

## 1 설치 환경

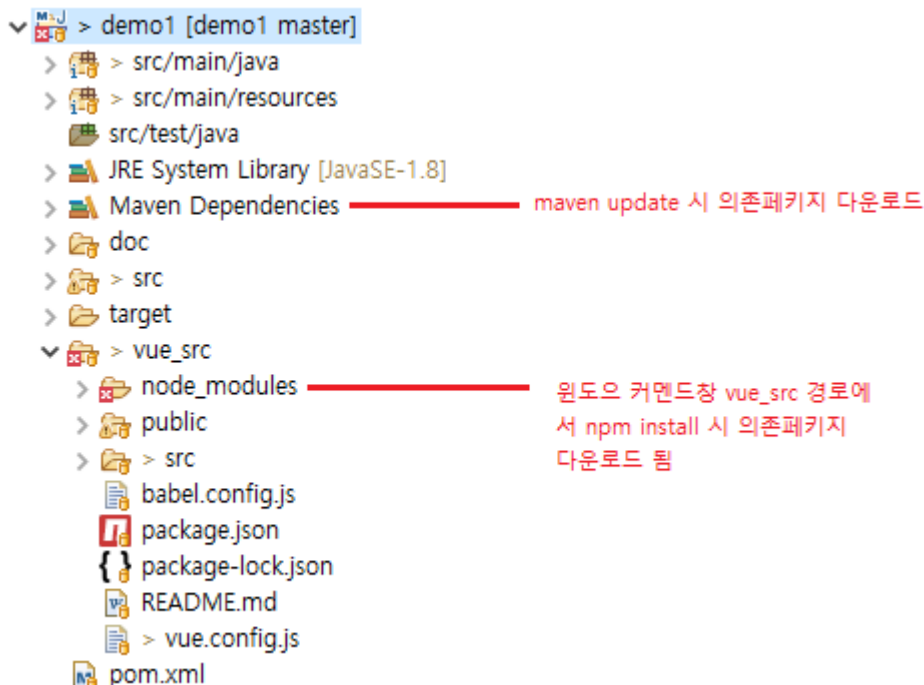
프로그램	버전	비고
Java	1.8.0_291	Oracle Java
eGovFrameDev	3.10.0-64bit	Java IDE
Node.JS	V14.17.5	
VueCLI	@vue/cli 4.5.15	
Visual Studio Code	1.63.2	VueJS IDE
GIT	2.32.0.windows.2	개발자 PC에 설치 형상관리 프로그램

## 2 설치 순서

- ① git clone 으로 프로젝트 복사
- ② eGovFrame 실행 후 복사한 프로젝트 import
- ③ eGovFrame(Eclipse) > Package Explorer 에서 마우스 오른쪽 클릭 > Maven Update Project 수행
- ④ domo1(프로젝트 루트 폴더)/vue\_src 경로를 윈도우 커멘드창으로 실행하여 npm install 수행(vue\_src 폴더 밑에 node\_modules가 생성되며 의존 패키지를 다운로드하게 된다.)

## 3 설치 구성

- ① 디렉토리 구성

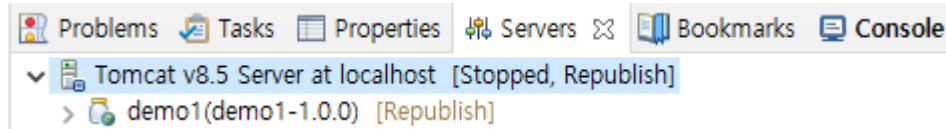


- ② Java Maven 프로젝트 생성시 부여한 특성
  - i. eGovFrame Web Project(Generate Example 소스로 시작하여 eGovFrame 기본 기술 최대한 활용)
- ③ Vue 프로젝트 생성시 부여한 특성
  - i. Vue version 2.X

- ii. Babel, Router(history mode), Vuex, CSS Pre-processors(node-sass), Linter/Formatter(ESLint + Prettier)

#### 4 실행 방법

- ① 프론트 : 윈도우 커맨드 vue\_src 폴더에서 npm run serve 실행(8888포트)
- ② 백엔드 : demo1 프로젝트를 톰캣으로 실행(8080포트)



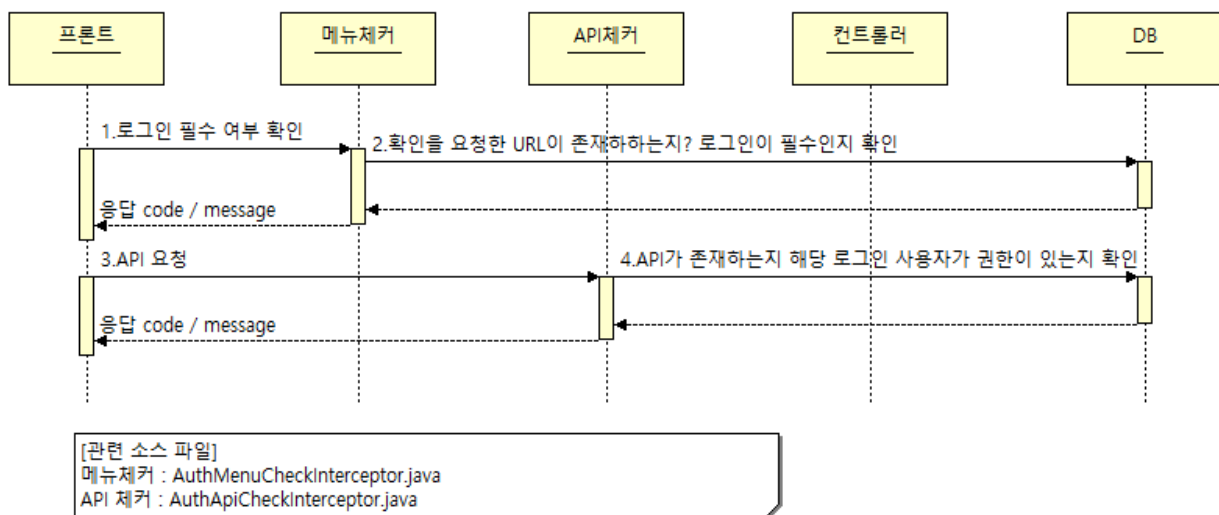
##### Web Modules

Configure the Web Modules on this server.

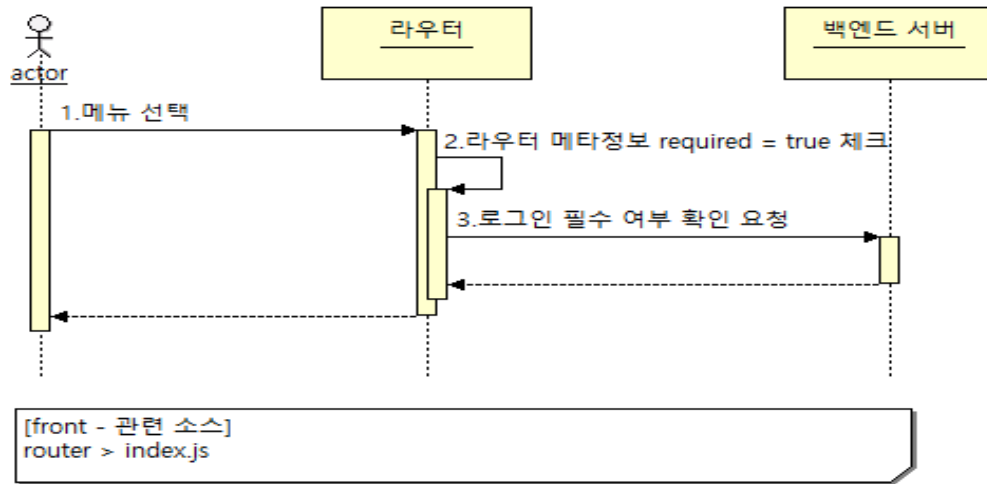
Path	Document Base	Module	Auto Reload
/	demo1	demo1	Enabled

#### 5 기반 기술 설명

- ① 접근권한 체크를 위한 구조 설명

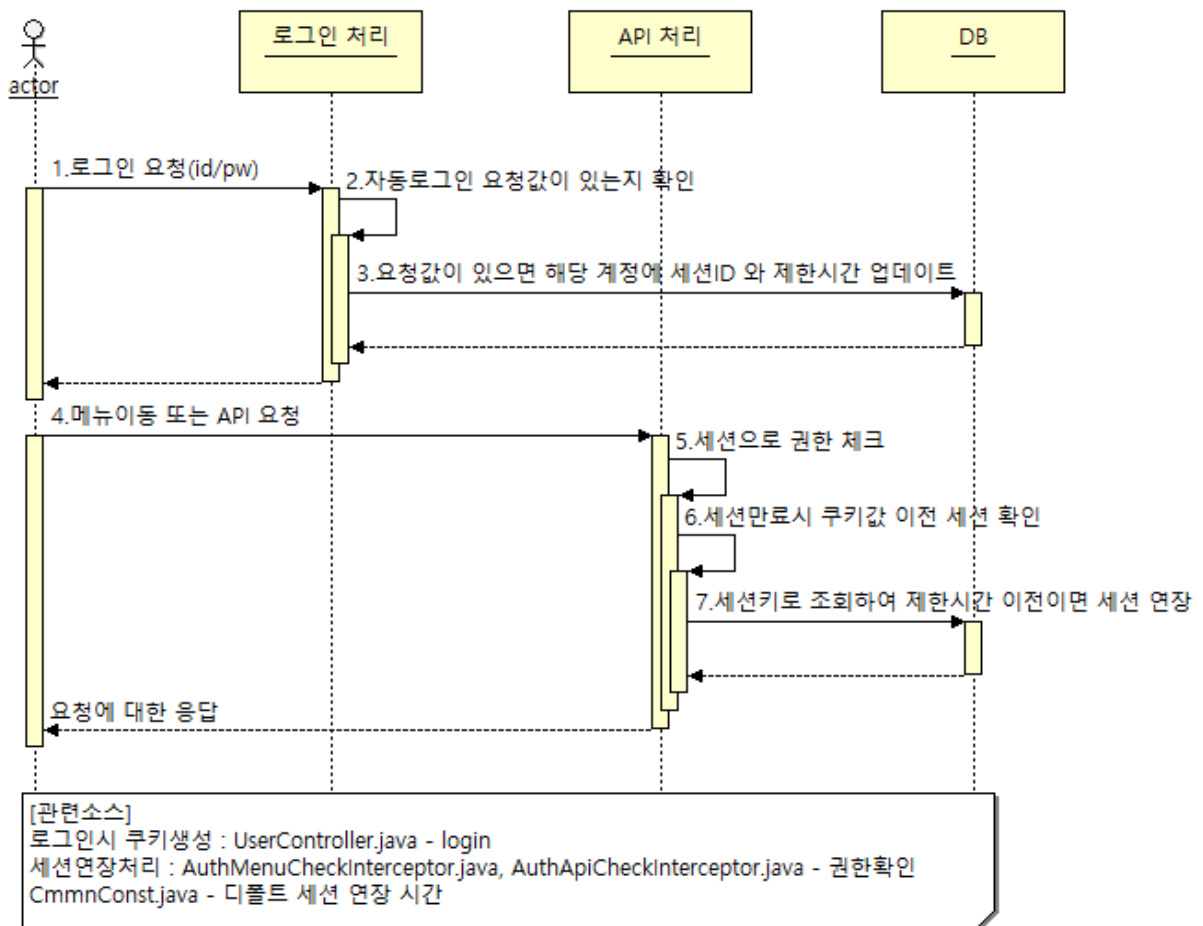


- i. 확인방법 : 데모프로젝트 실행 후 상단 메뉴에서 나의정보 클릭
- ii. 하단에 “토스트메시지로 로그인 후 이동가능합니다.” 메시지 출력
- iii. 이번에는 로그인 메뉴를 선택하고(참고로 로그인 메뉴는 누구나 접근할 수 있는 메뉴임) 아이디 및 비밀번호에 demo1 / 1234 입력 후 로그인 하면 정상 로그인 후 나의정보 페이지로 자동 이동함



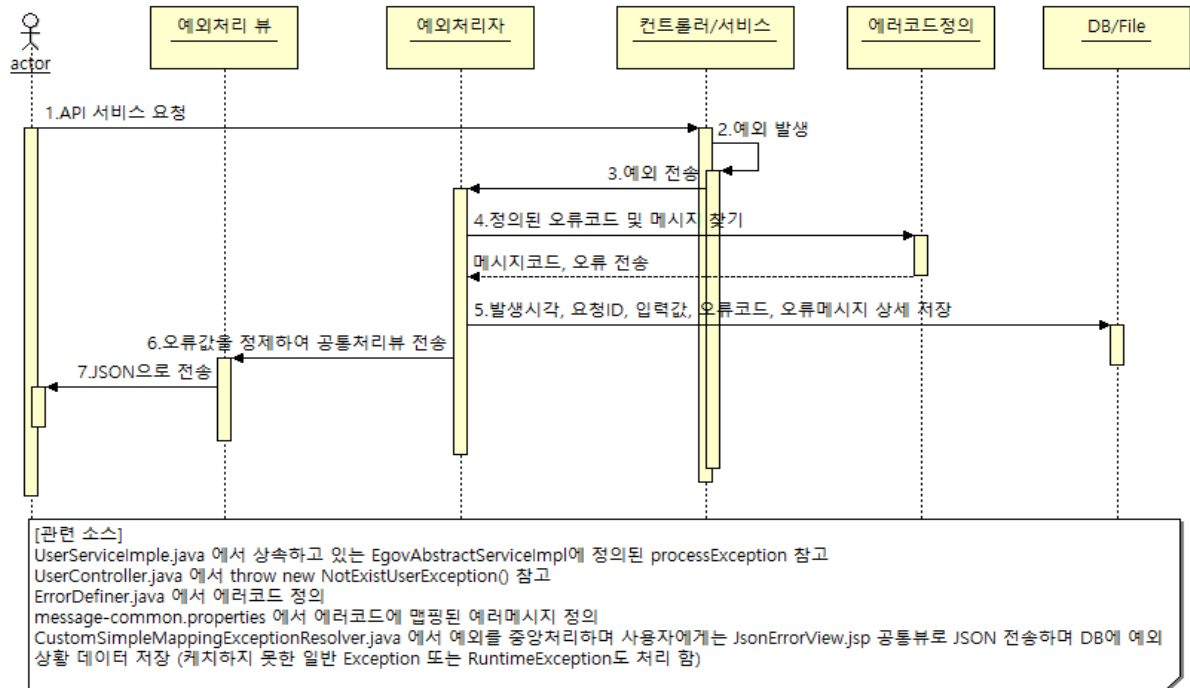
## ② 세션연장(자동 로그인)

### i. 로그인시 자동로그인 체크 후 확인

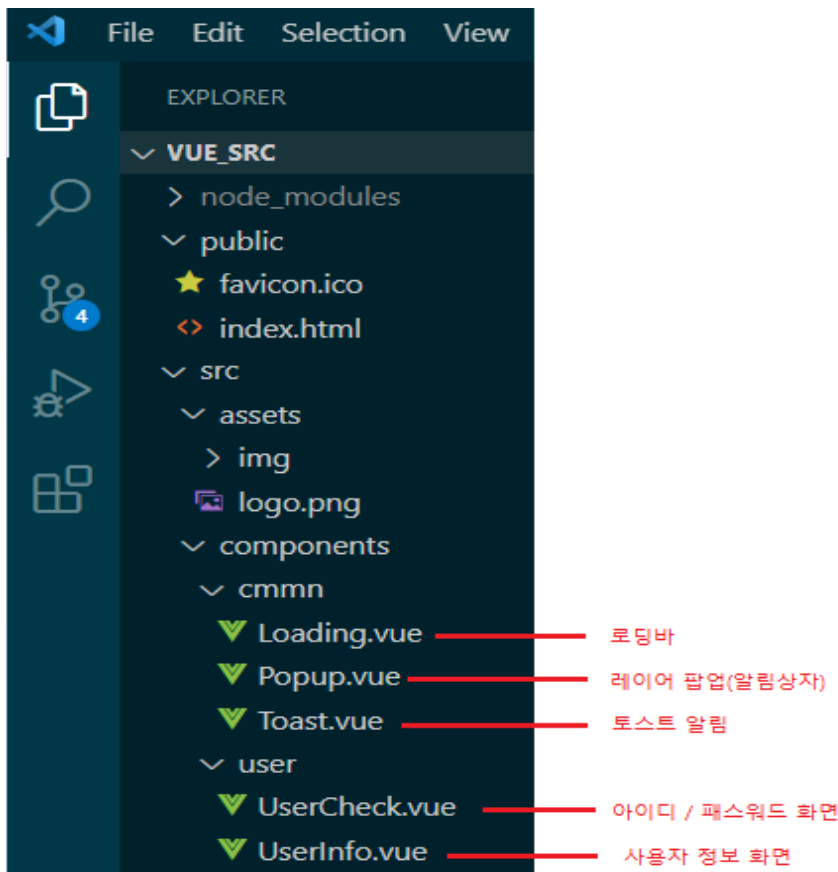


## ③ 에러, 안내처리 기능

- i. 확인방법 : 회원가입시 동일ID로 두번(중복) 등록하면 에러코드/에러메시지 전송되며 DB에 자세한 에러메시지 저장됨



#### ④ 화면에서 많이 사용하는 Component 모듈화



- 로딩바 : http 통신 전 보이고, http 통신 완료 후 가림 처리
- 레이어 팝업 : 오류메시지 등 자세하 알림
- 토스트 알림 : 간단한 메시지 표시(자동으로 보이고 잠시 후 사라짐)
- 아이디 패스워드 화면 : 사용자 / 관리자 등 재사용 목적
- 사용자 정보 화면 : 회원가입 / 나의 정보 등 재사용 목적

A. 사용하는 곳 : 선언부(App.vue - template에 공통 컴포넌트 설정)

```
<template>
  <div id="app">
    <div class="menu-container">
      <router-link class="menu" to="/">HOME</router-link>
      <router-link class="menu" to="/api/user/memberJoin">회원가입</router-link>
      <router-link class="menu" to="/api/user/login">로그인</router-link>
      <router-link class="menu" to="/api/user/myInfo">나의 정보</router-link>
    </div>
    <router-view></router-view>
    <loading ref="loading"></loading>
    <toast ref="toast"></toast>
    <popup ref="popup" :popupWidth="40"></popup>
  </div>
</template>
```

App.vue 상단에 전역으로 설정

(App.vue - Vue 최상위 인스턴스 생성시 이벤트 수신기 설정)

```
created() {
  // 부모객체에서 공통 수신함수 등록
  this.$EventBus.$on("showToast", (msg) => {
    this.showToast(msg);
  });
  this.$EventBus.$on("showLoading", () => {
    this.showLoading();
  });
  this.$EventBus.$on("showAlert", (title, msg, callback) => {
    this.showAlert(title, msg, callback);
  });
}
```

B. 사용하는 곳 : 이용부(MainView.vue)

```
<script>
export default {
  components: { },
  methods: {
    showAlert() {
      this.$EventBus.$emit("showAlert", "제목", "내용 : 하단 확인버튼을 클릭하면 파라미터  
함수 실행이 필요할 경우 적절하게 사용하시기 바랍니다.", (data) => { alert(data) })
    }
  }
};
</script>
```

## ⑤ 로그인 및 세션정보(store) 설정

### i. 관련 소스

store	
modules	
JS user.js	Store 모듈(필요시 추가하여 index.js 전체 설정에 추가)
JS index.js	Store 전체 설정

- ⑥ 유틸성 함수(팝업, 메시지 처리, 화면이동, 기타 등등)
  - i. 관련 소스
- ⑦ Component별 코딩 샘플
  - i. 관련 소스
- ⑧ 필수값 및 정합성 체크 코드
  - i. 관련 소스
- ⑨ 트랜잭션(CRUD)처리(동기, 비동기) 및 결과처리(팝업알림, 오류처리)
  - i. 관련 소스
- ⑩ 페이징 처리 샘플
  - i. 관련 소스
- ⑪ computed와 watch기능 샘플
  - i. 관련 소스