

Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Eduardo Oliveira Silva
Jônatas de Souza Rezende
Maciele Savier Rodrigues
Lucas Pereira de Azevedo

Recuperação de Senha
Enviar e-mail pelo Java

Outubro
2017

Sumário

1	Procedimento para recuperação	2
2	Código de exemplo	2
3	Explicação do código.	3
4	Referências	3

Recuperação de Senha - Enviar e-mail pelo Java

1 Proccimento para recuperação

Para recuperar a senha geralmente é usado o envio de um e-mail solicitando ao usuário a criação de uma nova senha, nesse e-mail é apresentado um link onde se realmente o usuário solicitou a recuperação da senha deve clicar para que possa ser providenciado a criação e envio de um novo e-mail com a senha provisória.

A forma de enviar e-mail usando o Java se encontra explicada nos capítulos seguintes.

2 Código de exemplo

```
1 import java.util.Properties;
2 import javax.mail.Address;
3 import javax.mail.Message;
4 import javax.mail.MessagingException;
5 import javax.mail.PasswordAuthentication;
6 import javax.mail.Session;
7 import javax.mail.Transport;
8 import javax.mail.internet.InternetAddress;
9 import javax.mail.internet.MimeMessage;
10
11 public class JavaMailApp
12 {
13     public static void main(String[] args) {
14         Properties props = new Properties();
15         props.put("mail.smtp.host", "smtp.gmail.com");
16         props.put("mail.smtp.socketFactory.port", "465");
17         props.put("mail.smtp.socketFactory.class", "javax.net.ssl.
18             SSLSocketFactory");
19         props.put("mail.smtp.auth", "true");
20         props.put("mail.smtp.port", "465");
21
22         Session session = Session.getDefaultInstance(props,
23             new javax.mail.Authenticator() {
24                 protected PasswordAuthentication
25                     getPasswordAuthentication()
26                 {
27                     return new PasswordAuthentication("
28                         testarjavamail@gmail.com", "
29                         tjm123456");
30                 }
31             });
32         session.setDebug(true);
33         try {
34
35             Message message = new MimeMessage(session);
36             message.setFrom(new InternetAddress("testarjavamail@gmail
37                 .com"));
38
39             Address[] toUser = InternetAddress
40                 .parse("testarjavamail@gmail.com");
41             message.setRecipients(Message.RecipientType.TO, toUser);
42             message.setSubject("Enviando email com JavaMail");
43             message.setText("Enviei este email utilizando JavaMail
44                 com minha conta GMail!");
45             Transport.send(message);
46             System.out.println("Feito!!!");
47         } catch (MessagingException e) {
48             throw new RuntimeException(e);
49         }
50     }
51 }
```

3 Explicação do código.

1. Importações necessárias para a execução do código Java:

```
1 import java.util.Properties;
2 import javax.mail.Address;
3 import javax.mail.Message;
4 import javax.mail.MessagingException;
5 import javax.mail.PasswordAuthentication;
6 import javax.mail.Session;
7 import javax.mail.Transport;
8 import javax.mail.internet.InternetAddress;
9 import javax.mail.internet.MimeMessage;
```

2. O objeto **props** define os parâmetros da conexão com o servidor Gmail.

```
1 Properties props = new Properties();
2 props.put("mail.smtp.host", "smtp.gmail.com");
3 props.put("mail.smtp.socketFactory.port", "465");
4 props.put("mail.smtp.socketFactory.class", "javax.net.ssl.
    SSLSocketFactory");
5 props.put("mail.smtp.auth", "true");
6 props.put("mail.smtp.port", "465");
```

3. O objeto **session** inicia a sessão para autenticação do usuário remetente do e-mail

```
1 Session session = Session.getDefaultInstance(props, new javax.mail.
    Authenticator() {
2     protected PasswordAuthentication getPasswordAuthentication()
3     {
4         return new PasswordAuthentication("testarjavamail@gmail.com", "
            tjm123456");
5     }
6 });
```

4. Para facilitar o acompanhamento da execução do código é ativado o debug para a sessão.

```
1 session.setDebug(true);
```

5. O objeto **message** define as informações do e-mail.

```
1 Message message = new MimeMessage(session);
2 message.setFrom(new InternetAddress("testarjavamail@gmail.com"));
3 Address[] toUser = InternetAddress.parse("testarjavamail@gmail.com");
4 message.setRecipients(Message.RecipientType.TO, toUser);
5 message.setSubject("Enviando email com JavaMail");
6 message.setText("Enviei este email utilizando JavaMail com minha conta
    GMail!");
```

6. A classe **FileReader** executa a leitura dos dados do arquivo, cria a variável **reader** e armazena os dados obtidos na variável criada.

```
1 FileReader reader = new FileReader(n_arquivo);
```

7. Método para enviar a mensagem criada.

```
1 Transport.send(message);
```

4 Referências

1. ORACLE. **JavaMail**. Disponível em: <<http://www.oracle.com/technetwork/java/javamail/index.html>>. Acesso em: 23 de outubro de 2017.