프론트에서 서버로 뜸듬히 요청을 보내면
세션 타임 아웃에 걸린다.

또한 서버가 여러대를 운용하여 로드밸런싱을하면
부하를 분산시키고 세션을 메모리기반에 저장하면
한 분산 컴의 메모리에 세션에 정보가 저장됨
하지만 다시
다른 요청을 통해서 다른 컴퓨터로
접속되면 메모리에 세션에 정보가 없기에

문제가 될 수있다.

세션을 DB에 저장하거나 혹은
분산된 컴들의 메모리들을 동기화하거나.

Api Server Security 기본 설정

요즘엔 MSA에서도 세션 유지가 힘들다.

정리 프론트엔드 기술로 서버에 요청을 뜸뜸히 보내면

세션 타임 아웃에 걸리는 문제가 있다.

세션 기반의 정보를 유지하기 힘들다.

=>

해결법. JWT jason web token. json문자열로 정보를 구성한 것.

Api Security 기본 설정

프로젝트

- secserver를 apiserver로 복사
  - settings.gradle에서 프로젝트명 apiserver로 변경

Api Security 기본 설정

Api 서버를 위한 기본 security 설정

- 완료 부분
  - PasswordEncorder 빈 등록
  - UserDetailsService 빈 등록
  - 한글 인코딩 필터 설정

- 추가할 부분
  - cors(Cross Origin Resource Sharing) 허용
  - csrf 기능 비활성화
  - formLogin 기능 비활성화

폼기반 로그인을 사용하지 않을 꺼니까

  - session의 생성 모드를 stateless 모드로 설정

세션에 상태정보를 저장하지 않게금

  - AuthenticationManager 빈 등록

Api Security 기본 설정

security.config.SecurityConfig.java

```
package org.scoula.security.config;
…
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
…

@Configuration
@EnableWebSecurity
@Log4j2
@MapperScan(basePackages  = {"org.scoula.security.account.mapper"})
@ComponentScan(basePackages  = {"org.scoula.security"})
@RequiredArgsConstructor
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    private final UserDetailsService userDetailsService;
```

Api Security 기본 설정

security.config.SecurityConfig.java

```
// 문자셋 필터
public CharacterEncodingFilter encodingFilter() {
    CharacterEncodingFilter encodingFilter = new CharacterEncodingFilter();
    encodingFilter.setEncoding("UTF-8");
    encodingFilter.setForceEncoding(true);
    return encodingFilter;
}

@Bean
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

// AuthenticationManager 빈 등록
@Bean
public AuthenticationManager authenticationManager() throws Exception {
    return super.authenticationManager();
}
```

Api Security 기본 설정

security.config.SecurityConfig.java

```
// cross origin 접근 허용
@Bean
public CorsFilter corsFilter() {
    UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
    CorsConfiguration config = new CorsConfiguration();
    config.setAllowCredentials(true);
    config.addAllowedOriginPattern("*");
    config.addAllowedHeader("*");
    config.addAllowedMethod("*");
    source.registerCorsConfiguration("/**", config);
    return new CorsFilter(source);
}

// 접근 제한 무시 경로 설정 – resource
@Override
public void configure(WebSecurity web) throws Exception {
    web.ignoring().antMatchers("/assets/**", "/*", "/api/member/**");
}
```

모든 오리진과 헤더에 대해서 허용하겠다

다음 경로에서는 security 관여 말라 보안체크 하지말라

회원가입, 조회처럼 로그인없이 접근가능해야 하는 페이지 경로들

루트레벨의 경로

vue에서 static경로

프로젝트마다 다르게 설정해야하는,,,

Api Security 기본 설정

security.config.SecurityConfig.java

```
@Override
public void configure(HttpSecurity http) throws Exception {
  // 한글 인코딩 필터 설정
  http.addFilterBefore(encodingFilter(), CsrfFilter.class);

  http.httpBasic().disable()                    // 기본 HTTP 인증 비활성화
    .csrf().disable()     // CSRF 비활성화
    .formLogin().disable()  // formLogin 비활성화 □ 관련 필터 해제
    .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS); // 세션 생성 모드 설정
}

// Authentication Manger 구성
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
  auth
    .userDetailsService(userDetailsService)
    .passwordEncoder(passwordEncoder());
}
}
```

세션에 상태정보를 저장하지 않게금 설정