https://www.google.com/settings/security/lesssecureapps에서 보안 수준이 낮은 앱의 액세스 활성화

활성화를 해줘야 메일을 받을 수 있어요!

dependency 추가

build.gradle

```
dependencies {
	implementation 'org.springframework.boot:spring-boot-starter-mail'
}
```

Colored by Color Scripter cs

properties

email.properties

```
mail.smtp.auth=true
mail.smtp.starttls.required=true
mail.smtp.starttls.enable=true
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false
mail.smtp.port=465
mail.smtp.socketFactory.port=465

#admin 구글 계정
AdminMail.id =
AdminMail.password =
```

Colored by Color Scripter cs

수신 메일(IMAP) 서버	imap.gmail.com SSL 필요: 예 포트: 993
발신 메일(SMTP) 서버	smtp.gmail.com SSL 필요: 예 TLS 필요: 예(사용 가능한 경우) 인증 필요: 예 SSL용 포트: 465 TLS/STARTTLS용 포트: 587
전체 이름 또는 표시 이름	이름
계정 이름, 사용자 이름 또는 이메일 주소	전체 이메일 주소
비밀번호	Gmail 비밀번호

EmailConfig

EmailConfig.java

```
import java.util.Properties;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.PropertySource;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.JavaMailSenderImpl;


@Configuration
@PropertySource("classpath:email.properties")
public class EmailConfig {

    @Value("${mail.smtp.port}")
    private int port;
    @Value("${mail.smtp.socketFactory.port}")
    private int socketPort;
    @Value("${mail.smtp.auth}")
    private boolean auth;
    @Value("${mail.smtp.starttls.enable}")
```

cs

```
    private boolean starttls;
    @Value("${mail.smtp.starttls.required}")
    private boolean startlls_required;
    @Value("${mail.smtp.socketFactory.fallback}")
    private boolean fallback;
    @Value("${AdminMail.id}")
    private String id;
    @Value("${AdminMail.password}")
    private String password;

    @Bean
    public JavaMailSender javaMailService() {
        JavaMailSenderImpl javaMailSender = new JavaMailSenderImpl();
        javaMailSender.setHost("smtp.gmail.com");
        javaMailSender.setUsername(id);
        javaMailSender.setPassword(password);
        javaMailSender.setPort(port);
        javaMailSender.setJavaMailProperties(getMailProperties());
        javaMailSender.setDefaultEncoding("UTF-8");
        return javaMailSender;
    }
    private Properties getMailProperties()
    {
        Properties pt = new Properties();
        pt.put("mail.smtp.socketFactory.port", socketPort);
        pt.put("mail.smtp.auth", auth);
        pt.put("mail.smtp.starttls.enable", starttls);
        pt.put("mail.smtp.starttls.required", startlls_required);
        pt.put("mail.smtp.socketFactory.fallback",fallback);
        pt.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        return pt;
    }
}
```

Colored by Color Scripter

EmailService

EmailService.java

```
public interface EmailService {
    String sendSimpleMessage(String to)throws Exception;
}
```

Colored by Color Scripter cs

EmailServiceImpl

EmailServiceImpl.java

cs

```
import java.util.Random;

import javax.mail.Message.RecipientType;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.MailException;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.stereotype.Service;

@Service
public class EmailServiceImpl implements EmailService{

    @Autowired
    JavaMailSender emailSender;

    public static final String ePw = createKey();

    private MimeMessage createMessage(String to)throws Exception{
        System.out.println("보내는 대상 : "+ to);
        System.out.println("인증 번호 : "+ePw);
        MimeMessage  message = emailSender.createMimeMessage();

        message.addRecipients(RecipientType.TO, to);//보내는 대상
        message.setSubject("Babble회원가입 이메일 인증");//제목

        String msgg="";
        msgg+= "<div style='margin:100px;'>";
        msgg+= ;
        msgg+= "<br>";
        msgg+= "<p>아래 코드를 회원가입 창으로 돌아가 입력해주세요<p>";
        msgg+= "<br>";
        msgg+= "<p>감사합니다!<p>";
        msgg+= "<br>";
        msgg+= "<div align='center' style='border:1px solid black; font-family:verdana';>";
        msgg+= "<h3 style='color:blue;'>회원가입 인증 코드입니다.</h3>";
        msgg+= "<div style='font-size:130%'>";
        msgg+= "CODE : <strong>";
        msgg+= ePw+"</strong><div><br/> ";
        msgg+= "</div>";
        message.setText(msgg, "utf-8", "html");//내용
        message.setFrom(new InternetAddress("properties email쓰세용!","Babble"));//보내는 사람

        return message;
    }

    public static String createKey() {
```

```
        StringBuffer key = new StringBuffer();
        Random rnd = new Random();

        for (int i = 0; i < 8; i++) { // 인증코드 8자리
            int index = rnd.nextInt(3); // 0~2 까지 랜덤

            switch (index) {
                case 0:
                    key.append((char) ((int) (rnd.nextInt(26)) + 97));
                    //  a~z  (ex. 1+97=98 => (char)98 = 'b')
                    break;
                case 1:
                    key.append((char) ((int) (rnd.nextInt(26)) + 65));
                    //  A~Z
                    break;
                case 2:
                    key.append((rnd.nextInt(10)));
                    // 0~9
                    break;
            }
        }

        return key.toString();
    }
    @Override
    public String sendSimpleMessage(String to)throws Exception {
        // TODO Auto-generated method stub
        MimeMessage message = createMessage(to);
        try{//예외처리
            emailSender.send(message);
        }catch(MailException es){
            es.printStackTrace();
            throw new IllegalArgumentException();
        }
        return ePw;
     }

}
```

Colored by Color Scripter

★ 저는 인증코드 일치 여부를 비교해주기 위해서 sendSimpleMessage() 메서드를 void가 아닌 String으로 만들어주었습니다!

Controller

Controller.java

```
@PostMapping("/emailConfirm")
    @ApiOperation(value = "회원 가입시 이메인 인증", notes = "기존사용하고 있는 이메일을 통해 인증")
    @ApiResponses({
            @ApiResponse(code = 200, message = "성공"),
            @ApiResponse(code = 401, message = "인증 실패"),
            @ApiResponse(code = 404, message = "사용자 없음"),
            @ApiResponse(code = 500, message = "서버 오류")
    })
    public ResponseEntity<? extends BaseResponseBody> emailConfirm(
            @RequestBody @ApiParam(value="이메일정보 정보", required = true) String email) throws Exception {

        String confirm = emailService.sendSimpleMessage(email);

        return ResponseEntity.status(200).body(BaseResponseBody.of(200, confirm));
    }
```