1. 문서 정리
   1. Liberica 17, EC2, Tomcat(Spring Boot 내장), 설정 값은 application.properties의 server.으로 시작하는 프로퍼티 값 참고, IntelliJ IDEA 2024.1.4
   2. 빌드 시 사용되는 환경 변수 등의 내용 상세 기재
      1. 프론트
         1. REACT\_APP\_CLIENT\_ID\_NAVER = JsIIDM6WtazKOs8qioPk
         2. REACT\_APP\_CLIENT\_SECRET\_NAVER = h9wqTMRbA1
         3. REACT\_APP\_CALLBACK\_URL\_NAVER = http://localhost:3000/user/login
         4. REACT\_APP\_CLIENT\_ID\_GOOGLE = 166624659423-ptsp3i4a8kuui6c6kkmlmitfiia000op.apps.googleusercontent.com
         5. REACT\_APP\_CLIENT\_SECRET\_GOOGLE = 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
         6. REACT\_APP\_CALLBACK\_URL\_GOOGLE = http://localhost:3000/user/login
         7. REACT\_APP\_CLIENT\_ID\_KAKAOTALK = ebd33aba8c52b24e63d656e7ef2b0e66
         8. REACT\_APP\_CALLBACK\_URL\_KAKAOTALK = http://localhost:3000/user/login
         9. REACT\_APP\_CLIENT\_ID\_NAVER\_MAP = w8wb535abw
         10. REACT\_APP\_CLIENT\_SECRET\_NAVER\_MAP = lCYDeRoMc6UA54M2Nallhh0UBj3neVjq3C9W4Oji
         11. REACT\_APP\_SEARCH\_API = 67DD3CD1-474E-373F-9935-501BBF3021BE
         12. REACT\_APP\_BACKEND\_URL = https://i11d206.p.ssafy.io/api
         13. REACT\_APP\_BACKEND\_URL = <http://localhost:8080/api>
      2. 백엔드
         1. #cloud
            1. cloud.aws.stack.auto=false
            2. cloud.aws.region.static=ap-northeast-2
            3. cloud.aws.credentials.access-key=AKIA45ZPUQU4JGCVYJ7V
            4. cloud.aws.credentials.secret-key=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
            5. cloud.aws.s3.bucket=bangtong-bucket
            6. cloud.aws.s3.bucket.url=https://s3.ap-northeast-2.amazonaws.com/bangtong-bucketd
         2. Redis
            1. #spring.data.redis.host=i11d206.p.ssafy.io
            2. spring.data.redis.host=localhost
            3. spring.data.redis.port=6379
            4. spring.data.redis.password=rufrhkwltkdwndmlS11!
         3. SMTP
            1. spring.mail.host=smtp.gmail.com
            2. spring.mail.port=465
            3. spring.mail.username=noreply.bangtong@gmail.com
            4. spring.mail.password=inanugqyudmlmxyi
            5. spring.mail.properties.mail.smtp.auth=true
            6. spring.mail.protocol=smtps
            7. spring.mail.properties.mail.smtp.timeout=20000
            8. spring.mail.properties.mail.smtp.connectiontimeout=20000
            9. spring.mail.properties.mail.smtp.writetimeout=20000
   3. backend 서버(폴더명: back)를 빌드할 때 ./mvnw이 아니라 mvn 명령어를 써야 합니다.
      1. DB: hostname: stg-yswa-kr-practice-db-master.mariadb.database.azure.com / 비밀번호: rkskekfk1234! / application.properties의 datasource.가 붙는 프로퍼티 값 참고
2. 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서
   * 1. OAUTH 2.0: GOOGLE
        1. spring.security.oauth2.client.registration.google.client-id=137286417252-41cvbmuvk4jgpgshf7i6kmduksjmfeue.apps.googleusercontent.com,
        2. spring.security.oauth2.client.registration.google.client-secret=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,
        3. spring.security.oauth2.client.registration.google.redirect-uri=https://{baseHost}{basePort}/login/oauth2/code/google,
        4. spring.security.oauth2.client.registration.google.scope=email, profile
     2. OAUTH 2.0 Naver
        1. spring.security.oauth2.client.registration.naver.client-id=bHi2ckoAKi4wwsYyzaTe
        2. spring.security.oauth2.client.registration.naver.client-secret=eNIdgyYWfb
        3. spring.security.oauth2.client.registration.naver.redirect-uri=http://localhost:8080/login/oauth2/code/naver
        4. spring.security.oauth2.client.registration.naver.authorization-grant-type=authorization\_code
        5. spring.security.oauth2.client.registration.naver.client-name=Naver
        6. spring.security.oauth2.client.registration.naver.scope=name, email
        7. spring.security.oauth2.client.provider.naver.authorization-uri=https://nid.naver.com/oauth2.0/authorize
        8. spring.security.oauth2.client.provider.naver.token-uri=https://nid.naver.com/oauth2.0/token
        9. spring.security.oauth2.client.provider.naver.user-info-uri=https://openapi.naver.com/v1/nid/me
        10. spring.security.oauth2.client.provider.naver.user-name-attribute=response
3. KAKAO
   * 1. OAUTH 2.0: KAKAO
        1. spring.security.oauth2.client.registration.kakao.client-id=706413bb2184c7ac1a655ef81304689b
        2. spring.security.oauth2.client.registration.kakao.redirect-uri=http://localhost:8080/login/oauth2/code/kakao
        3. spring.security.oauth2.client.registration.kakao.client-authentication-method=client\_secret\_post
        4. spring.security.oauth2.client.registration.kakao.authorization-grant-type=authorization\_code
        5. spring.security.oauth2.client.registration.kakao.scope=profile\_nickname, account\_email
        6. spring.security.oauth2.client.registration.kakao.client-name=Kakao
        7. spring.security.oauth2.client.provider.kakao.authorization-uri=https://kauth.kakao.com/oauth/authorize
        8. spring.security.oauth2.client.provider.kakao.token-uri=https://kauth.kakao.com/oauth/token
        9. spring.security.oauth2.client.provider.kakao.user-info-uri=https://kapi.kakao.com/v2/user/me
        10. spring.security.oauth2.client.provider.kakao.user-name-attribute=id
        11. vworld api 검색 API -> 검색을 통해 사용자나 매물 근처의 편의 시설 조회하는 용도로 사용
        12. Geocoder API -> 사용자의 IP 주소를 바탕으로 한글 주소를 반환 받거나, 주소를 통해 매물이나 사용자의 위경도를 받거나, 관리하기 용이하게 regionId 를 반환받기 위해 사용