0. 프로젝트 설치 전, VUE 환경 개발 설정이 필요, 다음을 체크

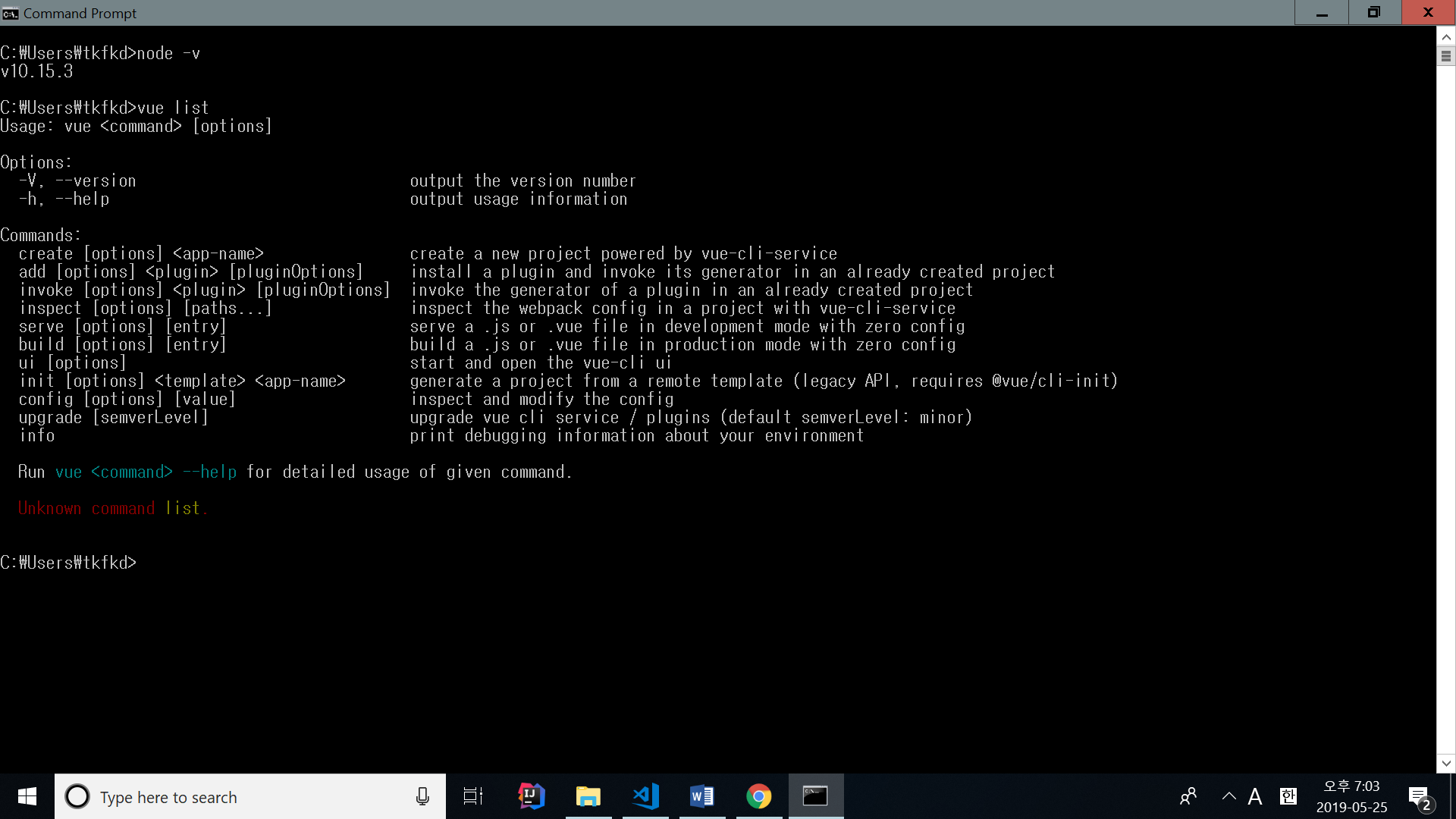
🞏 Node 설치 되어있는가? (인터넷 검색, 다운 그리고 node –v 을 cmd에 입력하여 확인)

🞏 Vue-cli 설치되어 있는가? (cmd에서 npm install -g @vue/cli)

1. Frontend 뷰 웹 팩 설치

- 템플릿

vue-cli 에서는 미리 세팅된 몇가지 템플릿을 제공한다. 제공되는 템플릿은 vuejs-templates 에 각각의 레퍼런스로 저장되어 있다. vue list 명령어를 통해서 제공되는 템플릿 리스트를 확인가능



- Webpack

Webpack은 웹 앱의 자원들을 자바스크립트 모듈로 변환하여, 하나로 묶어 웹 성능을 향상시켜 주고, 빌드 자동화 기능까지 포함한 강력한 자바스크립트 모듈 번들러

출처: <https://songta.tistory.com/54> [./Song]

- Vue init webpack projectName 과 vue create project 차이점

|  |
| --- |
| 버전 3.x  vue create helloworld |
| 버전 2.9  vue init webpack-simple 파일 경로 |

|  |  |  |
| --- | --- | --- |
|  | Vue CLI 3.0 | Vue CLI 2.9 |
| 언어 | ES6 기준 | ES5 기준 (ES6 선택 가능) |
| 웹팩 | 웹팩 설정 파일이 숨겨져 있음  (필요 설정 추가하는 형식) | 웹팩 설정 파일에서 바로 수정 가능 |
| GUI | GUI 툴 제공 (vue ui) | X |
| 구성 | 뷰 플러그인 형식 | 프로젝트 템플릿을 깃헙에서 다운로드 |
| 명령어 | vue create '프로젝트 이름' | vue init 템플릿 App 프로젝트 폴더 경로' |

|  |
| --- |
| cd backend  PS C:\Users\tkfkd\immsystem\backend> cd ..  PS C:\Users\tkfkd\immsystem> vue init webpack frontend  vue-cli · Generated "frontend".  # Installing project dependencies ...  # ========================  npm WARN deprecated browserslist@2.11.3: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools.  npm WARN deprecated bfj-node4@5.3.1: Switch to the `bfj` package for fixes and new features!  npm WARN deprecated flatten@1.0.2: I wrote this module a very long time ago; you should use something else.  npm WARN deprecated browserslist@1.7.7: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools.  npm WARN deprecated circular-json@0.3.3: CircularJSON is in maintenance only, flatted is its successor.  > core-js@2.6.8 postinstall C:\Users\tkfkd\immsystem\frontend\node\_modules\core-js  > node -e "try { require('./scripts/postinstall'); } catch (e) { /\* empty \*/ }"  Thank you for using core-js ( https://github.com/zloirock/core-js )!  Please consider supporting of core-js on Open Collective or Patreon:  > https://opencollective.com/core-js  > https://www.patreon.com/zloirock  Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)  > uglifyjs-webpack-plugin@0.4.6 postinstall C:\Users\tkfkd\immsystem\frontend\node\_modules\webpack\node\_modules\uglifyjs-webpack-plugin  > node lib/post\_install.js  npm notice created a lockfile as package-lock.json. You should commit this file.  npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node\_modules\fsevents):  npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})  added 1203 packages from 675 contributors and audited 11959 packages in 50.895s  found 12 vulnerabilities (7 moderate, 5 high)  run `npm audit fix` to fix them, or `npm audit` for details  Running eslint --fix to comply with chosen preset rules...  # ========================  > frontend@1.0.0 lint C:\Users\tkfkd\immsystem\frontend  > eslint --ext .js,.vue src "--fix"  # Project initialization finished!  # ========================  To get started:  cd frontend  npm run dev  Documentation can be found at https://vuejs-templates.github.io/webpack  PS C:\Users\tkfkd\immsystem> cd frontend  PS C:\Users\tkfkd\immsystem\frontend> npm run dev  > frontend@1.0.0 dev C:\Users\tkfkd\immsystem\frontend  > webpack-dev-server --inline --progress --config build/webpack.dev.conf.js  12% building modules 23/29 modules 6 active ...\tkfkd\immsystem\frontend\src\App.vue{ parser: "babylon" } is deprecated; we now treat it as { parser: "babel" }.  95% emitting  DONE Compiled successfully in 7923ms 18:48:25  I Your application is running here: http://localhost:8081  Terminate batch job (Y/N)? y  PS C:\Users\tkfkd\immsystem\frontend> |

**- proxyTable 역할**

일반적으로 프록시란 다른 클래스의 인터페이스로 동작하는 클래스

Frontend 개발 중 <http://localhost:8080/api> 요청시에 <http://localhostL3000/api> 로 프록시 함

프론트엔드에서 백엔드로 http request가 보내기 위해 필요

Webpack 사용시 proxyTable 필요, 보통 proxy란 이름으로 다음 예시 코드와 같이 설정

|  |
| --- |
| devServer: {  host: '127.0.0.1',  port: 8010,  proxy: {  '/counsel/apply/api': { // /counsel/apply/api 는 예시다  target: '[https://개발서버\_URL/counsel/apply/api](https://xn--_url-9k3p618e2ua570a/counsel/apply/api)',  changeOrigin: true,  pathRewrite: {  '^/counsel/apply/api': ''  }  }  }  } |

<https://lovemewithoutall.github.io/it/webpack-config-for-debug/>

**- http-proxy-middleware 기술**

<https://github.com/chimurai/http-proxy-middleware>

<https://expressjs.com/ko/guide/writing-middleware.html>

**- vue-cli3 proxy 설정 (내가 못 찾았던 부분)**

vue-clli가 3으로 버전업이 되면서 프록시 설정 부분이 살짝 달라짐. 우선 설정 파일이 없어서 생성. root 폴더에 vue.config.js 라는 파일을 만들고, 다음의 코드를 작성

|  |
| --- |
| 'use strict'  module.exports = { devServer: { proxy: { // proxyTable 설정 '/api': { target: '[http://localhost:8080](http://localhost:8080/)', changeOrigin: **true**  }  } } } |

출처: https://cpdev.tistory.com/114 [하루하나]

**- axios 의미**

**Axios는 HTTP통신을 하는데 매우 인기있는 Javascript 라이브러리**

**Axios는 브라우저와 Node.js 플랫폼에서 모두 사용**

HTTP 클라이언트 라이브러리로써, 비동기 방식으로 HTTP 데이터 요청을 실행

내부적으로 AXIOS는 직접적으로 XMLHttpRequest 를 다루지 않고 “AJAX 호출”

<https://velog.io/@rohkorea86/%EB%B9%84%EB%8F%99%EA%B8%B0-%EB%B9%84%EB%8F%99%EA%B8%B0%EC%97%90-%EB%8C%80%ED%95%B4%EC%84%9C-%EC%B0%A8%EA%B7%BC%EC%B0%A8%EA%B7%BC-%EB%8B%A4%EB%A3%A8%EB%A0%A4%EA%B3%A0-%ED%95%A9%EB%8B%88%EB%8B%A4.>

- axios api vs fetch api

- Error solution

|  |
| --- |
| [vue/require-v-for-key]  Elements in iteration expect to have 'v-bind:key' directives.eslint-plugin-vue |

By adding => :key="user.id"

|  |
| --- |
| <div v-for="user in users" :key="user.id" class="user"> |

-eslint 에러

Eslint는 코드 표준화를 위한 도구.

일단 useEslint: true -> false로 바꿔놓고 개발 중.

-계속 똑 같은 페이지… 왜???

- vuetify 추가

npm에서 vue add vuetify, 그리고 main.js에 추가

|  |
| --- |
| import vuetify from 'vuetify' // new 추가  Vue.config.productionTip = false  Vue.prototype.$http = axios  Vue.use(vuetify) // new 추가 |

-현재 에러: backend 페이지 계속 똑 같은 페이지를 리턴(이미 프론트엔드에서 삭제한 페이지임에도 불구하고)

(a) app.use(require('connect-history-api-fallback')()); //5.26 vue router 연동 위한 모듈 추가

해당 코드를 코멘트 처리하면 프론트엔드 연동 없이 백엔드 기본 페이지에서 데이터를 불러옴.

(b) 문제점은 npm run build를 통해 migration같은 걸 하면 해결!

-'connect-history-api-fallback' 모듈이란(front end의 히스토리 모드를 위한 backend history 모드 관련 모듈 설정)

Vue router와 express의 연동을 위해, express(nodejs)에 connect-history-api-fallback이라는 미들웨어 필요함

<좀 더 깊게 알아보기>

기본적으로 vue-router를 사용하게 되면 hash mode와 history mode 중 디폴트로 hash mode가 적용된다.

여기서 hash mode란 URL hash를 사용하여 전체 URL을 시뮬레이트 하는데, URL이 변경될 때 페이지가 다시 로드되지 않는다. 즉 hash 기호 다음의 경로는 페이지 내보의 이름으로 여겨지기 때문에, hash 이후의 경로가 바뀌더라도 페이지가 다시 로드되지 않는다.

Vue-router 사용 중 hash를 제거하기 위해서는 라우터의 history 모드를 설정할 수 있다. 그래서 다시 로드하지 않고 URL을 탐색할 수 있다.

History 모드를 사용할 때 주의사항: 서버는 404 에러를 보고하지 않는 문제가 발생할 수 있음. 그 이유는 발견되지 않는 경로가 이제 inde.html파일에서 제공하기 때문이다. 이를 위해서 Vue 앱에서 catch-all router를 구현하여 404 페이지를 표시하여야 한다. 다음 주소 값이 그 차이이다.

<http://localhost:3000/#/users/1>

<http://localhost:3000/api/users>