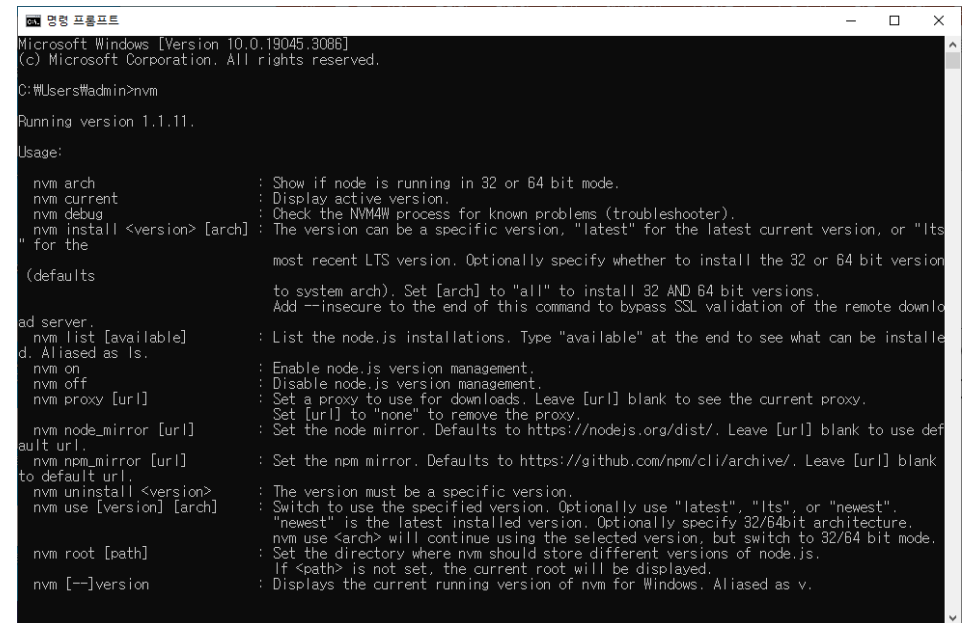


node.js

2023-06-22

1. nvm 설치

- <https://github.com/coreybutler/nvm-windows/releases>
- nvm-setup.exe 다운받아서 설치하면 됨
- nvm : node.js 형상관리 프로그램
- 명령프롬프트 열어서
- nvm 치고 엔터치면
- 버전하고 정보 나오면 설치완료
- 명령어 목록 중 install uninstall list use
- 이정도만 알면 됨



```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>nvm

Running version 1.1.11.

Usage:

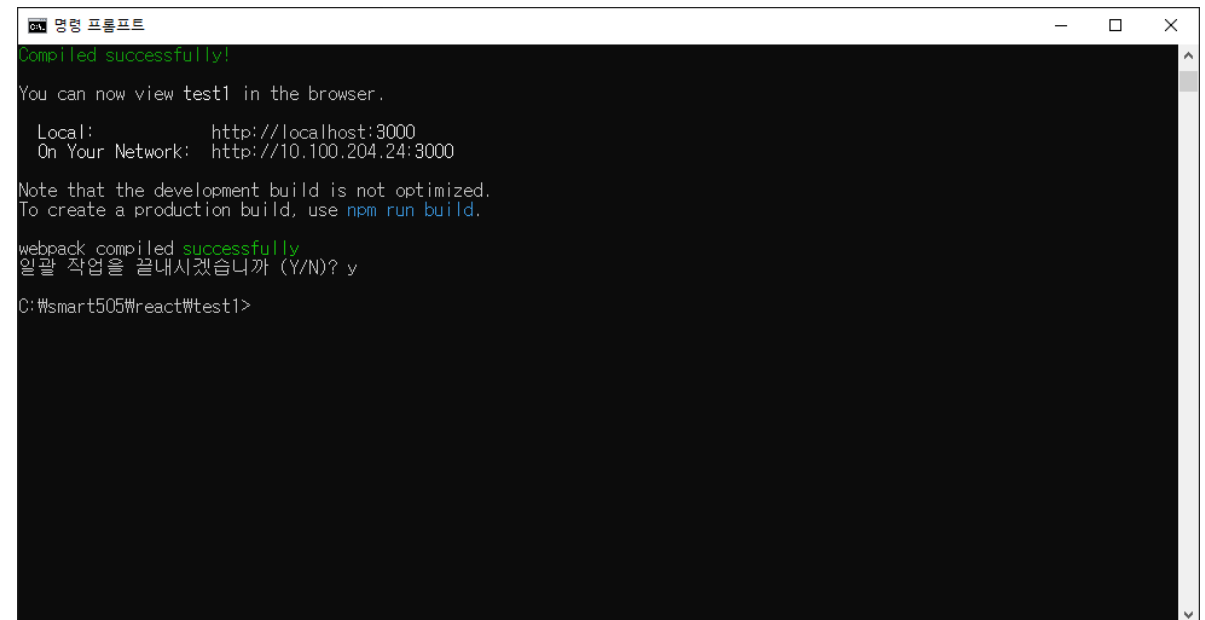
nvm arch           : Show if node is running in 32 or 64 bit mode.
nvm current        : Display active version.
nvm debug          : Check the NVM4W process for known problems (troubleshooter).
nvm install <version> [arch] : The version can be a specific version, "latest" for the latest current version, or "lts"
                             for the
                             most recent LTS version. Optionally specify whether to install the 32 or 64 bit version
                             to system arch). Set [arch] to "all" to install 32 AND 64 bit versions.
                             Add --insecure to the end of this command to bypass SSL validation of the remote download
                             server.
nvm list [available] : List the node.js installations. Type "available" at the end to see what can be installed.
nvm on             : Enable node.js version management.
nvm off            : Disable node.js version management.
nvm proxy [url]    : Set a proxy to use for downloads. Leave [url] blank to see the current proxy.
nvm node_mirror [url] : Set the node mirror. Defaults to https://nodejs.org/dist/. Leave [url] blank to use default url.
nvm npm_mirror [url] : Set the npm mirror. Defaults to https://github.com/npm/cli/archive/. Leave [url] blank to default url.
nvm uninstall <version> : The version must be a specific version.
nvm use [version] [arch] : Switch to use the specified version. Optionally use "latest", "lts", or "newest".
                             "newest" is the latest installed version. Optionally specify 32/64bit architecture.
                             nvm use <arch> will continue using the selected version, but switch to 32/64 bit mode.
nvm root [path]    : Set the directory where nvm should store different versions of node.js.
                             If <path> is not set, the current root will be displayed.
nvm [--]version    : Displays the current running version of nvm for Windows. Aliased as v.
```

2. nvm 사용

- `nvm install 20.3.1`
- 치고 엔터 치면 다운로드 어찌구 함
- 완료시 `nvm use 20.3.1`
- 그리고 `nvm list` 치면 전체목록이랑 사용중인 버전 나옴
- `nvm install lts` 치고 엔터 (long term support 안정적인 버전 다운) - 이거 사용
- `node --version` 사용중인 node 버전: node.js 실행기
- `npm --version` 사용중인 npm 버전: node 패키지 매니저 - js는 클래스가아니라 모듈 형식인데, 다른사람이 만든 js를 사용하려면 모듈을 받아야함. 이 모듈관리를 해주는 프로그램

3. 리액트 - visual studio

- 리액트 프로젝트 만들기
- cmd 열어서 cd 경로 쳐서 프로젝트폴더를 사용하도록 설정
- 아래 사이트 참고
- <https://create-react-app.dev/docs/getting-started>
- npm --version : 버전정보 나옴
- npx create-react-app test1 : test1 프로젝트 생성
- cd test1 : test1 폴더 사용
- npm start : 프로젝트 시작
- ctrl + C : 종료



```
C:\> 명령 프롬프트
Compiled successfully!
You can now view test1 in the browser.
  Local:            http://localhost:3000
  On Your Network:  http://10.100.204.24:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
일괄 작업을 끝내시겠습니까 (Y/N)? y
C:\smart505\react\test1>
```

4. 리액트 - 인텔리제이

- 강 프로젝트 만들면 인텔리제이가 3번방식 그대로 써서
- 프로젝트 만들어줌
- jquery쓸려면 터미널에 윈도우 터미널 열어서
- `npm install jquery`
- 실행하면 package.json에 dependencies에 제이쿼리 추가됨
- 그리고 src내의 js파일에 `import $ from 'jquery';`
- cmd에서 만든 프로젝트나 깃허브에서 다운받은 프로젝트의 내용물이 인텔리제이에서 인식을 못하는 경우가 있음 -> 인텔리제이에서 만든 프로젝트에만 있는 파일들을 복사해서 옮겨주면됨
- (실행파일만들기): 구성편집->구성추가->npm ->이름설정, 명령어 run 그대로(서비스용으로 출력할 때는 build(단일파일로 만들어줌)), 스크립트 start, 노드인터프리터 버전확인
- 구성추가-> JavaScript 디버그 ->이름설정, url localhost:3000(리액트 기본주소)

4. 리액트 - 인텔리제이

- 배포 : 서버, 프론트엔드 동작시키려면?
- 프론트엔드(리액트) : 구성편집 -> npm start 말고 npm build 해서 추가후 실행
- 백엔드(스프링 gradle) : 우측 Gradle 탭 클릭수 Tasks > build > bootJar 더블클릭하면됨

--- build.gradle에서 `version = '0.0.1-SNAPSHOT'` 수정하면 버전바뀌는거임

실행 : window + R -> cmd -> cd C:\smart505\react\backend\build\libs

dir후 확인

C:\smart505\react\backend\build\libs 디렉터리

2023-07-25 오전 09:45 <DIR> .

2023-07-25 오전 09:45 <DIR> ..

2023-07-25 오전 09:45 18,975,272 backend-0.0.1-SNAPSHOT.jar

1개 파일 18,975,272 바이트

2개 디렉터리 110,815,432,704 바이트 남음

C:\smart505\react\backend\build\libs>java --version

java 17.0.6 2023-01-17 LTS

Java(TM) SE Runtime Environment (build 17.0.6+9-LTS-190)

Java HotSpot(TM) 64-Bit Server VM (build 17.0.6+9-LTS-190, mixed mode, sharing)

C:\smart505\react\backend\build\libs>java -jar backend-0.0.1-SNAPSHOT.jar

엔터치면 실행됨

4. 리액트 - 인텔리제이

- 백엔드 폴더에 src > main 폴더에 프론트엔드 build 폴더안 내용 붙여넣기
- build.gradle에 오른쪽내용 추가

```
// 리액트 프로젝트 포함하기
// 변수 선언 및 리액트 프로젝트 폴더 설정, $projectDir == 스프링 프로젝트
def frontendDir = "$projectDir/src/main/frontend"

// 리액트 프로젝트가 복사될 위치 설정
sourceSets {
    main {
        resources {
            srcDirs = ["$projectDir/src/main/resources"]
        }
    }
}

// processResources 태스크는 스프링 부트 프로젝트에서 기본적으로 제공되는 태스크로서
// src/main/resources에 있는 리소스 파일들을 빌드 결과물에 포함시키기 위해 사용
processResources {
    dependsOn "copyReactBuildFiles"
}

// 리액트 프로젝트 인스톨
task installReact(type: Exec) {
    workingDir "$frontendDir"
    inputs.dir "$frontendDir"
    group = BasePlugin.BUILD_GROUP

    // 리액트 모듈 다운로드
    if(System.getProperty("os.name").toLowerCase(Locale.ROOT).contains("windows")) {
        commandLine "npm.cmd", "audit", "fix"
        commandLine "npm.cmd", "install"
    }
    else {
        commandLine "npm", "audit", "fix"
        commandLine "npm", "install"
    }
}
```

```
// 리액트 프로젝트 빌드
task buildReact(type: Exec) {
    // 해당 명령 실행 후 동작
    dependsOn "installReact"
    workingDir "$frontendDir"
    inputs.dir "$frontendDir"
    group = BasePlugin.BUILD_GROUP

    // 리액트 프로젝트 빌드 명령
    if(System.getProperty("os.name").toLowerCase(Locale.ROOT).contains("windows")) {
        commandLine "npm.cmd", "run-script", "build"
    }
    else {
        commandLine "npm", "run-script", "build"
    }
}

// 빌드된 리액트 프로젝트를 스프링 프로젝트로 복사
task copyReactBuildFiles(type: Copy) {
    dependsOn "buildReact"
    // build 폴더의 내용을 static 에다 집어넣어라
    from "$frontendDir/build"
    into "$projectDir/src/main/resources/static"
}

// buildJar 명령 실행 시만 아래 명령을 실행
tasks.bootJar {
    dependsOn "copyReactBuildFiles"
}
```