 Token Key 를 설정해야 합니다.

다음 명령어를 입력하여 명령줄에서 로그인합니다.



이것으로, Token 설정이 끝났습니다. 참 쉽죠?
하지만 이것으로 만사 해결된 것은 아닙니다.

먼저, storycraft 의 패키지는 npm 이 아니라 github 에서 다운받겠다는 설정을 해주어야 합니다.
새로운 프로젝트 폴더 (npm init 한 상태) 에서 package.json 파일이 있는 경로에, .npmrc 파일을 하나 만들어줍니다.

앞에 '!' 가 들어가야 합니다.

그리고 한 줄만 추가해 줍니다.

@storycraft:registry=<https://npm.pkg.github.com>

이러면 설정이 끝났습니다. 이제 설치가 정상적으로 되는 것을 확인할 수 있습니다.

□ Javascript 그리고 Typescript

node-kakao 는 Javascript 가 아닌 Typescript 를 사용합니다.

자료형의 관리가 개판인 자바스크립트에서, 타입스크립트는 말 그대로 자료형 (Type) 의 관리가 용이하도록 만든 언어입니다.



뉴스 속기침도 보면 수직과 수평채널에 개채가 되어있는 거지?

번두 소기와는 준병 님사도 시악했는데 왜 색제가 되어있는 거지?

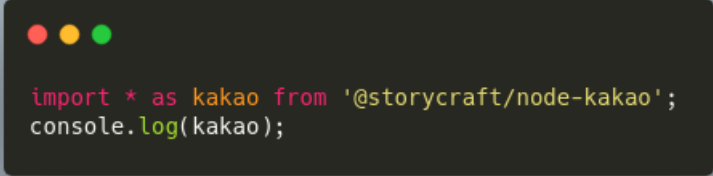
자바스크립트를 사용해도 되지만, 이 글에선 이왕 타입스크립트를 사용하도록 하겠습니다.

타입스크립트는 트랜스파일을 거쳐 결국엔 자바스크립트가 됩니다.

새로운 언어를 배우는 것이 아니라, 자료형을 관리하기 쉽도록 해주는 거라고 편히 생각하세요.

타입스크립트 모듈을 설치해줍니다. 전역으로 설치할 수 있겠지만, 우리는 간단하게 하기 위해서 "**npm install typescript @types/node**" 로, 프로젝트 로컬에다가만 설치할 것입니다.

index.ts 에서 설치한 node-kakao 모듈을 불러보겠습니다.



```
import * as kakao from '@storycraft/node-kakao';  
console.log(kakao);
```

이제 타입스크립트를 자바스크립트로 트랜스파일 할 차례입니다. 우리는 아까 typescript 모듈을 설치하면서, 트랜스파일 하는 스크립트를 설치하였습니다.



```
./node_modules/.bin/tsc index.ts
```

위 명령을 실행시키면 index.js 가 생성됩니다. "**node index.js**" 명령으로 실행시키면, 모듈이 정상적으로 불러와 진 것을 확인할 수 있습니다.



물론, 이것은 하나의 동작을 이해하는 과정입니다. 실제로 이런 방식을 사용하여 프로젝트를 진행한다면 소스 구조는 물론이고 개발하는 본인마저 혼란의 상태에 빠지게 됩니다.

코드는 더럽더라도, 정리정도는 잘 하자고요.
하지만, 이미 정돈된 상태에서 시작한다면 좀 더 구성을 쉽게 만들 수 있지 않을까요?

그런 생각으로 만들어진 이 **템플릿**을 사용하면 엄청나게 간단한 프로젝트 환경을 구성할 수 있습니다.
템플릿의 설치는 간단합니다.

```
npx degit tree-some/typescript-template ts-project
cd ts-project
npm install
npm run dev
```

npx 는 npm 버전 5.2.0 에서부터 추가된 도구입니다. 기본적으로 사용할 수 있습니다.
위 명령어들을 실행해서 "**Hello World**" 가 나온다면 성공적으로 설치된 것입니다.

.npmrc 를 추가하고, node-kakao 를 설치한다면 이제 src 폴더에서 프로젝트를 수정하실 수 있습니다.
nodemon 을 사용하기 때문에 만약 src 폴더를 수정하여 "**저장**" 을 한다면 자동으로 새로운 코드를 실행시킵니다.

글은 길게 썼지만, 마지막 내용만 잘 보고 따라한다면 프로젝트를 만드는데 3분도 걸리지 않습니다.
이 간단한 것을 아침부터 저녁까지 입만 벌리고 기다리는 모습을 보면, 부아가 다 치밀어 오릅니다.



다 읽으셨으면 이제 나가십쇼.



윤님의 게시물 더보기 >

❤️ 좋아요 27 💬 댓글 31

🔗 공유 | 신고

댓글 등록순 최신순 C

댓글알림 ☐



윤 작성자

약속 하나만 합시다. "본인은 해당 모듈이 무엇을 하는지 숙지하고 있으며, 모듈을 사용하여 발생한 모든 불이익에 대해선 본인이 책임집니다."

2020.07.10. 14:56 답글쓰기



HELL WORLD

사실 깃허브에 클론한 게 있었던건 안비밀

2020.07.10. 15:48 답글쓰기



윤 작성자

기존 클론한 것들은 리비전 144가 최대입니다. 지금 3.0.3 스테이블 리비전은 308입니다

2020.07.10. 15:53 답글쓰기



HELL WORLD

윤 알겠습니다.
감사합니다

2020.07.10. 16:06 답글쓰기



NenkaLab

이걸 봐도 똑같은짓 하면.. ㅋㅋㄹㅁㅁㅁ



2020.07.10. 15:53 답글쓰기



윤 작성자

화병나서 카페 탈퇴할듯

2020.07.10. 15:54 답글쓰기



지민

아는 사람은 많은데 쓰는 사람이 없음

2020.07.10. 16:13 답글쓰기



윤 작성자

쓰는 사람이 없다가보다 못 쓰는 사람이 많죠

2020.07.10. 16:14 답글쓰기



지민

윤 ㄹㅇㅋㅋ 기본지식이 부족하니

2020.07.10. 16:19 답글쓰기



진영
구나ㅏㅏ악

2020.07.10. 16:28 답글 쓰기



진애용
ㄹㅇ 좋은글이다....

2020.07.10. 17:04 답글 쓰기



윤 작성자
좋게 봐주셔서 감사합니다.

2020.07.10. 17:32 답글 쓰기



더이상 후회하지 말자

와 진짜 제가 하고싶던말이 다 있어 와... 프로젝트 구조도 npm도 npm run start도 모르는 인간들이 Rhino JS 무슨 자응답 조금 할줄안다고 LOCO 어떻게 해요 하는거 너무 답답했는데 진짜 너무너무감사합니다

아 참 타입스크립트는 컴파일러가 잇다고 하더라구용

2020.07.10. 17:21 답글 쓰기



윤 작성자
공식 주소에서도 Typescript Compiler, tsc 도 TypeScriptCompiler 의 줄인말이지만, 결국엔 비슷한 수준의 추상화가 되는 것이라 정확한 트랜스파일링이 맞다 생각되어 작성하였습니다.

물론, 트랜스파일링을 컴파일링이라 칭할 순 있습니다.

2020.07.10. 17:31 답글 쓰기



더이상 후회하지 말자
윤 그쵸

2020.07.10. 17:36 답글 쓰기



핀밀크
죄송합니다 ㅠㅠ

2020.07.10. 18:20 답글 쓰기



HELL WORLD

npm login --registry=https://npm.pkg.github.com를 입력 후 username과 password,email을 뭐라고 입력해야 하나요?

2020.07.10. 17:27 답글 쓰기



윤 작성자
본인 깃헙 계정으로 로그인하시면 됩니다.

2020.07.10. 17:30 답글 쓰기



Hibot

네 나가겠습니다(?)

2020.07.10. 18:30 답글 쓰기



윤 작성자



2020.07.10. 18:35 답글 쓰기



storycraft

좋아요 누르고 갑니다

2020.07.10. 20:01 답글 쓰기



윤 작성자
오예



2020.07.10. 20:05 답글쓰기



사로로

좋은 글이네요 읊음

2020.07.10. 20:23 답글쓰기



AlphaDo

형님 새싹 떼고싶는데...

2020.07.10. 23:19 답글쓰기



윤 작성자

왜조 전 이거 빼면 남는 게 없는 서민인데

2020.07.10. 23:37 답글쓰기



AlphaDo

윤 새싹에 물이 고였습니다

2020.07.11. 00:29 답글쓰기



Hibot

index.js를 직접 만들어서 쓰면 tsc쳐가며 빌드해서 만들필요 없다는건 안 비밀
그래도 쓰는게 좋다는것
node_modules/.bin/tsc ./index.ts;node ./index.js
한-줄코드

2020.07.11. 07:19 답글쓰기



Hibot

+ 사이다 ㄹㅇ 시원하네요

2020.07.11. 07:19 답글쓰기



윤 작성자

맞습니다. 하지만 저는 동작 과정을 이해하기 위하여 나눠 친 것이었고, 템플릿은 변경점이 있으면 저 명령어조차 자
동으로 입력해 주는 것입니다

2020.07.11. 10:42 답글쓰기



극성애플이

정말 좋은 글...

2020.07.11. 20:26 답글쓰기



윤 작성자

감사합니다

2020.07.11. 20:37 답글쓰기

Hibot

댓글을 남겨보세요



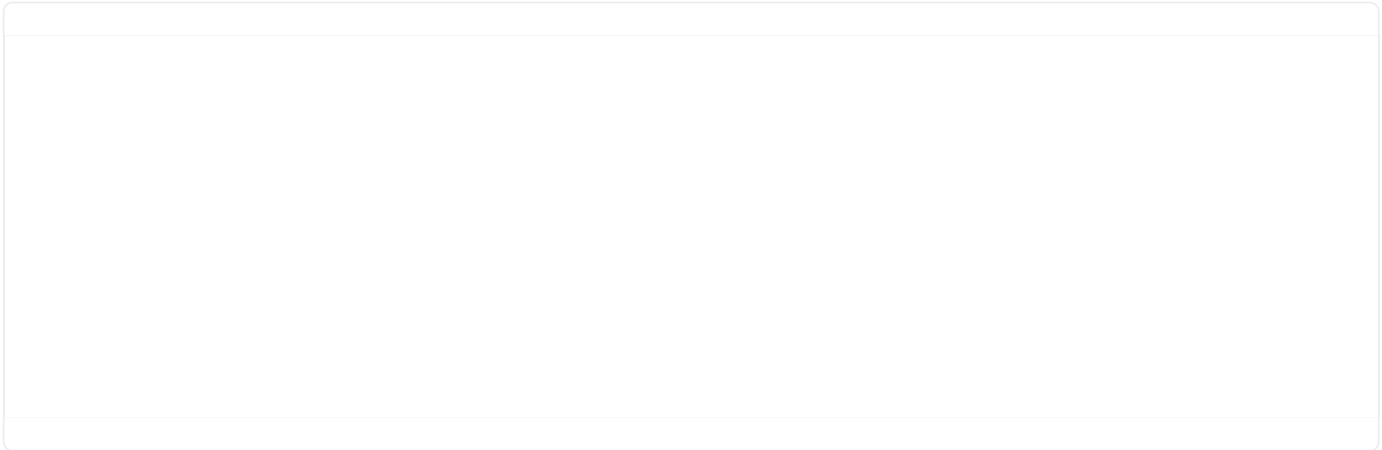
등록

글쓰기

답글

목록

▲ TOP



'자유 게시판' 게시판 글

이 게시판 새글 구독하기 ☐

카링에대해 [12]	젤리파덕	2020.07.10.
누가 또 카링 이상한거 만들었네요 🙄 [7]	춤돌이치킨	2020.07.10.
[스압] 사람들은 node-kakao를 사용하고 싶어하면서 왜 못 하는가 🙄 [31]	윤	2020.07.10.
솔직히 네이버도 🙄 [3]	더이상 후회하지 말자	2020.07.10.
소스를 올릴 때 [15]	Kiri	2020.07.10.

이 카페 인기글



학교 급식 정보 Remake

5만 2005
카카오링크 관련 글 금지

AlphaDo

♡ 2 💬 24

로코가 뭐길래 사람들이 환장하는걸까?

쿨칠

♡ 1 💬 18

기본 개념에 대한 강좌 | 조건문과 반복문

OtakoidTony

♡ 0 💬 6

안녕하십니까.

JSR

♡ 0 💬 8

원주율 100자리 가져오기

T봇

♡ 0 💬 6

카카오링크 소스 공유 및 다운로드금지

지민

♡ 3 💬 8

이발

키알봇

♡ 0 💬 5

꺼무위키 하다가 발견했는데, 나무뉴스라는 게 있더라고요...?

OtakoidTony

♡ 2 💬 2