

문자 API

누리고 SMS (coolSMS 사용)



비밀번호 찾기 시, 이름, 사번, 휴대폰 번호를 입력한다. → 일치하는 정보가 데이터베이스에 존재하는 경우, 인증번호를 발송한다.

```
package com.springboard.pracitceApi;

import java.security.SecureRandom;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RestController;

import lombok.extern.slf4j.Slf4j;
import net.nurigo.sdk.NurigoApp;
import net.nurigo.sdk.message.model.Message;
import net.nurigo.sdk.message.request.SingleMessageSendingRequest;
import net.nurigo.sdk.message.response.SingleMessageSentResponse;
import net.nurigo.sdk.message.service.DefaultMessageService;

@Slf4j
@RestController // @Controller+@ResponseBody -> Json 형태로 객체 데이터를 반환
public class nurigoMsgTest1 {
    final DefaultMessageService messageService;

    public nurigoMsgTest1() {
        this.messageService = NurigoApp.INSTANCE.initialize("NCS9ZQYDTT6C7I6N", "F0UYYWY6GTA2LLUTUQZBPSRVVUYQSA9K",
            "https://api.coolsms.co.kr");
    }

    @PostMapping("/api/nurigo/send-one")
    public SingleMessageSentResponse sendOne() {
        log.info("맵핑 잘 되나?");
        Message message = new Message();

        // 번호 형태 : 하이픈 없이, 01012345678
        message.setFrom("01012345678"); // 발신번호
        message.setTo("01012345678"); // 수신번호

        String tempPassword = generateCertificationNumber(); // 인증번호 발급
        String sendText = String.format("[AllRounder] 인증번호는 %s 입니다.", tempPassword);
        message.setText(sendText);

        SingleMessageSentResponse response = this.messageService.sendOne(new SingleMessageSendingRequest(message));
        log.info("발송메세지 : {}", sendText);
        log.info("발송메세지 : {}", response);
        return response;
    }

    /**
     * length 길이만큼 대문자, 소문자, 숫자를 포함하여 비밀번호를 생성하는 메소드
     *
     * @param length
     * @return
     */
    private static String generateCertificationNumber() {
```

```

int length = 6; // 6자리 길이로 변경
String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
StringBuilder code = new StringBuilder();
SecureRandom random = new SecureRandom();

for (int i = 0; i < length; i++) {
    int randomIndex = random.nextInt(characters.length());
    char randomChar = characters.charAt(randomIndex);
    code.append(randomChar);
}

log.info("임시비밀번호 : {}", code.toString());
return code.toString();
}
}

```

결과 : 대문자&소문자&숫자 조합의 6자리 인증번호 발송 !