

---

# 심,

2021.01.10

---

SIST 2조				
김보라	김태민	박여은	안수현	최재현



숙소 예약 서비스를 통해  
편안한 휴식을 제공합니다.

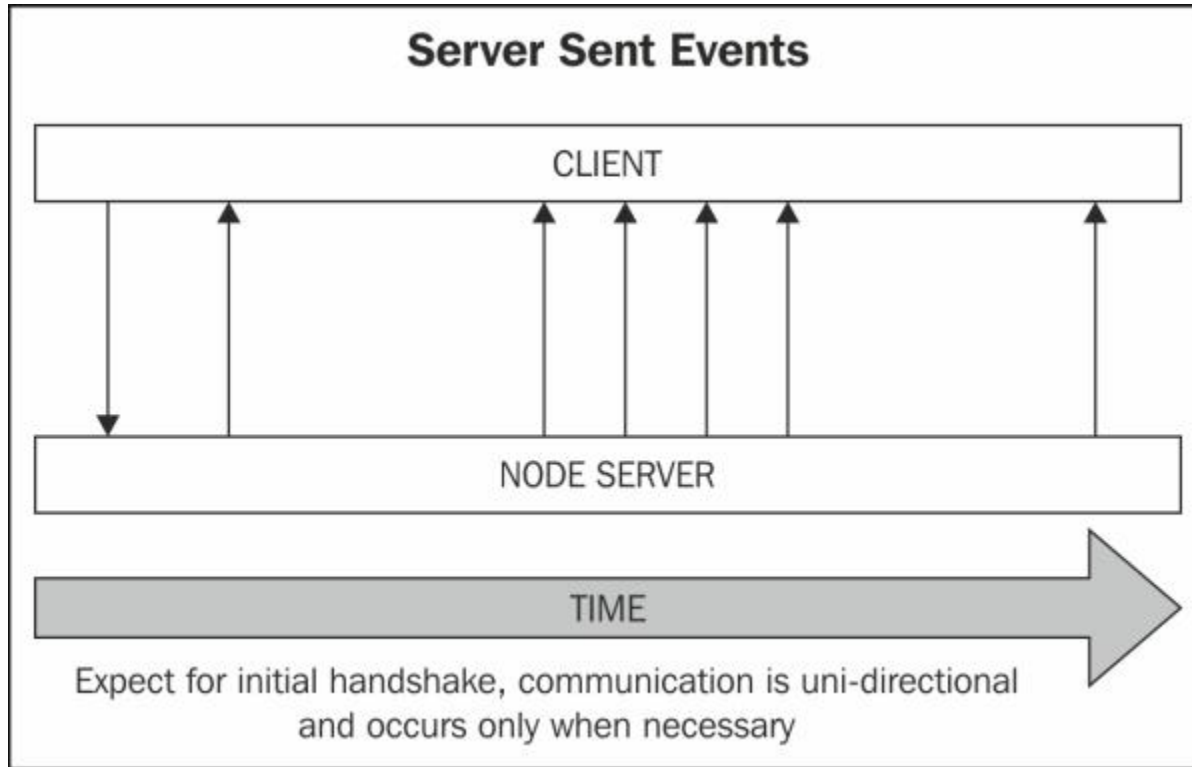


숨,  
시연



심,  
주요 코드 [채팅]

## SSE 통신



```
//채팅방 요청
var eventSourceRoom = new EventSource("http://localhost:8080/chat/chatting/${sessionScope.myid}");

eventSourceRoom.onmessage = function(event) {
    var dataRooms = JSON.parse(event.data);

    createRooms(dataRooms); //dataRooms.msg로 찾을 수 있음
};
```

---

```
SseEmitter emitterRoom;

//채팅방 목록
@GetMapping(path="/chatting/{sender}", produces = MediaType.TEXT_EVENT_STREAM_VALUE)
public @ResponseBody ResponseBodyEmitter getChattingRooms(@PathVariable String sender){
    if(emitterRoom == null) {
        emitterRoom = new SseEmitter();
    }
    chatService.roomAdd(emitterRoom, sender);

    return emitterRoom;
}
```

```
Map<ResponseBodyEmitter, String> emitterRoomMap = new HashMap<>();

//채팅방 MapEmitter
public void roomAdd(ResponseBodyEmitter emitter, String sender) {
    emitterRoomMap.put(emitter, sender);
}
```

```
ResponseBodyEmitter emitterRoom, emitterChat;
//room콜링
@Scheduled(fixedRate = 1000L)
public void emitRoom() {

    for(Map.Entry<ResponseBodyEmitter, String> entry : emitterRoomMap.entrySet()) {

        emitterRoom = entry.getKey();

        try {
            for(ChatDto chatDto : chatMapper.getChattingRooms(entry.getValue())) {
                if(historyRoom.contains(chatDto)) {
                    //채팅방이 있으면 채팅방 전송X
                }else {
                    채팅방이 없으면 채팅방 전송O
                    historyRoom.add(chatDto);
                    emitterRoom.send(chatDto);
                }
            }
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```



심,  
주요 코드 [암호화]



```
@Configuration
public class WebSecurityConfig extends WebSecurityConfigurerAdapter{

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

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.cors().disable()
            .csrf().disable()
            .formLogin().disable()
            .headers().frameOptions().disable();
    }
}
```

```
@Autowired
PasswordEncoder passwordEncoder;
```

```
//패스워드 암호화
dto.setPass(passwordEncoder.encode(dto.getPass()));
```

```
if (passwordEncoder.matches(pass, check.getPass())) {
```

id	pass
rose	\$2a\$10\$xo2BC.CVVzWBvNCD1hhWexHfLiRyFgFJKB614XcVgEmzi4cVx8kq
bora31425@gmail.com	\$2a\$10\$rTbArw9gCBcL5Fp6SYiuPeNswZDcP8U/lmr3jSBZkc1bLB3nDd9yy
jenny	\$2a\$10\$obEg0yVX1TJdLsqDrz4F.CLlxHftU5xc/Png3BQ6nlNNEtnMJtC
taemin	\$2a\$10\$kEQ7I8SAhYc2s7pYpxMAae6LLuvdTgmf/2GTCzDFOpDr1PEKIr7W
admin	\$2a\$10\$fvvGVc4aLD.KBP.oUugfMuRZpFZ6LolsxTSSdmoBDeUZ/fnj7tMYC
stay	\$2a\$10\$Zx8kInMgALEc5nvygtUsYOR3JOc7O13KxRqq0PRUBgmBrWTym9USi
dlwlrma	\$2a\$10\$WwTftk2JZo5EZWbv0mdrnOu9rANaw6.n8rMJyc.zw71MmF6EZSAGS
apple	\$2a\$10\$QQn7hnDlzbJioQNhRAHneXbAxpV2fBwh1216V25IL3vPJNMSDE2
bobo	\$2a\$10\$OxPSg..EaddqLxCF.QwfMOUPrASI7H1Razkx3oInAdzgfQGgZhPZ.



심,  
주요 코드 [카카오 결제]

```
try {
    URL address = new URL("https://kapi.kakao.com/v1/payment/ready");
    HttpURLConnection connection = (HttpURLConnection) address.openConnection(); // 서버연결
    connection.setRequestMethod("POST");
    connection.setRequestProperty("Authorization", "KakaoAK 5e26b9346af700eb3b3a3c9286aa7c87"); // 어드민 키
    connection.setRequestProperty("Content-type", "application/x-www-form-urlencoded;charset=utf-8");
    connection.setDoOutput(true); // 서버한테 전달할게 있는지 없는지

    String parameter = "cid=TC00NETIME" // 가맹점 코드
        + "&partner_order_id=partner_order_id" // 가맹점 주문번호
        + "&partner_user_id=" + myid // 가맹점 회원 id
        + "&item_name=Swim," // 상품명
        + "&quantity=1" // 상품 수량
        + "&total_amount=" + allPrice // 총 금액
        + "&vat_amount=" + taxPrice // 부가세
        + "&tax_free_amount=0" // 상품 비과세 금액
        + "&approval_url=http://localhost:8080/reser/reservationlist" // 결제 성공 시
        + "&fail_url=http://localhost:8080/" // 결제 실패 시
        + "&cancel_url=http://localhost:8080/"; // 결제 취소 시

    OutputStream send = connection.getOutputStream(); // 이제 뭔가를 줄 수 있다.
    DataOutputStream dataSend = new DataOutputStream(send); // 이제 데이터를 줄 수 있다.
    dataSend.writeBytes(parameter); // OutputStream은 데이터를 바이트 형식으로 주고 받기로 약속되어 있다. (형변환)
    dataSend.close(); // flush가 자동으로 호출이 되고 닫는다. (보내고 비우고 닫다)

    int result = connection.getResponseCode(); // 전송 잘 됐나 안됐나 번호를 받는다.
    InputStream receive; // 받다

    if (result == 200) {
        receive = connection.getInputStream();

        reservationService.insertReservation(reserDto);
    } else {
        receive = connection.getErrorStream();
    }
}
```



심,  
주요 코드 [카카오 로그인]

```
<!-- 카카오톡 로그인 -->
<c:if test="${sessionScope.kakaologin eq null}">
    <a href="https://kauth.kakao.com/oauth/authorize?client_id:
        
    </a>
</c:if>
/div>
```

```
public String kakaoLogin(@RequestParam("code") String code, HttpSession session) throws ParseException {
    String access_Token = kakao.getAccessToken(code);
```

```
    HashMap<String, Object> userInfo = kakao.getUserInfo(access_Token);
```

```
    JsonObject properties = element.getAsJsonObject().get("properties").getAsJsonObject();
    JsonObject kakao_account = element.getAsJsonObject().get("kakao_account").getAsJsonObject();
```

```
    String nickname = properties.getAsJsonObject().get("nickname").getString();
    String profile_image = properties.getAsJsonObject().get("profile_image").getString();
    String email = kakao_account.getAsJsonObject().get("email").getString();
    String birthday = kakao_account.getAsJsonObject().get("birthday").getString();
```

```
    userInfo.put("nickname", nickname);
    userInfo.put("profile_image", profile_image);
    userInfo.put("email", email);
    userInfo.put("birthday", birthday);
```

```
    MemberDto mdto = new MemberDto();
    mdto.setId(e_mail);
    mdto.setName(name);
    mdto.setBirth(realbirthday);
    mdto.setPhoto(photo);
    mdto.setE_mail(e_mail);
    mapper.insertMember(mdto);
```



심,  
주요 코드 [카카오 맵]

```
<!-- 지도 -->
<div class="map-wrap">
  <div class="map" id="map"></div>
</div>
```

```
// 지도 생성
```

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
      center: new kakao.maps.LatLng(37.498095, 127.027610), // 지도의 중심좌표
      level: 3 // 지도의 확대 레벨
    };

```

```
$('.room-list-wrap').mouseover(function(){
  var divAddr = $(this).attr("id");

```

```
  marker.setMap(null);

```

```
  // 주소-좌표 변환 객체를 생성합니다.

```

```
  var geocoder = new kakao.maps.services.Geocoder();

```

```
  // 주소로 좌표를 검색합니다

```

```
  geocoder.addressSearch(divAddr, function(result, status) {

```

```
    // 정상적으로 검색이 완료됐으면

```

```
    if (status === kakao.maps.services.Status.OK) {
      var coords = new kakao.maps.LatLng(result[0].y, result[0].x);

```

```
      markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize, imageOption),
      markerPosition = new kakao.maps.LatLng(result[0].y, result[0].x);

```

```
      // 결과값으로 받은 위치를 마커로 표시합니다

```

```
      marker = new kakao.maps.Marker({
        map: map,
        position: coords,
        image: markerImage // 마커이미지 설정
      });

```

```
      // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
      map.setCenter(coords);

```

```
    }

```

```
  });

```

```
});
```