@Value("${spring.mail.username}")

String username;

@Value("${spring.mail.password}")

String password;

@Value("${isc.homepage}")

String clientHomeUrl;

private String PASSWORD\_RESET\_SUBJECT = "Password reset request";

private String PASSWORD\_RESET\_BODY = "<h1>Một yêu cầu reset password được gửi từ bạn</h1>" +

"<p>Chào bạn, " +

"một ai đó đã yêu cầu một mật khẩu mới với tài khoản của bạn. Nếu đó là hành động của bạn, vui lòng bấm vào link phía dưới để lấy mật khẩu mới." +

"<br/><a href='$clientPage/verification-service/password-reset?token=$tokenValue'>Click vào đây</a><br/><br/>" +

"Cảm ơn.";

@Override

public void sendEmail(Email email) throws MessagingException {

Properties properties = new Properties();

properties.put("mail.smtp.auth", "true");

properties.put("mail.smtp.starttls.enable", "true");

properties.put("mail.smtp.host", "smtp.gmail.com");

properties.put("mail.smtp.port", "587");

properties.setProperty("mail.smtp.allow8bitmime", "true");

properties.setProperty("mail.smtps.allow8bitmime", "true");

Session session = Session.getInstance(properties, new Authenticator() {

@Override

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(username, password);

}

});

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress(username, false));

message.setRecipients(Message.RecipientType.TO, InternetAddress.parse("tamht298@gmail.com"));

message.setSubject(email.getFromAddress() + " - " + email.getSubject());

message.setContent(email.getBody(), "text/html");

message.setSentDate(new Date());

Transport.send(message);

}

@Override

public void resetPassword(String email, String token) throws MessagingException {

Properties properties = new Properties();

properties.put("mail.smtp.auth", "true");

properties.put("mail.smtp.starttls.enable", "true");

properties.put("mail.smtp.host", "smtp.gmail.com");

properties.put("mail.smtp.port", "587");

properties.setProperty("mail.smtp.allow8bitmime", "true");

properties.setProperty("mail.smtps.allow8bitmime", "true");

Session session = Session.getInstance(properties, new Authenticator() {

@Override

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(username, password);

}

});

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress(username, false));

message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(email));

message.setSubject(email + " - " + PASSWORD\_RESET\_SUBJECT);

String htmlBodyWithToken = PASSWORD\_RESET\_BODY.replace("$tokenValue", token).replace("$clientPage", clientHomeUrl);

message.setContent(htmlBodyWithToken, "text/html");

message.setSentDate(new Date());

Transport.send(message);

}

#SMTP

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.username=phong@gmail.com

spring.mail.password=####

# Other properties

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.connectiontimeout=5000

spring.mail.properties.mail.smtp.timeout=5000

spring.mail.properties.mail.smtp.writetimeout=5000

# TLS , port 587

spring.mail.properties.mail.smtp.starttls.enable=true

#SPA

isc.homepage=http://localhost:4200