TEST EVIDENCE – SPRING 44- Sending email from spring boot app

Code changes:

1. Dependency:

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-mail</artifactId>  
</dependency>

1. Properties:

spring.mail.host=smtp.gmail.com  
spring.mail.port=587  
spring.mail.username=snehalwagh2121@gmail.com  
spring.mail.password=<16 digit password>  
spring.mail.properties.mail.smtp.auth=true  
spring.mail.properties.mail.smtp.starttls.enable=true

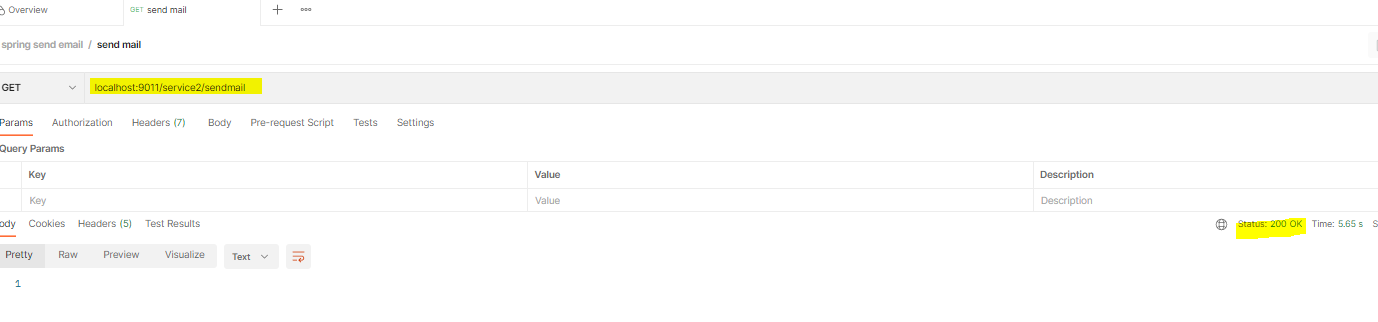
we can set these properties in code too eg:

@Bean  
public JavaMailSender getJavaMailSender() {  
 JavaMailSenderImpl mailSender = new JavaMailSenderImpl();  
 mailSender.setHost("smtp.gmail.com");  
 mailSender.setPort(587);  
  
 mailSender.setUsername("snehalwagh2121@gmail.com");  
 mailSender.setPassword("hello@world1");  
  
 Properties props = mailSender.getJavaMailProperties();  
 props.put("mail.transport.protocol", "smtp");  
 props.put("mail.smtp.auth", "true");  
 props.put("mail.smtp.starttls.enable", "true");  
 props.put("mail.debug", "true");  
  
 return mailSender;  
}

1. Send mail:

@RestController  
@RequestMapping("/service2")  
public class SendEmail {  
  
 @Autowired  
 private JavaMailSender emailSender;  
  
 public void sendSimpleMessage(  
 String to, String subject, String text) {  
  
 SimpleMailMessage message = new SimpleMailMessage();  
 message.setFrom("snehalwagh2121@gmail.com");  
 message.setTo(to);  
 message.setSubject(subject);  
 message.setText(text);  
 emailSender.send(message);  
 }  
  
 @GetMapping("/sendmail")  
 public void emailSender(){  
 sendSimpleMessage("snehalwagh2121@gmail.com", "email sent from spring app", "sample text");  
 }  
  
}

Postman:

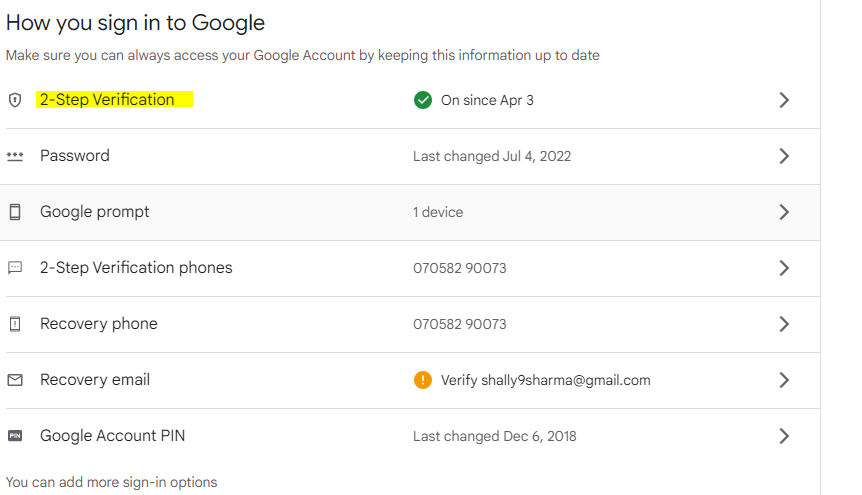


Gmail:

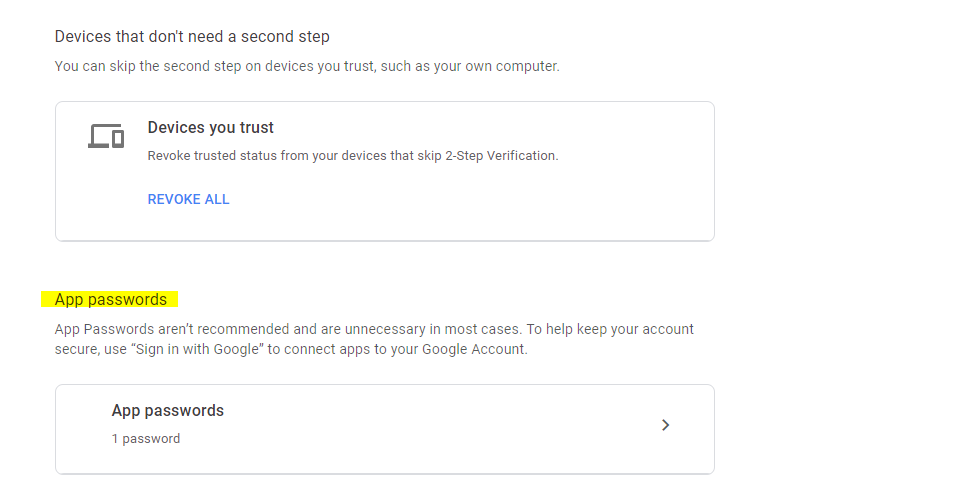


Steps for setting app password:

1. Login to gmail
2. Go to 2-step verification.



1. Turn on 2-step verification.
2. In the bottom there will be an option to add application passwords



1. Add an app as custom app. A password will be generated.

