



포팅매뉴얼

사전준비

1) openJDK11 설치

2) java 환경변수 설정

- [PATH]

앞서 설치한 jdk에는 java / javac 등 우리에게 필요한 파일을 어디서나 CLI로 실행하기 위해서는 PATH라는 환경변수에 등록되어 있어야 한다.

- [JAVA_HOME]

Tomcat을 실행시킬 때 JAVA_HOME 변수에서 jdk를 확인하기 때문에 필요

gitlab 소스 클론 이후 빌드 및 배포 관련

1) 웹서버

- [NGINX]

: version 1.18.0

- [Apache Server]

2) WAS

- [Tomcat]

: Tomcat 9

3) Back-End

- [Eclipse]

: Eclipse Java 2018-09 이후 버전

- [Spring Tool Suite3]

: sts-3.9.13.RELEASE 버전

- [MySql]

: MySQL Workbench 8.0

- [JPA]

: spring data jpa

```

spring.jpa.hibernate.naming.implicit-
strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-
strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
spring.jpa.hibernate.ddl-auto=validate
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect

```

- [SPA]

```

spring.resources.static-locations=classpath:/dist/
spa.default-file=/dist/index.html
spring.mvc.throw-exception-if-no-handler-found=true
spring.resources.add-mappings=false

```

- [Jwt]

```

:
jwt.secret=dyAeHubOOc8KaOfYB6XEQoEj1QzRIVgtjNL8PYs1A1tymZvvqkcEU7L1imkKHeDa
: jwt.expiration=43200

```

- [Swagger]

```

: API 생성 및 테스트

```

- [Gradle]

```

def querydslDir = 'src/main/generated'
querydsl {
    library = "com.querydsl:querydsl-apt"
    jpa = true
    querydslSourcesDir = querydslDir
}

sourceSets {
    main {
        java {
            srcDirs = ['src/main/java', querydslDir]
        }
    }
}

compileQuerydsl {
    options.annotationProcessorPath = configurations.querydsl
}

configurations {
    querydsl.extendsFrom compileClasspath
}

```

1) QueryDSL 플러그인

```

: classpath"gradle.plugin.com.ewerk.gradle.plugins:querydsl-plugin:${querydslPluginVer}"

```

: apply plugin: 'com.ewerk.gradle.plugins.querydsl'

2) node

: download = true

: version = '14.17.0'

: nodeModulesDir = file("\${project.projectDir}/../frontend")

3) WebRTC 클라이언트 의존성 추가

: implementation('org.webjars.bower:webrtc-adapter:7.4.0')

4) Front-End

- [Visual Studio code]
- [Vue]
 - 1) "axios": "^0.21.1",
 - 2) "core-js": "^3.6.5",
 - 3) "openvidu-browser": "^2.19.0",
 - 4) "vue": "^2.6.11",
 - 5) "vue-router": "^3.2.0",
 - 6) "vuetify": "^2.4.0",
 - 7) "vuex": "^3.4.0"
- [외부라이브러리]
 - 1) jQuery : 2.1.4
 - 2) lodash : 3.10.1

5) WebRTC

- [OpenVidu]
 - : OpenVidu 2.19.0 버전
 - : 일대일, 일대다, 다대다로 화상회의가 진행되며, Apache License v2로 지정된 오픈소스이다.
- [Docker]
 - : Docker 20.10.7 버전
 - : Dockerfile.getaroom - 반드시 사설인증서를 생성한 후에 http-server 를 https 로 기동하여야, 웹브라우저에거 마이크와 카메라 권한 획득 가능

빌드 환경변수

- [gradle]
 - : GRADLE_HOME={GRADLE_HOME}\bin
- [.eslintrc.js]

배포 특이사항

- [AWS EC2]

: 웹 서비스 인터페이스를 사용해 다양한 운영 체제로 인스턴스를 시작하고, 이를 사용자 정의 애플리케이션 환경으로 로드하며, 네트워크의 액세스 권한을 관리하고, 원하는 수의 시스템을 사용해 이미지를 실행할 수 있는 진정한 가상 컴퓨팅 환경

데이터베이스

- [DataSource]

spring.datasource.url=jdbc:mysql://i5b208.p.ssafy.io:3306/ssafy_web_db

spring.datasource.username=root

spring.datasource.password=ssafy

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

- [주요계정]

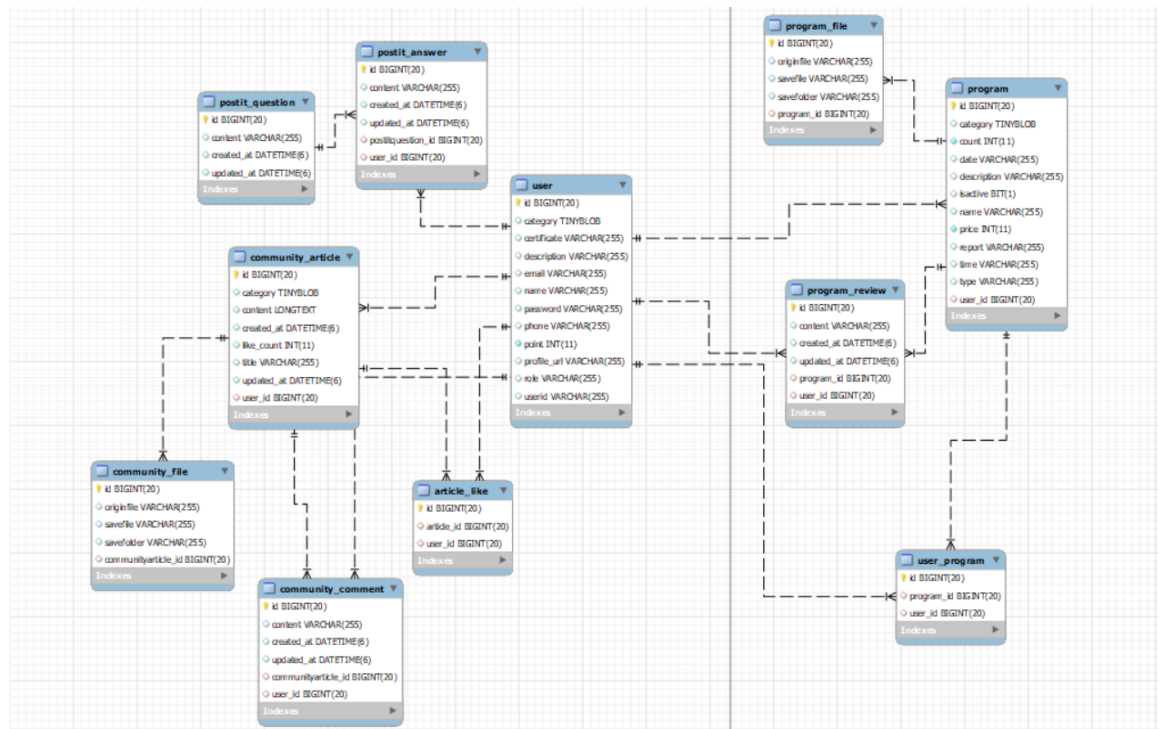
1) 관리자

ID : admin / PW : admini5b208

2) 사용자

ID : ssafy / PW : ssafyi5b208

- [ERD]



외부서비스

1) 결제 API

- I'mport(아임포트)

: 관리자 모드로 이니시스에서 제공하는 테스트계정으로 결제가 진행

: 실제로 결제승인이 이루어지지만, 매일 자정이 되기 전에 일괄 자동 취소가 이루어지는 상황

: 기본 PG사 KG이니시스, 추가 PG사 토스, 카카오페이, 다날(신용카드, 휴대폰), PAYCO 등

: jQuery 기반이기 때문에 jQuery라이브러리를 추가해줘야 한다.

```
<!-- jQuery -->
```

```
<script type="text/javascript" src="https://code.jquery.com/jquery-2.1.4.min.js" ></script>
```

```
<!-- iamport.payment.js -->
```

```
<script type="text/javascript" src="https://cdn.iamport.kr/js/iamport.payment-1.1.5.js"></script>
```