



아키텍처 설계 초안

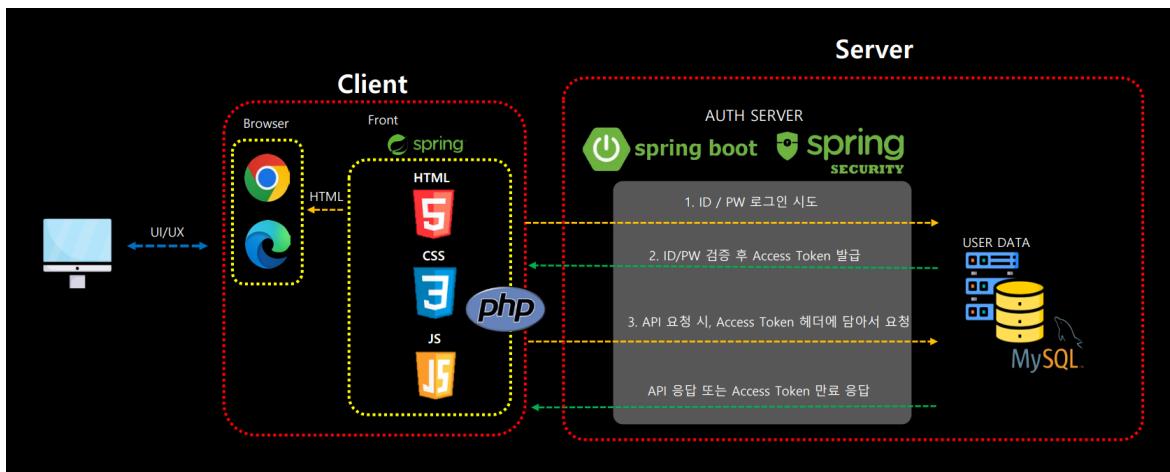
- 아키텍처 설계
 - 팀 전체 서버들을 하나의 그림으로 도식화
 - 각자 맡은 서버 각각에 대한 구체적 그림

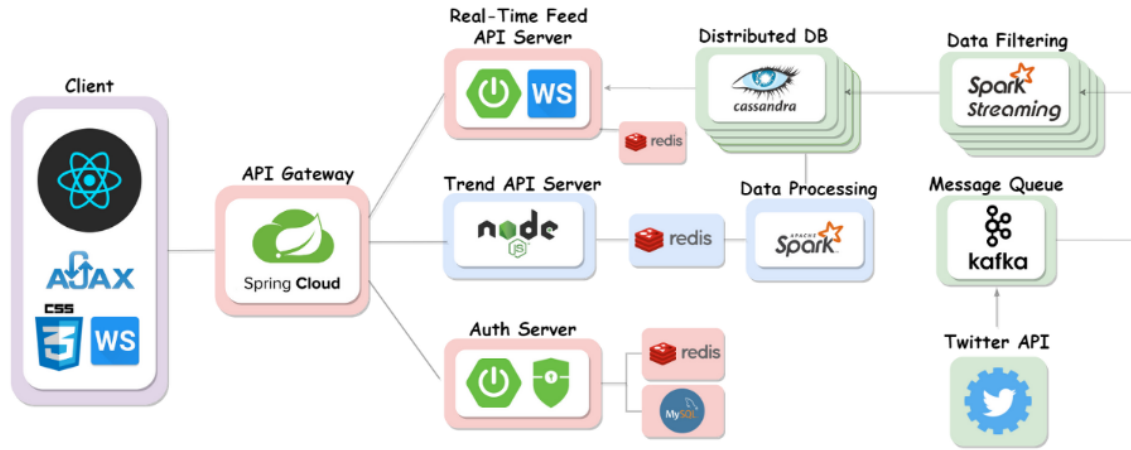
- 아키텍처 툴
 - gliffy
 - LucidChart
 - CloudSkew
 - CloudCraft
 - Prezi(?)

- 팀원들과 공유하면 좋을 링크

<https://velog.io/@agugu95/%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-%EA%B2%B0%EC%A0%95-%EC%82%AC%ED%95%AD>

- 아키텍처 초안





- 인증(로그-인&아웃, 회원가입...)

<https://github.com/belljun3395/msa-jongjun>

2 Spring Security 기본 - 로그인, 로그아웃 인증

스프링 시큐리티는 스프링 기반의 어플리케이션의 보안을 담당하는 프레임워크이다. 스프링 시큐리티가 구성하는 영역은 두가지이다. 인증(Authentication) : 애플리케이션의 작업을 수행할 수 있는 주체 권한(Authorization) : 인증된 주체가 특정 동작을 수행할 수 있는 권한을 가졌는지를 결정 스프링 시큐리티는 보안과 관련해 많은 옵션

<https://jungeunlee95.github.io/java/2019/07/17/2-Spring-Security/>

GitHub - gkdud583/auth-server: smilgate 인증 시스템 구현 리포지토리

사용자는 이메일과 비밀번호 정보를 이용하여 회원가입 후 로그인한다. 서버는 정보 확인 후 access token(인증용, 유효시간 30분), refresh token(access token 재발급용, 유효시간 2시간, 1회 사용 가능) 을 응답으로 보내준다. 사용자는 refresh token이 유효하다면 refresh token를 이용해서 access token을 재발행 할 수 있다.

<https://github.com/gkdud583/auth-server>

gkdud583/auth-server

smilgate 인증 시스템 구현 리포지토리

Rx 1 Contributor, 0 Issues, 0 Stars, 0 Forks

스프링 Security_로그인_로그인 성공 대응 로직 [4/9]

Develop OS : Windows10 Ent, 64bit - WEB/WAS Server : Tomcat v9.0 - DBMS : MySQL 5.7.29 for Linux (Docker) - Language : JAVA 1.8 (JDK 1.8) - Framework : Spring 3.1.1 Release - Build Tool : Maven 3.6.3 - ORM : Mybatis 3.2.8 커스터마이징 순서대로 총 9개의 포스팅으로 나뉘져 있습니다.

<https://codevang.tistory.com/269>

[JWT, OAuth]서브 도메인 분리와 인증 인가 서버 설계 소개

기존에 서비스A와 추가 개발 진행하고 있는 서비스B의 도메인이 추가하게 됨과 동시에, 통합 인증인가를 위한 서버를 별도로 구성하기하였다. 기존 : www.domain.com 신규 : sub.doamin.com 서버의 구성은 다음과 같다. 각 서비스의 인증과 인가는 Auth Server와 통신하여 이루어 질 것이다. 여기서, 인증 권한은 크게 세션 과 토큰으로

<https://granger.tistory.com/16>

Spring Boot로 만드는 Spring Security 로그인 구현 - JWT(2)

코드는 Github를 통해 확인할 수 있습니다. JWT 로그인 방식은 CSRF와 REST API에 대한 지식이 요구 됩니다. 목차 화면 구성 JWT(JSON Web Token) 소개 JWT 한계 코드 분석 - REST API 구현과 JWT 구현을 분리해서 설명합니다. 화면 구성 JWT(JSON Web Token) 소개 세션, 쿠키의 로그인 방식과 JWT의 가장 큰 차이점은 서

<https://stir.tistory.com/275?category=1039528>

http://localhost:8080

사용자이름

비밀번호

Spring Security : 계정 생성 + 권한 설정 + 인증/인가 설정

이 글은 인프런 정수원님의 강의를 복습하며 작성한 글입니다. Spring Security : 계정 생성 + 권한 설정 + 인증/인가 설정 Spring Security는 기본적으로 인증 / 인가를 바탕으로 동작한다. 인증/인가를 위해서 필요한 것은 바로 계정이고, 그 계정에 어떤 권한이 설정되는지가 중요하다. 그리고 어떤 자원으로 접근할 때 인증이 필요한지, 그리


<https://oj90902.tistory.com/844>

• 네이버 로그인 관련

Spring Boot + Spring Security + OAuth2 네이버 로그인 해보기

외부 로그인인 기존 프로젝트에 적용해본적이 없어서 OAuth2에 대해 알아보고 정리해보았다. 네아로 등록 애플리케이션 이름: 사용할 애플리케이션의 이름을 작성해주면 된다. 사용 API: 네이버 로그인 선택 제공 정보 선택: 서비스에 활용할 정보들을 필수로 추가하거나 추가하면 된다. 여기에서는 별명(nickname), 이메일주소(email),

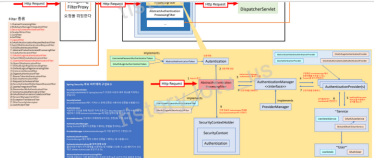
<https://velog.io/@mardi2020/Spring-Boot-Spring-Security-OAuth2-%EB%84%A4%EC%9D%B4%EB%B2%84-%EB%A1%9C%EA%B7%B8%EC%9D%B8-%ED%95%B4%EB%B3%B4%EA%B8%B0>



[Spring Security] OAuth2 네이버 로그인하기

목차 이전글 <https://lotuus.tistory.com/79> [Spring Security] OAuth2 구글 로그인하기 목차 [이전 게시글] 꼭! 봐주세요 [Spring Security] 동작방법 및 Form, OAuth2 로그인하기 (Feat.Thymeleaf 타임리프) 목차 Spring Security란? Spring을 사용할 때 애플리케이션에 대한 인증, 권한 부여 등의 보.. lotuus.tistory.com Spring


<https://lotuus.tistory.com/80>



[SpringBoot] (7) Spring Security OAuth 2.0 로그인 기능 (네이버 소셜 로그인)

네이버가 해주는 내 로그인 기능 [네이버 오픈 API](#) 이렇게 등록까지 완료하면 clientID와 clientSecret이 생성된다. 이제 이 키 값들을 application-oauth.properties에 등록한다. 그런데 이때, 네이버에서는 스프링 시큐리티를 공식 지원하지 않기 때문에 Common-OAuth2Provider에서 해주던 값들도 전부 수동 입력 해야 한다. [클](#)

<https://velog.io/@dbsrud11/SpringBoot-7-Spring-Security-OAuth-2.0-%EB%A1%9C%EA%B7%B8%EC%9D%B8-%EA%B8%B0%EB%8A%A5-%EB%84%A4%EC%9D%B4%EB%B2%84-%EC%86%8C%EC%85%9C-%EB%A1%9C%EA%B7%B8%EC%9D%B8>



• 네이버 로그인 + 일반 로그인

Spring Boot: "네이버 아이디로 로그인하기" 연동 - 스프링 시큐리티와 연결 (1) - BGSMM

깃허브에서 전체 코드 보기 - <https://github.com/ayaysir/spring-boot-security-example-1> 보다 개선된 네이버 로그인 - 스프링 부트(Spring Boot): 구글 로그인 연동 (스프링 부트 스타터의 oauth2-client) 이용 + 네이버 아이디로 로그인 이전글의 두 상황을 결합하여 네이버 아이디로 로그인(이하 네아로)을 스프링 시큐리티와 연결하는

<http://yoonbumtae.com/?p=1940>

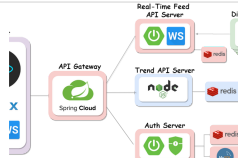
word	role	food	em
2\$D6IW2YxEzlsUJlUkIplUuhyRgCff4d4steNlOrdD6G1...	ADMIN	Chicken	
2\$D6IW2YxEzlsUJlUkIplUuhyRgCff4d4steNlOrdD6G1...	GUEST	Pizza	
0\$SxSWKcc1MjBfVb26fmy27e8FfJ0kIjwXgfaYtksXUE...	GUEST	치킨	
0\$39mkmJ3Q510Jd8sEtf0BeOTW5.nyTH5qHk7K5/6j3K...	GUEST	피자	gol
0\$vdVUtwffqXs9Kpb.GMyLauXtZdlH12J8Fxp6/s/mzCY...	GUEST	돼지수육	ii@
0\$AJUn9V6WlaJdfNudhalCFe30BMkhWwLkqZqhg7jhc...	GUEST	고기	che

• 전체 서버 아키텍처

[API Gateway + Refresh JWT 인증서버 구축하기] Spring boot + Spring Cloud Gateway + Redis + mysql JPA 1편

로그인 한 번으로 여러 서비스를 이용하는 것. Google 아이디 가지고 여러 서비스를 사용하는 것이 대표적인 예시인데 OAuth2와 아주 깊게 연관되어 있는 듯 하다. SSO라고 하면 '하나의 로그인으로 여러 서비스 이용하기'라는 컨셉 자체이고 OAuth는 그것을 실행하기 위한 대표적인 기술이라고 생각하면 될 것 같다.


<https://velog.io/@tlatdms/API-Gateway-Refresh-JWT-%EC%9D%B8%EC%A6%9D%EC%84%9C%EB%B2%84-%EA%B5%AC%EC%B6%95%ED%95%98%EA%B8%B0-Spring-boot-Spring-Cloud-Gateway-Redis-mysql-JPA-1%ED%8E%B8>



[서버개발캠프] MSA 아키텍처의 API Gateway 프레임워크 결정

우리 프로젝트 MSA 아키텍처 심슨20200123104504_1.png API Gateway > API 게이트웨이는 클라이언트 앱과 마이크로서비스 사이에 위치합니다. 클라이언트에서 서비스로 요청을 라우팅하는 역할만 프록시로 사용됩니다. 또한 인증, SSL 종료 및 캐시와 같은 추가 교차 편집 기능을 제공할 수 있습니다. 사용자의 API 게이트웨이


<https://velog.io/@tlatdms/%EC%84%9C%EB%B2%84%EA%B0%9C%EB%B0%9C%EC%BA%A0%ED%94%84-MSA-%EC%95%84%ED%82%A4%ED%85%8D%EC%B3%90%EC%9D%98-API-Gateway-%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC-%EA%B2%B0%EC%A0%95>



스프링 시큐리티 - JWT 인증 방식 도입 하기

진행 중인 프로젝트 랜손켓 애플리케이션의 백엔드 서버를 개발하면서 인증, 인가와 관련된 서버 보안 설계에 대해 정리해보려고 한다. 기존 설계 방식의 제한점 -> 대안 적용 -> 결과 -> 얻은 점 [1](#) 보여주기 부끄러운 초기 서버 구상도 채팅 애플리케이션의 유저들을 관리하기 위해 인증 방식을 고민하다가 스프링 시큐리티에서 아주 간단

<https://devwithpug.github.io/project/randhand-1-spring-security-with-jwt/>



• 참고가 가능한 수준의 아키텍처

<https://techblog.woowahan.com/6900/>

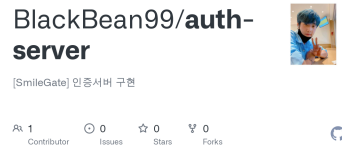
• 깃허브 참고

https://github.com/sienna011022/msa_auth_smilegate

GitHub - hscom96/msa-auth: 스마일게이트 데브캠프 개인 프로젝트 #2 '인증서버'
총 1인 회원가입 로그인 이메일 인증 사용자 정보 조회 (관리자권한만 가능) ✓ 회원가입 구현과정
<https://lovecode.tistory.com/122> ✓ 로그인 구현과정 <https://lovecode.tistory.com/124> Springboot MYSQL, JPA spring cloud (zuul, eureka) HTML CSS JavaScript JQuery gateway : 8090 auth-server : 8081 user-
🔗 <https://github.com/hscom96/msa-auth>



GitHub - BlackBean99/auth-server: [SmileGate] 인증서버 구현
SmileGate] 인증서버 구현. Contribute to BlackBean99/auth-server development by creating an account on GitHub.
🔗 <https://github.com/BlackBean99/auth-server>



GitHub - hayarobys/spring-security-for-ajax: Restful서버를 대상으로 한 Spring Security 커스텀 프로젝트 입니다. 웹기반 권한/계정 관리 기능을 포함
Restful서버를 대상으로 한 Spring Security 커스텀 프로젝트 입니다. 웹기반 권한/계정 관리 기능을 포함합니다. - GitHub - hayarobys/spring-security-for-ajax: Restful서버
상으로 한 Spring Security 커스텀 프로젝트 입니다. 웹기반 권한/계정 관리 기능을 포함합니다.

🔗 <https://github.com/hayarobys/spring-security-for-ajax>

• 스프링 시큐리티 자료(github)

<https://github.com/Youngerjesus/spring-security>

• 서버개발캠프 관련 자료

[서버개발캠프] Spring boot + Spring security + Refresh JWT + Redis + JPA 1편
ORM: Object-relational mapping (객체 관계 매핑) 객체는 객체대로 설계하고, 관계형 데이터베이스는 관계형 데이터베이스대로 설계한다. ORM 프레임워크가 중간에서 매핑해준다. JPA: Java Persistence API ORM을 사용해 오브젝트와 RDB 사이에 존재하는 개념과 접근을 객체지향적으로 다루기 위한 기술 관계형 데이터베이스를
📖 <https://velog.io/@tlatdms/Spring-boot-Spring-security-Jwt-1-fvk5h10l8u>

velog

[스마일게이트 서버 개발 캠프 4기]#3 두번째 개인과제 리뷰(로그인 인증 서버개발)
스마일게이트 서버개발 캠프 두번째 개인과제는 인증서버개발이었다. 처음 인증서버를 들었을때는 뭔지 감이 오지 않았다. 지금까지 개발을 해오면서 구현해온 로그인, 회원가입 방식은 DB에 접근해 아이디 있는지 체크해서 로그인 이런 간단한 방식만 생각했기 때문이다. 하지만 이번 과제를 수행하면서 회원가입, 로그인에 정말
🔗 <https://cjw-awdsd.tistory.com/4>



🚀 스마일게이트 서버개발캠프 4기 - 회고
스마일게이트에서 두 달 간 진행되었던 서버개발캠프가 종료되었다. 코로나 때문에 갑작스럽게 마무리되어 아쉬웠지만, 많은 것을 얻을 수 있었던 시간이었다. 이번 포스팅에서 서버개발캠프를 통해 얻은 점과 느낌점에 대해 적어보려한다. RESTful API 개발 이번 프로젝트에서는 클라이언트와 서버를 분리하여 개발을 진행하게되었다. 네
🔗 <https://yunb2.tistory.com/10>

이름	담당	역할	기술	주요 업무	주요 성과
김민준	개발	백엔드	Spring, Spring Security, JWT, Redis, MySQL, JPA, Zuul, Eureka	인증서버 개발	로그인, 회원가입, 비밀번호 변경 기능 구현
이민준	개발	백엔드	Spring, Spring Security, JWT, Redis, MySQL, JPA, Zuul, Eureka	인증서버 개발	로그인, 회원가입, 비밀번호 변경 기능 구현
정민준	개발	백엔드	Spring, Spring Security, JWT, Redis, MySQL, JPA, Zuul, Eureka	인증서버 개발	로그인, 회원가입, 비밀번호 변경 기능 구현
박민준	개발	백엔드	Spring, Spring Security, JWT, Redis, MySQL, JPA, Zuul, Eureka	인증서버 개발	로그인, 회원가입, 비밀번호 변경 기능 구현
최민준	개발	백엔드	Spring, Spring Security, JWT, Redis, MySQL, JPA, Zuul, Eureka	인증서버 개발	로그인, 회원가입, 비밀번호 변경 기능 구현

[서버개발캠프] 인증 서버 - 소셜로그인

네이버 아이디로 로그인 인증서버에 네이버 아이디로 로그인(네아로) 기능을 추가했다. 네아로의 전체적인 로직은 다음과 같다. 1. NAVER Developers에서 Application을 등록해 client_id와 client_secret을 발급받는다. 2. client_id로 access_code를 요청한다. 3. client_id와 client_secret, access_code로 사용자 정보에 접근할 수
👉 <https://yunb2.tistory.com/9>

네이버



인증서버

- 1 네이버 아이디로 로그인 요청(client_id)
- 2 access_code 발급
- 3 소셜 로그인 요청(access_code)
- 4 access_token 및 refresh_token 발급 요청(a)
- 5 access_token 및 refresh_token 발급
- 6 사용자 정보 요청(access_token)
- 7 사용자 정보 제공
- 8 사용자 정보 확인 후 access_token 및 refre

• 이메일 인증

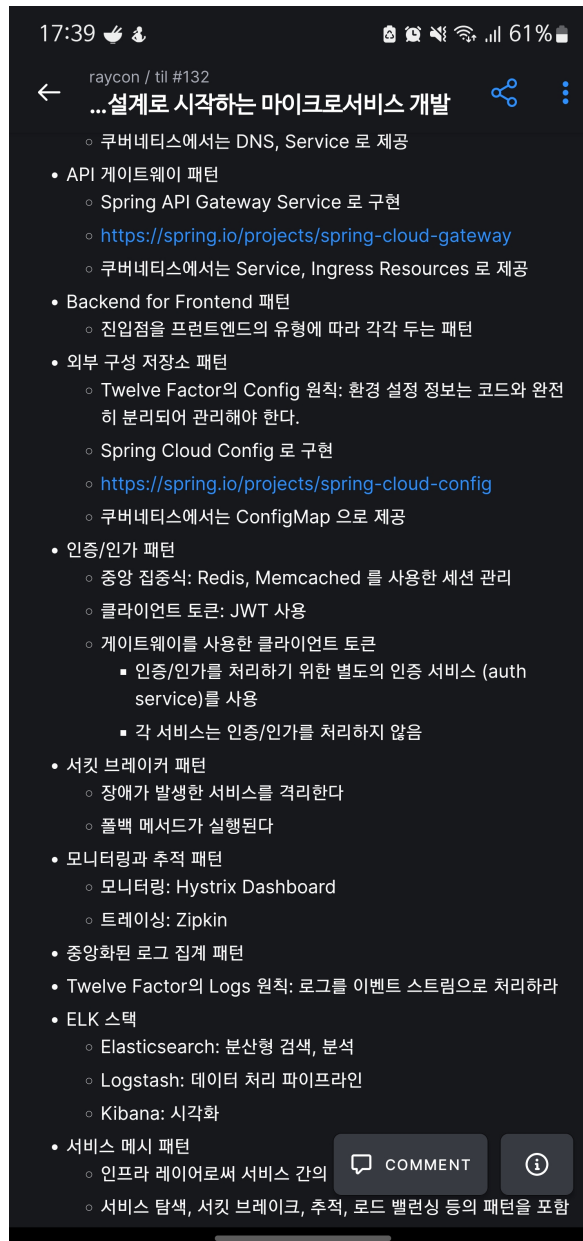
Spring를 이용한 이메일 인증(feat. 네이버,구글)

프로젝트에서 일반 회원가입에서 회원의 인증을 위해서 이메일 인증 기능을 넣게 되었다. 기능을 구현하면서 진행했던 절차들을 정리해보도록 하겠다. google SMTP 서버, Naver SMTP 서버등 다양한 서버를 사용하여 메일을 전송할수 있는데 이번 프로젝트에서는 Naver SMTP 서버를 이용하여 메일 인증을 진행하였다. 하지만 Google

📄 <https://velog.io/@mqhstr2297/Spring%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EC%9D%B4%EB%A9%94%EC%9D%BC-%EC%9D%B8%EC%A6%9Dfeat.-%EB%84%A4%EC%9D%B4%EB%B2%84>

velog

• 그외 공부 자료



https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4f749cea-a039-4cdc-8468-bf7778731fdb/%EC%84%9C%EB%B2%84%EA%B0%9C%EB%B0%9C%EC%BA%A0%ED%94%84_%EC%98%88%EC%8B%9C_PPT.pdf

• 유저 관리 서버 관련

[spring/java] 웹사이트 만들기 5 - 회원정보 수정 / 마이페이지(my page)

이전 포스팅들에서는 spring/java를 이용하여 회원가입 인증, 로그인/로그아웃, 아이디/비밀번호 찾...

<https://m.blog.naver.com/heartflow89/221115916689>

```

<script type="text/javascript">
    $(function() {
        $('#loginForm').submit(function() {
            $.ajax({
                url: '/login',
                type: 'POST',
                data: $('#loginForm').serialize(),
                success: function(response) {
                    // 로그인 성공
                },
                error: function(response) {
                    // 로그인 실패
                }
            });
        });
    });

```

JAVA / SPRING 프로젝트 삼삼하개(3) 마이페이지(개인/업체) 내 작성글, 작성댓글 조회 구현

팀프로젝트가 끝나고 되돌아보는 시간 (3) 마이페이지(개인/업체) 내 작성글, 작성댓글 조회 구현 내 작성글, 작성 댓글을 불러오는 페이지를 작업하면서 돌이킬수 없는 후회가 생겼습니다. 바로 테이블 설정.. 구성 당시에는 먼 미래를 생각하며 각게시판을 카테고리화해서 구성하면 데이터가 많이 쌓였을때 성능저하가 생기지 않을까? 라는 의

<https://recoendr.tistory.com/9>

자유게시판(3)	제목	작성일
1016	이제야사여	2021-01-28 00:27:20.0
1015	이제야를 찾아줘	2021-01-27 00:42:13.0
1014	이제야를	2021-01-27 00:42:58.0

스프링으로 이미지 업로드 구현하기

프론트 : html-formData를 보낼 input, form 태그 작성백 : Controller, AWS마이페이지 컨트롤러에 작성, 직원의 ID와 이미지의 이름이 똑같도록 하기split을 통해 png, jpeg, jpg 확장자를 지원하도록 하기 AWS의 이미지 서버

<https://velog.io/@justdoitho1/%EC%8A%A4%ED%94%84%EB%A7%81%EC%9C%BC%EB%A1%9C-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%97%85%EB%A1%9C%EB%93%9C-%EA%B5%A C%ED%98%84%ED%95%98%EA%B8%B0>



[Spring Boot/JPA] #3 View와 Controller 제작 : 마이페이지 만들기 (1)

Spring MVC 구현, 로그인한 사용자의 마이페이지 구현, Spring Security를 통해 현재 사용자 정보 얻기, Paging 구현

<https://prod.velog.io/@dls4585/Spring-BootJPA-3-View%EC%99%80-Controller-%EC%A0%9C%EC%9E%91-%EB%A7%88%EC%9D%B4%ED%8E%98%EC%9D%B4%EC%A7%80-%EB%A7%8C%EB%93%A4%EA%B8%B0-1>



• 네이버 자체 회원 정보 수정 정보

이하얀 44 99

기본정보

이하얀
dlgkdis801@naver.com

+82 10-9***-8**

wn*****@n*****.com

wn*****@n*****.com

프로션 정보수신 동의

휴대전화 ON

이메일 OFF


생일 관리 친구에게 보여줄 내 생일을 설정하세요. 확인 >

배송지 관리 쇼핑, 예약 등에서 사용하신 배송지를 관리할 수 있어요. 총3건 >

개인정보 이용내역 조회

프로필 수정

네이버 대표 프로필과 별명을 수정하실 수 있습니다.

프로필 사진	 <input type="button" value="사진변경"/> <input type="button" value="캐릭터 만들기"/> <input type="button" value="삭제"/>
별명	<input type="text" value="이하얀"/>

- URI 관련

기능			
API	METHOD	URI	ROLE
회원가입	POST	/api/v1/users/register	PUBLIC
로그인	POST	/api/v1/users/login	PUBLIC
로그아웃	POST	/api/v1/users/logout	USER OR ADMIN
토큰 재발행	POST	/api/v1/users/token/refresh	USER OR ADMIN
전체 유저 조회	GET	/api/v1/users	ADMIN

```

main / auth-server / src / main / resources / static / js / registerjs / <> Jump to
gkdud583 feat: 페이지 구현 Latest commit dac3b2e 3 weeks ago History
1 contributor
30 lines (27 sloc) 794 Bytes
1
2 let registerForm = document.querySelector("#registerForm");
3 let emailInput = document.querySelector("#email");
4 let passwordInput = document.querySelector("#password");
5
6 function register() {
7   fetch('http://localhost:8080/api/v1/users/register', {
8     method: 'POST',
9     cache: "no-cache",
10    headers: {
11      'Content-Type': 'application/json'
12    },
13    body: JSON.stringify({
14      email: emailInput.value,
15      password: passwordInput.value,
16    })
17  })
18  .then(async (response) => {
19    if (response.status == 200) {
20      alert("회원가입 되었습니다!");
21      window.location.href = "http://localhost:8080/users/login";
22    }
23    else {
24      let json = await response.json();
25      alert(json.message);

```

- RESTful API의 URI 작성 규칙

RESTful API의 URL 작성 규칙

REST API란? REST(Representational State Transfer)는 HTTP 네트워크 상의 리소스(Resource, 자원)를 정의하고 해당 리소스를 URI라는 고유한 주소로 접근하는 접근 방식을 의미하며, REST API란 REST 방식을 통해서 리소스에 접근하기 위한 서비스 API를 지칭합니다. REST에서 의미하는 리소스란? REST에서 의미하는 자원

🔗 <https://itvillage.tistory.com/35>




- Response(응답)

HTTP 서버 응답 코드 (Response Code) 정리

클라이언트(웹 브라우저)에서 서버에 어떠한 요청(리퀘스트, Request)를 하면 서버는 응답(리스폰스, Response)을 보내줍니다. 적절한 처리가 이루어져서 성공 응답과 함께 결과 값을 보내주기도 하고, 정상적인 처리가 되지 않은 경우에는 실패 응답과 함께 에러 정보를 보내주기도 합니다. 웹 개발자라면 기본적으로 서버가 정


🔗 <https://codediver.tistory.com/132>



REST API 를 보다 RESTful 하게 만들기

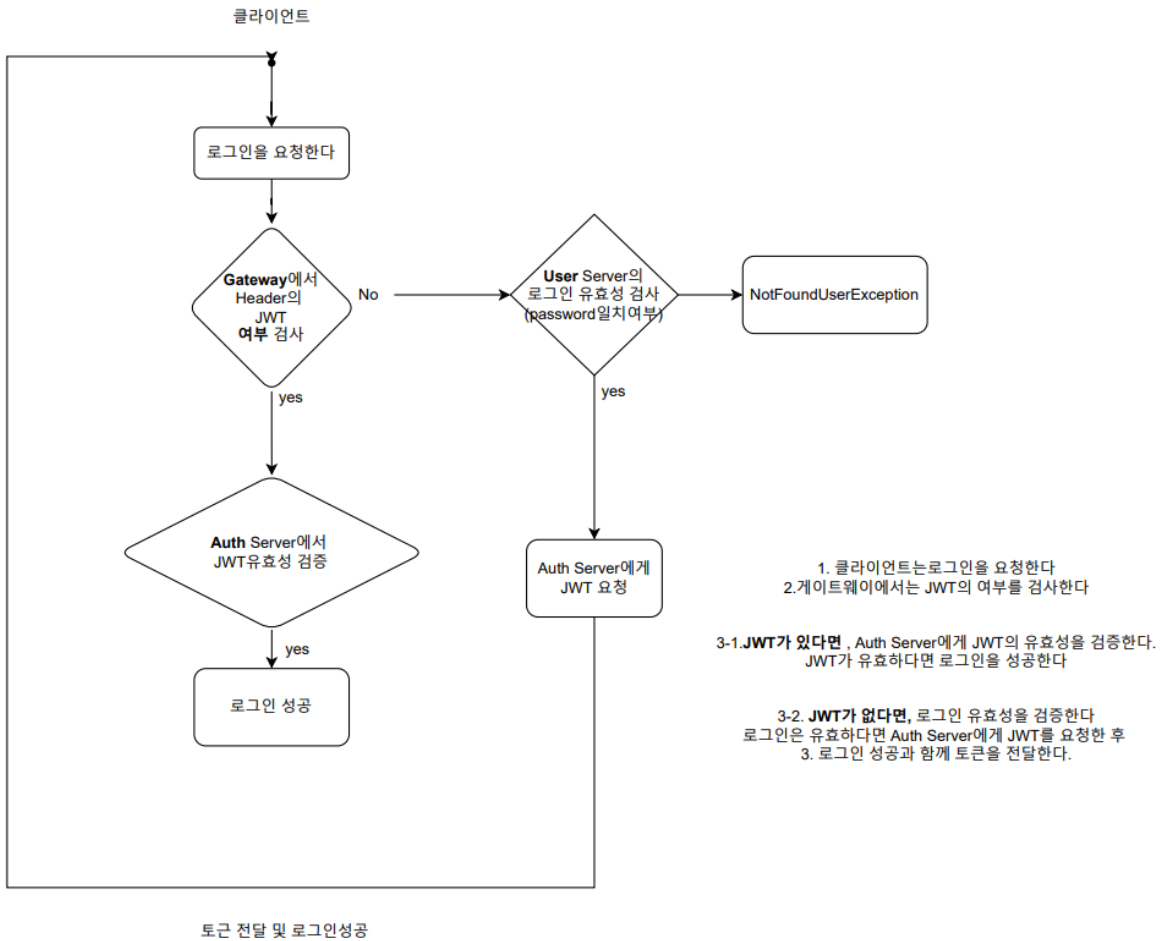
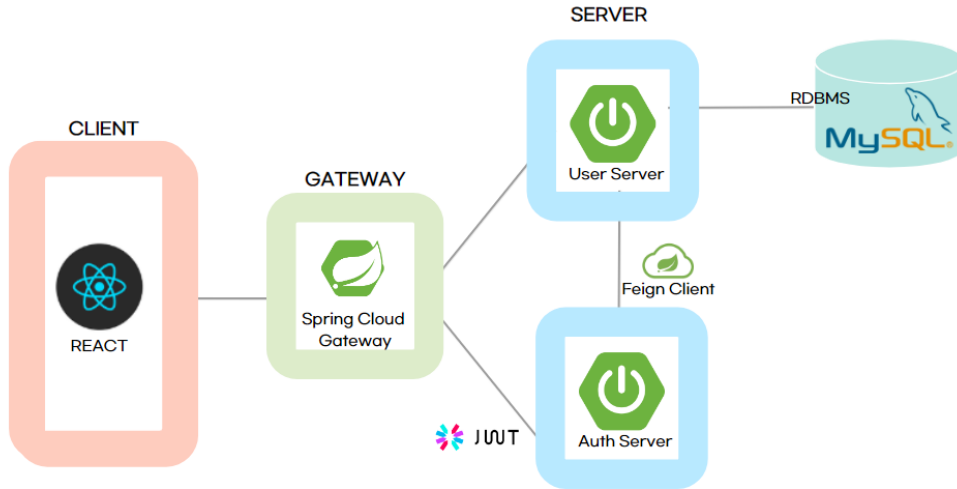
REST API 인증 파트가 어느정도 마무리되고 써볼 글은 API 에 대한 이야기다. 가장 처음, REST(Representational State Transfer) API 에 대한 이야기를 해보고자 한다. REST API 는 SPA(Single Page Application) 방식으로 개발된 프론트엔드에서 백엔드의 데이터를 가져올 때 가장 많이 사용되는 자원

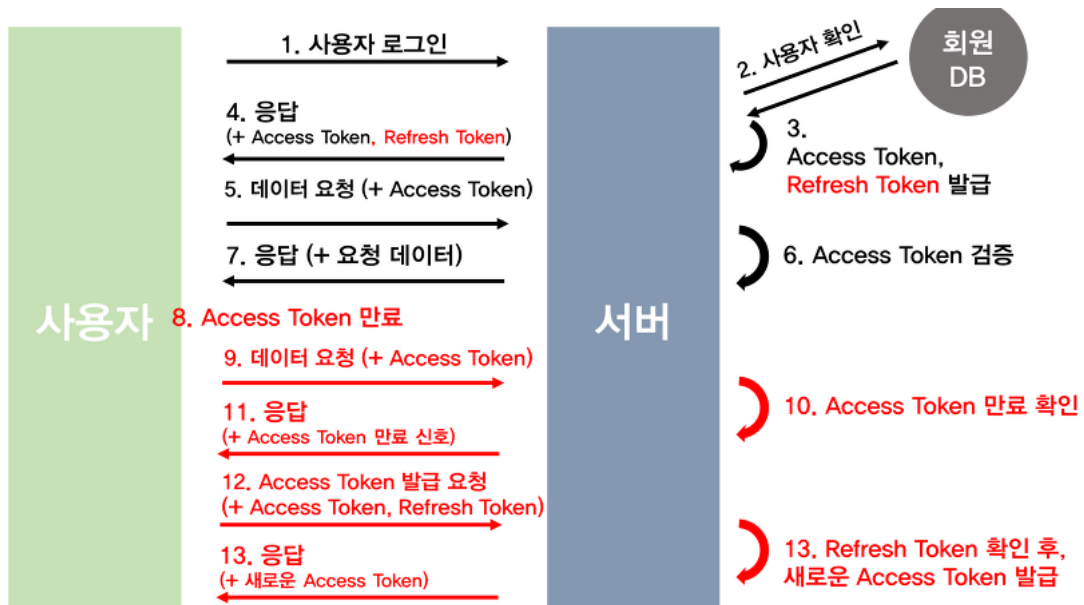
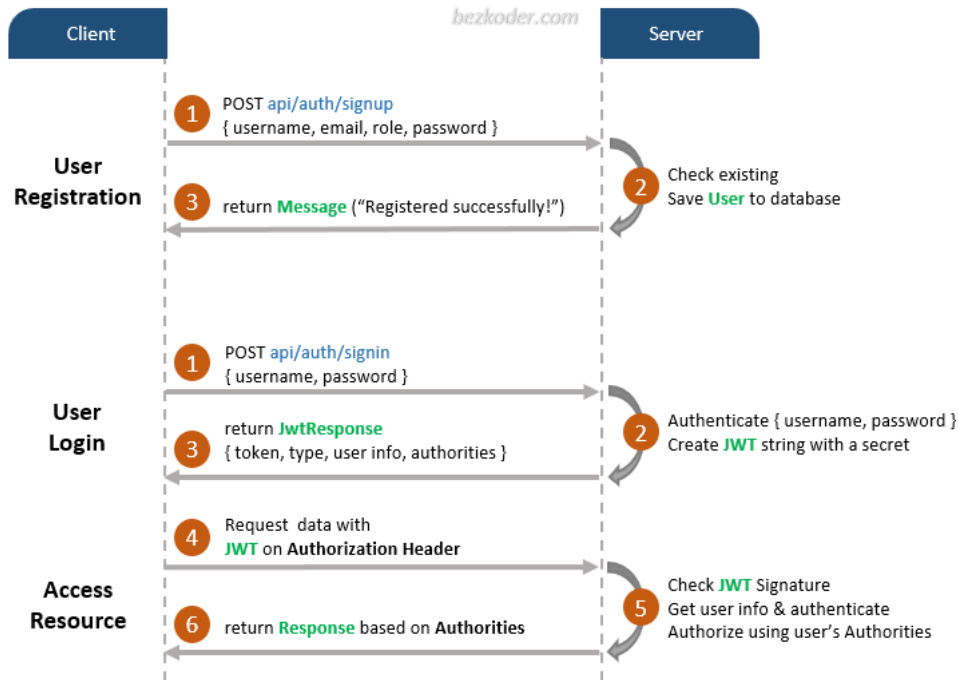
🔗 <https://pronist.dev/146>



- 주요 서버 플로우 참고1

시스템 아키텍처





Spring Security + JWT + React - 01. 백엔드

Spring Security에서 원래 구현하려 했던 것은 Thymeleaf 방식이었으나, 이것은 현실적으로 서비스를 구현할 때 재사용성도 크게 없고, 실무에 쓰일 가능성도 현저히 적다고 생각하여, React로 간단한 프론트엔드를 생성한 후 통신을 주고받는 방식으로 설정하였다. DB와 그것을 연결하는 Entity는 회원을 나타내는 Member 하나로 구현하였다.
<https://velog.io/@juno0713/Spring-Security-JWT-React-w3wpg5yi>

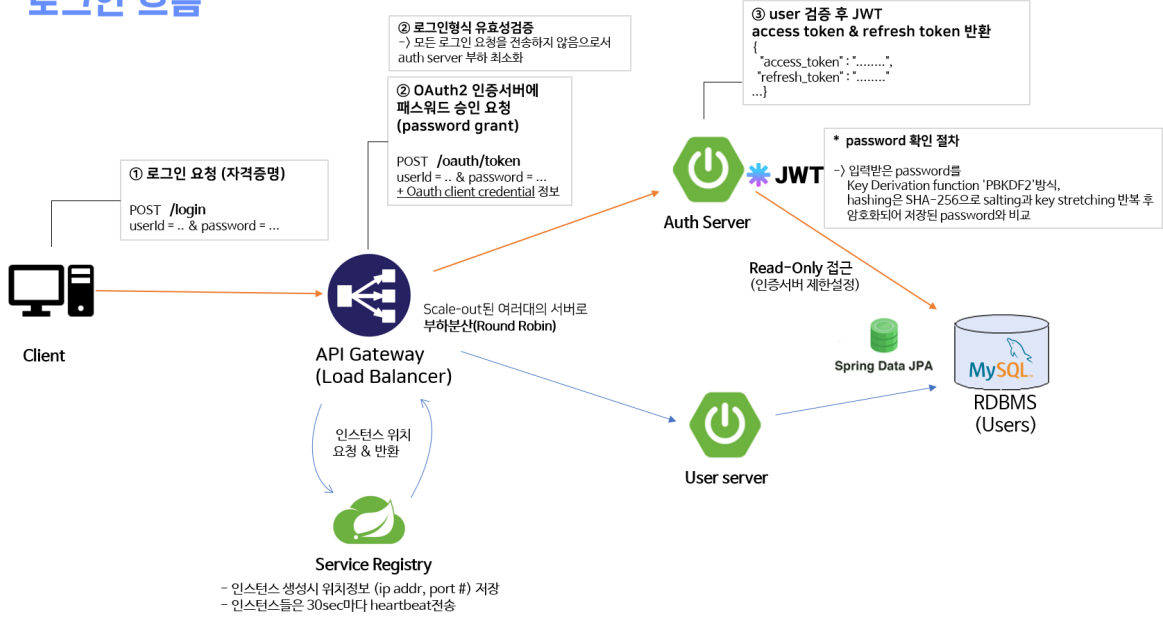


◦ 이메일인증 관련 플로우 추가해야함

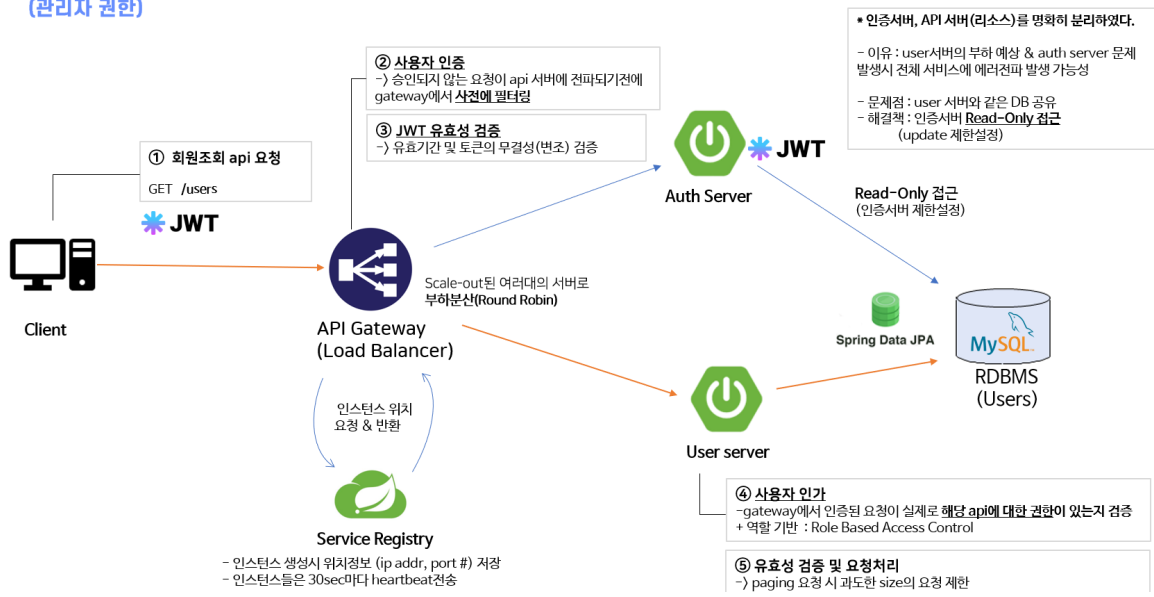
• 주요 서버 플로우 참고2

[GitHub - hscom96/msa-auth: 스마일게이트 데브캠프 개인 프로젝트 #2 '인증서버'](#)

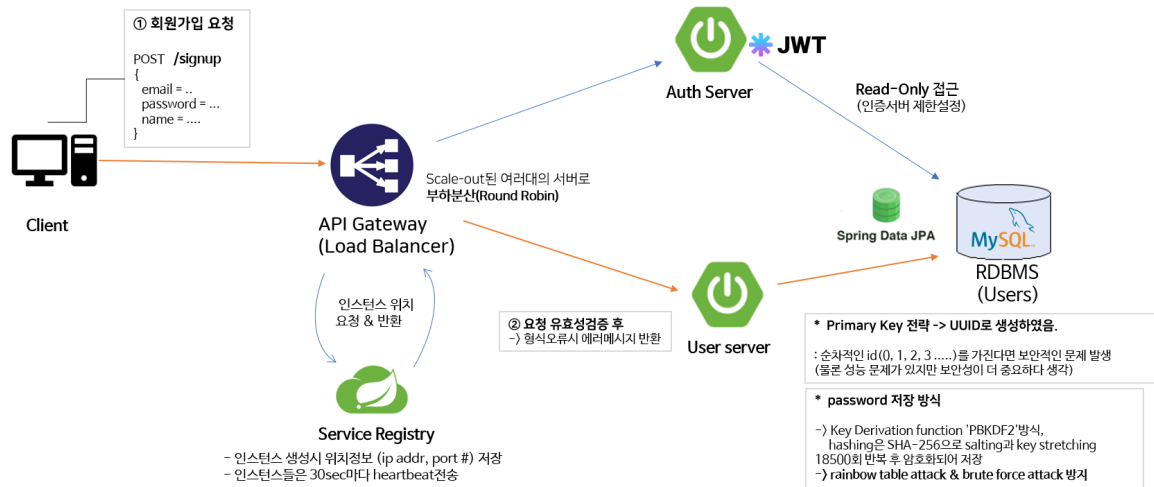
로그인 흐름



사용자 조회 (관리자 권한)

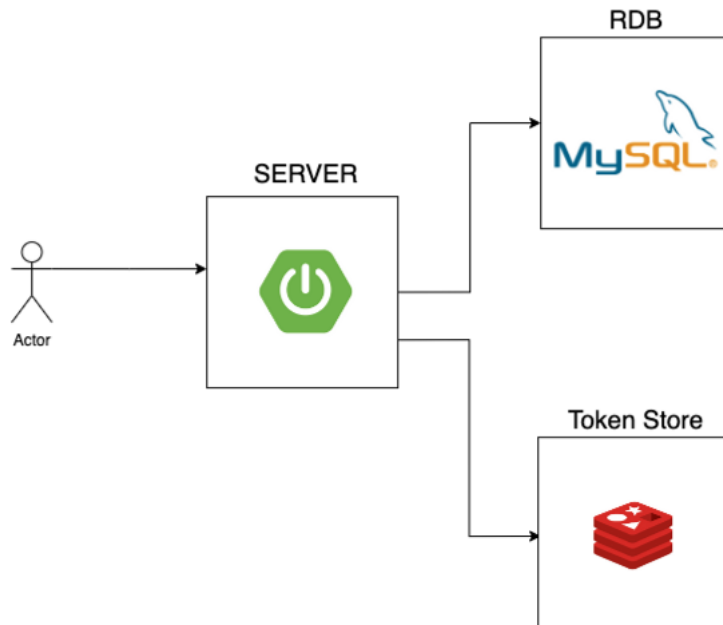


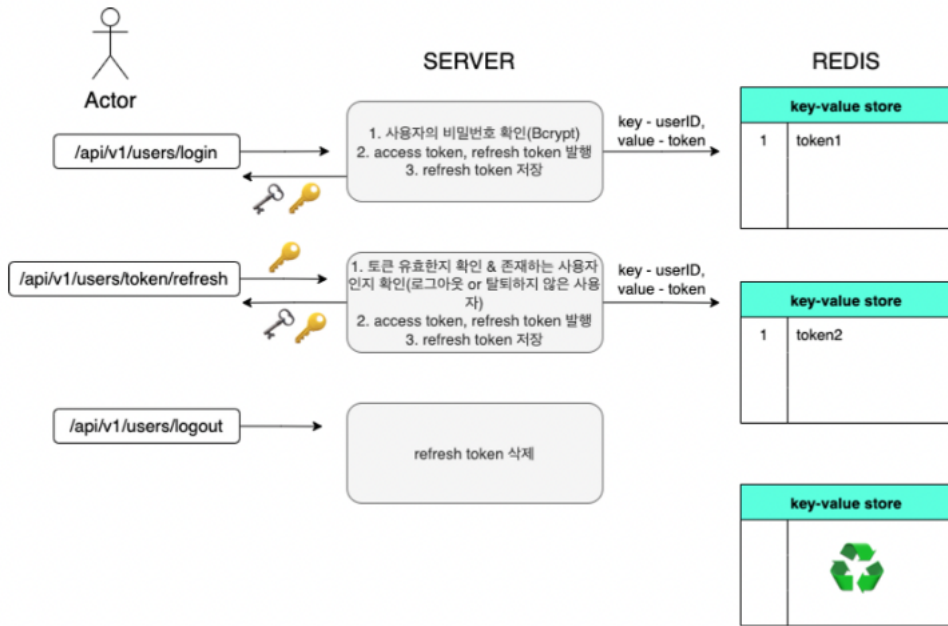
회원가입 흐름



- 주요 서버 플로우 참고3

[GitHub - gkdud583/auth-server-smilgate](https://github.com/gkdud583/auth-server-smilgate) 인증 시스템 구현 리포지토리





- 사용자는 이메일과 비밀번호 정보를 이용하여 회원가입 후 로그인한다. 서버는 정보 확인 후 access token(인증용, 유효시간 30분), refresh token(access token 재발급용, 유효시간 2시간, 1회 사용 가능)을 응답으로 보내준다.
- 사용자는 refresh token이 유효하다면 refresh token를 이용해서 access token을 재발행 할 수 있다. refresh token은 일회용이기 때문에 재발행 시 access token, refresh token을 새롭게 생성하여 응답으로 보내준다.
- redis 데이터는 만료시간을 정해 일정 시간 후에 삭제될 수 있도록 한다.

redis를 사용한 이유:

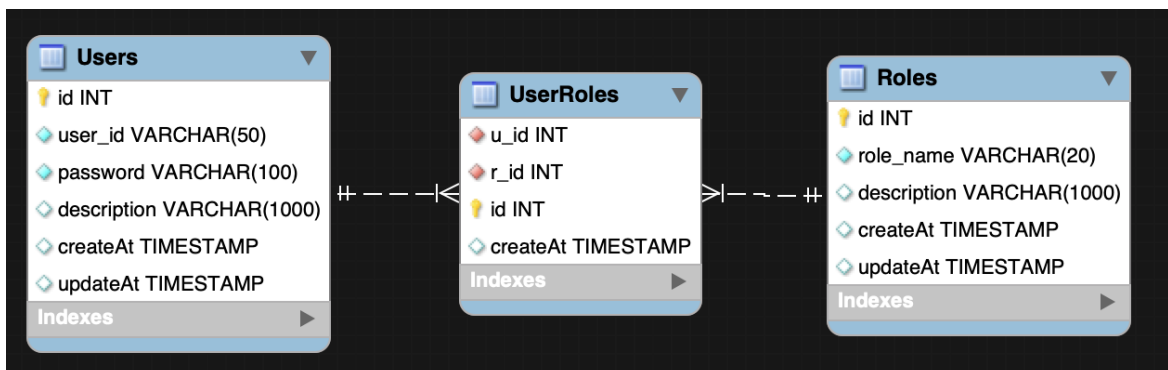
- inmemory database: rdb에 비해 접근 속도가 빠름
- 다양한 자료구조 지원
- 백업 방법 지원
- 싱글 스레드로 동작: 동시성 문제 해결

refresh token 일회용으로 처리한 이유:

refresh token도 탈취될 가능성이 있기 때문에 한 번만 사용할 수 있도록 처리

- DB 설계

[인증서버 구축기 - 2 - lahuman](#)



회원 가입 및 로그인을 위한 테이블 설계 - RastaLion's IT Blog

개발자의 글쓰기라는 책을 읽다보면 이런 에피소드가 나옵니다. 문장을 주고 특정 핵심 단어로 문장을 요약해 보라고 했을 때 DBA는 추려낸 단어에서 중복된 단어는 제거해야 한다고 말한다고 하죠. 이처럼 습관적으로 데이터 중복에 민감한을 표출하는 사람들이 DBA입니다. 많은 데이터베이스 이론들이 정규화의 필요성을 강조합니다.

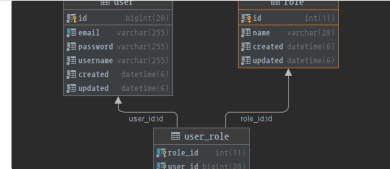
📄 <https://rastalion.me/%ED%9A%8C%EC%9B%90-%EA%B0%80%EC%9E%85-%EB%B0%8F-%EB%A1%9C%EA%B7%B8%EC%9D%B8%EC%9D%84-%EC%9C%84%ED%95%9C-%ED%85%8C%EC%9D%B4%EB%B8%94-%EC%84%A4%EA%B3%84/>



angular, react, vue 등의 최신프론트엔드로 풀스택 개발

백엔드는 spring boot, Node로 프론트엔드는 react, angular, vue 의 최신프론트엔드를 사용하여 풀스택사이트 개발!

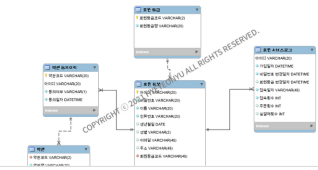
📄 https://eastflag.co.kr/fullstack/rest-with-nodejs/react-rest_auth_entity/



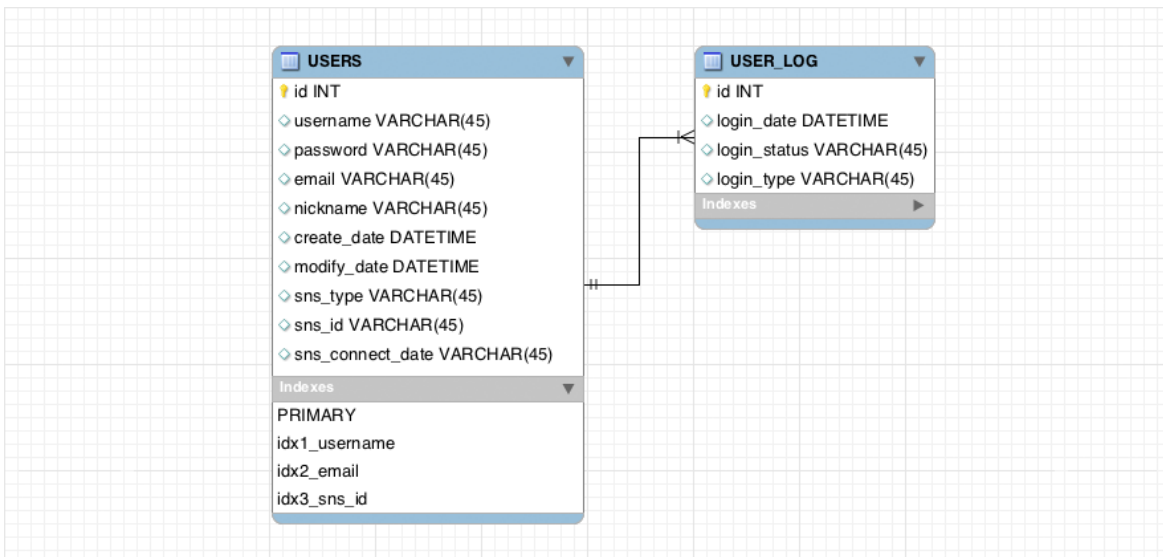
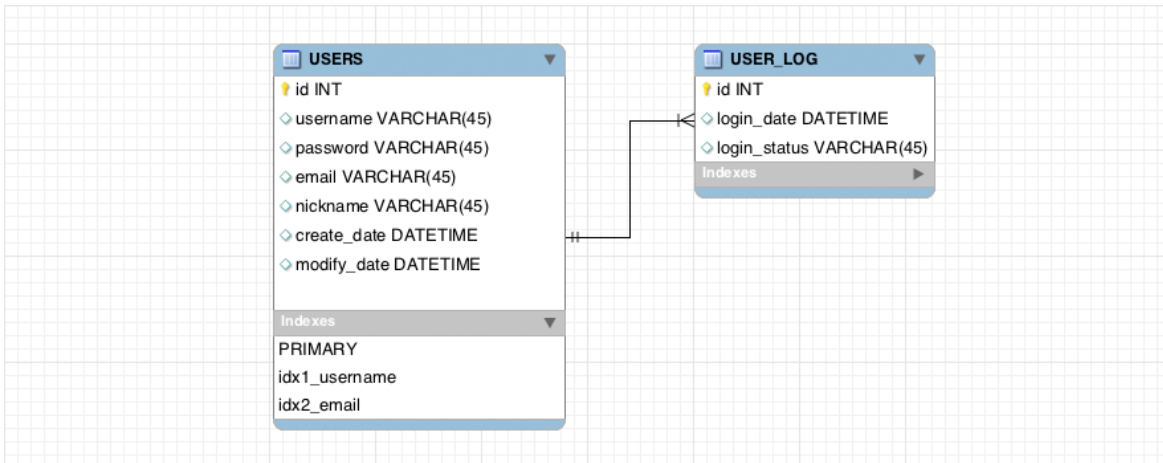
01. 회원 데이터 설계

1단계: 우선 어드민의 회원 관리 화면에서 회원 관련 데이터는 무엇이 있는지 나열해보았다. - 회원정보 조회는 검색과 결과로 구분된다. 검색 시에는 어떤 항목을 기준으로 검색하는지, 결과에는 어떤 데이터가 보여지는지 구분했다. - 설명: 해당 항목의 셀렉트 박스나 라디오 버튼을 보면 해당 항목에 어떤 데이터가 쌓이는지 짐작할 수 있

📄 <https://hyeyun133.tistory.com/entry/01-%ED%9A%8C%EC%9B%90%EA%B4%80%EB%A6%AC-%EB%8D%B0%EC%9D%B4%ED%84%B0>



<https://developers.naver.com/docs/login/devguide/devguide.md>



<https://pjh3749.tistory.com/162>

회원		회원약관		공동코드디테일	
회원 ID	N/A VARCHAR2(10)	약관항목코드	N/A VARCHAR2(2)	코드그룹ID	N/A VARCHAR2(30)
비밀번호	N/A VARCHAR2(10)	약관내용	N/A VARCHAR2(1000)	코드명	N/A VARCHAR2(100)
회원구분코드	N/A VARCHAR2(2)	적용일자	N/A DATE	조회순서	N/A NUMBER(3)
회원이름	N/A VARCHAR2(50)	필수여부	N/A VARCHAR2(1)	부연설명	N/A VARCHAR2(1000)
성별	N/A VARCHAR2(1)	약관적용구분코드	N/A VARCHAR2(2)	등록일시	N/A DATE
생년월일	N/A DATE	조회순서	N/A NUMBER(3)	등록자	N/A VARCHAR2(6)
이메일	N/A VARCHAR2(50)	사용여부	N/A VARCHAR2(1)	수정일시	N/A DATE
국가코드	N/A VARCHAR2(4)	등록일시	N/A DATE	수정자	N/A VARCHAR2(6)
휴대전화번호	N/A VARCHAR2(20)	등록자	N/A VARCHAR2(6)		
단체대표자명	N/A VARCHAR2(50)	수정일시	N/A DATE		
단체종류코드	N/A VARCHAR2(2)	수정자	N/A VARCHAR2(6)		
가입일시	N/A DATE				
탈퇴여부	N/A VARCHAR2(1)				
탈퇴사유	N/A VARCHAR2(300)				
탈퇴일시	N/A DATE				
수정일시	N/A DATE				

공동코드마스터		약관동의이력	
코드그룹ID	N/A VARCHAR2(30)	약관항목코드	N/A VARCHAR2(2)
코드그룹명	N/A VARCHAR2(300)	회원 ID	N/A VARCHAR2(10)
코드길이	N/A NUMBER(3)	동의여부	N/A VARCHAR2(1)
조회순서	N/A NUMBER(3)	동의일시	N/A DATE
등록일시	N/A DATE		

<https://catsbi.oopy.io/75127c5e-fe22-41b6-9c71-cf93529e7f09>

- 클래스 다이어그램(UML)

<https://github.com/devAon/SpringSecurity-JWT>

[Spring Security] JWT(JSON Web Token) 의 코드 구현

🌟 깃허브 : 전체 코드 Spring Boot에 Spring Security와 JWT를 사용해 로그인 구현 1. 동작과정 Generating
 JWT Client : 로그인 요청 POST (id, pw) Server : id, pw가 맞는지 확인 후 맞다면 JWT를 SecretKey로 생성
 Client : Server에게 받은 JWT를 로컬 또는 세션에 저장 Server : 서버에 요청할 때 항상 헤더에 Token을 포함시킴
 🔗 <https://aonee.tistory.com/72>

