



포팅 매뉴얼

Dog-Live 포팅 매뉴얼

| SSAFY 5기 공통프로젝트 서울 5반 1팀 견과류 🥜

조다운 이상현 이영주 최혜원 한승희



목차

- 1 개발 환경 세팅
- 2 계정 정보
- 3 프로퍼티 정보
- 4 빌드 및 배포 방법
- 5 외부 서비스



개발 환경 세팅

- OS : Windows 10
- Server : AWS EC2
 - Ubuntu 20.04.1
 - Docker 20.10.8
- Backend
 - Java : Java 1.8.0

- Framework : SpringBoot 2.4.5
- ORM : JPA(Hibernate)
- Nginx : 1.18.0 (Ubuntu)
- IDE : IntelliJ 2021.1.3 / Visual Studio Code 1.59
- Dependency tool : gradle-6.7
- Database : MariaDB-10.3.31
- Image Server : AWS S3 & CloudFront
- Frontend
 - HTML5, CSS3, Javascript(Es6)
 - Vue 3.0.0
 - @vue/cli 4.5.0
 - Vuex 4.0.0
 - Vue-axios 3.2.4

계정 정보

DB 접속 계정 정보

- ID : doglive
- PW : doglivedoggi

Jenkins 접속 계정 정보

- ID : admin
- PW : doglivedoggi
- Jenkins 관리자 페이지
 - <http://i5a501.p.ssafy.io:9090> 로 접속

- 주의 : https가 아니라 **http** 입니다!

프로퍼티 파일 목록

```
backend
├── src
│   ├── main
│   │   └── resources
│   │       ├── application.yaml
│   │       └── application-aws.yaml
```

- **application.yaml** (프로젝트 셋팅, 배포 셋팅, DB 셋팅 정보)
- **application-aws.yaml** (S3 서버 accessKey, secrecKey 정보)

빌드 및 배포 방법

1. 로컬 실행 가이드

- a. Project Git Clone :

```
$ git clone https://lab.ssafy.com/s05-webmobile1-sub3/S05P13A501
```

- b. 테스트 방법

```
$ cd backend/build/libs
$ java -jar common-project-1.0-SNAPSHOT.jar
```

<https://localhost:8081> 로 들어가서 확인한다.

c. IDE로 프로젝트 Import

- IntelliJ 기준
 - File > New > Project from Existing sources > 프로젝트 경로 선택 > import 방식 gradle
- 프로젝트 실행 : Gradle Project > bootRun
- 브라우저에서 https://localhost:8081로 접속

d. 로컬 테스트 시 주의사항

- 로컬 실행 시 **로그인** 기능을 이용하려면, 카카오 로그인 **Redirect URI** 를 변경해주어야, 로그인 후 로컬 페이지로 Redirect된다.

(현재는 배포 사이트로 Redirect 되도록 셋팅되어 있음)

⇒ 추가 환경 설정은 하단 [외부 서비스] ⇒ 1. 카카오 로그인 참조

- 로컬 실행 시 **채팅** 기능을 이용하려면 URI를 변경해주어야 채팅이 원활하게 작동된다.

(현재는 배포 사이트 기준으로 URI가 셋팅되어 있음)

- 채팅 기능은 https 환경에서 동작하게 셋팅되어 있으므로, 로컬에서 이용 시 추가 환경설정이 요구된다.

⇒ 추가 환경 설정은 하단 [외부 서비스] ⇒ 3. 웹 소켓 채팅 참조

2. 배포 가이드

a. **직접 빌드** 후 배포

- 프로젝트 직접 빌드

```
$cd S05P13A501/backend  
$./gradlew clean build
```

- 프로젝트 실행

```
$cd S05P13A501/backend/build/libs  
$java -jar common-project-1.0-SNAPSHOT.jar
```

<https://i5a501.p.ssafy.io/> 로 접속

b. **Jenkins** 를 이용한 자동 빌드 및 배포

- Jenkins 관리자 페이지
 - <http://i5a501.p.ssafy.io/> 로 접속
 - 주의 : https가 아니라 **http** 입니다!
 - 보안오류 시, 크롬 브라우저에서 캐시를 삭제해주세요
- 직접 프로젝트 실행

```
$cd /var/lib/jenkins/workspace/build/backend/build/libs  
  
$java -jar common-project-1.0-SNAPSHOT.jar
```

외부 서비스

1. 카카오 API

- 카카오계정과 본 애플리케이션을 연결하고 토큰을 발급받아 카카오 API를 사용
- 직접 서비스 ID 및 비밀번호를 입력받고 검증하는 과정을 구현하는 대신, 카카오 로그인 을 통해 사용자를 인증하고 회원 정보를 수집
- 카카오 API가 제공되지 않는 가입 및 탈퇴 등 회원 정보에 대한 처리는 서비스에서 자체 적으로 구현

환경설정

```

backend
├── src
│   ├── java
│   │   ├── resources
│   │   │   ├── com
│   │   │   │   ├── ssafy
│   │   │   │   │   ├── api
│   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   └── KakaoAPI.java

```

- `KakaoAPI.java` 의 `APPKEY` 를 내 애플리케이션의 REST API로 셋팅
(이미 셋팅되어 있으므로 변경할 필요 없다)
 - 카카오 로그인을 위한 RESTAPI key : `bacd72f58ac01490602415c683ad8c05`
 - 카카오 공유하기를 위한 Javascript key : `d0106aa9ba1feb9c379bbb82194695fe`

```

@Service
public class KakaoAPI {

    public HashMap<String, Object> getAccessToken(String authorize_code){
        String access_Token = "";
        String refresh_Token = "";

        HashMap<String, Object> Token = new HashMap<>();
        String reqURL = "https://kauth.kakao.com/oauth/token";
        String APPKEY = "bacd72f58ac01490602415c683ad8c05";
    }
}

```

- 로컬에서 테스트 시 미리 설정해 놓은 `redirect URI` 를 바꿔주어야, 로그인 후에 로컬 페이지로 redirect된다.
(현재 설정은 배포 페이지로 redirect 되도록 셋팅)

redirect URI 변경법

1. Backend
 - a. KakaoApi.java

```

backend
├── src
│   ├── java
│   │   ├── resources
│   │   │   └── com

```

```

└─ssafy
  └─api
    └─service
      └─KakaoAPI.java

```

```

24 try {
25     URL url = new URL(reqURL);
26     HttpURLConnection conn = (HttpURLConnection) url.openConnection();
27
28     // POST 요청을 위해 기본값이 false인 setDoOutput을 true로
29     conn.setRequestMethod("POST");
30     conn.setDoOutput(true);
31
32     // POST 요청에 필요로 요구하는 파라미터 스트림을 통해 전송
33     BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(conn.getOutputStream()));
34     StringBuilder sb = new StringBuilder();
35     sb.append("grant_type=authorization_code");
36     sb.append("&client_id=").append(APPKEY);
37
38     // sb.append("&redirect_uri=https://localhost:8082/kakao/callback");
39     sb.append("&redirect_uri=https://i5a501.p.ssafy.io/kakao/callback"); // 배포용
40
41     sb.append("&code=" + authorize_code);
42     bw.write(sb.toString());
43     bw.flush();
44 }

```

38, 39번 라인의 `redirect_uri` 를 현재 환경에 맞는 `URI` 로 변경시켜주고, 해당하지 않는 URI는 주석처리한다.

- <https://i5a501.p.ssafy.io/kakao/callback> : 배포용
- <https://localhost:8082/kakao/callback> : 로컬테스트용

2. Frontend

a. login-dialog.vue

```

frontend
└─src
  └─views
    └─main
      └─login-dialog.vue

```

```

68
69 <script>
70 export default {
71   name: "login-dialog",
72   setup() {
73     if (!Kakao.isInitialized()) {
74       Kakao.init("bacd72f58ac01490602415c683ad8c05");
75     }
76     // 카카오 로그인
77     const loginWithKakao = function() {
78       Kakao.Auth.authorize({
79         redirectUri: "https://i5a501.p.ssafy.io/kakao/callback" //배포용
80         //redirectUri: "https://localhost:8082/kakao/callback"
81       });
82     };
83     return { loginWithKakao };
84   }
85 };
86 </script>
87

```

79,80번 라인의 `redirectUri` 를 현재 환경에 맞는 `URI` 로 변경시켜주고, 해당하지 않는 URI는 주석처리한다.

- <https://i5a501.p.ssafy.io/kakao/callback> : 배포용
- <https://localhost:8082/kakao/callback> : 로컬테스트용

카카오 로그인 방법

- 개인의 카카오 계정으로 자유롭게 서비스 이용 가능
(가입 시 모든 사항 동의 권장)
- 계정이 없을 시 하단에 기재한 공용 아이디로 이용 권장
 - ID : doglivetest@kakao.com
 - PW : ssafy123

2. AWS S3 & CloudFront

- AWS S3와 CloudFront를 연동한 이미지 서버 활용
- 환경 설정

- aws s3 서버의 계정 및 환경 정보가 담긴 `appliation-aws.yaml` 파일을 `application.yaml` 에 import 시켜놓은 상태
(추가 수정 필요 없음)

3. 웹소켓 채팅

- 로컬 테스트 시 url 변경 필요
(ec2 배포시 변경할 필요 없음)

a. chat-detail.vue

```

frontend
├── src
│   ├── views
│   │   ├── chat
│   │   └── chat-detail.vue

```

```

288
289 // 웹 소켓 통신 Connect
290 function connect(){
291     const url = "https://i5a501.p.ssafy.io/api/v1/chat-server" // 배포용
292     //const url = "https://localhost:8081/api/v1/chat-server"
293     socket = new SockJS(url, { transports: ['websocket', 'xhr-streaming', 'xhr-polling']})
294     client = Stomp.over(socket)
295     client.connect({withCredentials : true, userId : userId }
296         ,frame => {
297             console.log("연결 성공", frame);
298             onConnected()
299         }
300         ,error => {
301             console.log("연결 실패", error);
302         })

```

291, 292번 라인의 `url` 를 현재 환경에 맞는 URI로 변경시켜주고, 해당 하지 않는 URI는 주석처리한다.

- <https://i5a501.p.ssafy.io/api/v1/chat-server> : 배포용
- <https://localhost:8081/api/v1/chat-server> : 로컬테스트용

+ 단, HTTPS 인증서 등록이 되어 있지 않은 로컬 환경에서는 채팅 기능에 추가 설정이 필요하다.



채팅방에 들어 갔을 때, 콘솔창에 다음과 같은 보안오류가 뜨고 무한 로딩이 진행된다면 빨간 박스에 해당하는 URL을 클릭한다.



연결이 비공개로 설정되어 있지 않습니다.

공격자가 **localhost**에서 정보(예: 비밀번호, 메시지, 신용카드 등)를 도용하려고 시도 중일 수 있습니다. [자세히 알아보기](#)

NET::ERR_CERT_AUTHORITY_INVALID



Chrome에서 가장 강력한 보안 기능을 사용하려면 [향상된 보호 모드를 사용 설정](#)하세요.

세부정보 숨기기

안전한 페이지로 돌아가기

이 서버가 **localhost**임을 입증할 수 없으며 컴퓨터의 운영체제에서 신뢰하는 보안 인증서가 아닙니다. 서버를 잘못 설정했거나 불법 사용자가 연결을 가로채고 있기 때문일 수 있습니다.

[localhost\(안전하지 않음\)\(으\)로 이동](#)

URL 클릭 시 다음과 같은 경고 페이지가 뜨며 '고급 - localhost(안전하지 않음)으로 이동'을 클릭해준다. 그러면 'Welcome to SockJS!'라는 글이 적힌 페이지가 뜨며, 이제 서비스로 돌아가서 새로고침을 하면 웹 소켓을 이용할 수 있다.