

```

1 package in.gov.icegate.custodian.controller;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 /*
19  * @Developer: Abhinav Tyagi
20  *
21  *
22  */
23
24 public class MainClass {
25
26     private static Logger logger = Logger.getLogger(MainClass.class);
27     Connection connection = null;
28     PreparedStatement preparedStatement = null;
29     PreparedStatement preparedStatement2 = null;
30     PreparedStatement preparedStatement3 = null;
31     PreparedStatement preparedStatement4 = null;
32     ResultSet resultSet = null;
33     ResultSet resultSet2 = null;
34     ResultSet resultSet3 = null;
35     ResultSet resultSet4 = null;
36     CustodianBean custodian = null;
37     List<CustodianBean> custodians = null;
38     String locationcd = null;
39     List<String> locationcds = null;
40
41     public static void main(String[] args) throws Exception {
42
43         MainClass mainClass = new MainClass();
44         String sentStatus = mainClass.getData();
45
46         System.out.println("Email sent status: " + sentStatus);
47         logger.info("Email sent status: " + sentStatus);
48     }
49
50     public String getData() throws Exception {
51         try {
52             Properties emailProp = CommonFunction.getMailProperties();
53
54
55             System.out.println("Inside getData method ");
56             logger.info("Inside getData method ");
57             //Class.forName("com.mysql.jdbc.Driver");
58             connection = DBConnection.getConnection();
59             preparedStatement = connection.prepareStatement(" SELECT COUNT(*),(SYSDATE - 1) FROM MEMBER WHERE ROLE_ID = '13' AND trunc(CREATION_DATE) = trunc(SYSDATE - 1)");
60             resultSet = preparedStatement.executeQuery();

```

```
60 resultSet = preparedStatement.executeQuery();
61
62
63 Integer custodianCount = null;
64 Date previousDate = null;
65 while(resultSet.next()){
66     custodianCount = resultSet.getInt(1);
67     previousDate = resultSet.getDate(2);
68 }
69
70 System.out.println("Total number of custodians: " + custodianCount + ", Date: " + previousDate);
71 logger.info("Total number of custodians: " + custodianCount + ", Date: " + previousDate);
72
73
74 if(custodianCount > 0){
75
76     preparedStatement2 = connection.prepareStatement("select ICEGATE_ID, MEMBER_PARENT_NUM, CODE, AGENCY_NM, MEMBER_NUM from MEMBER where ROLE_ID = '13' AND trunc(CREATION_DATE) = trunc(SYSDATE - 1)");
77     resultSet2 = preparedStatement2.executeQuery();
78
79     custodians = new ArrayList<CustodianBean>();
80     while(resultSet2.next()){
81         custodian = new CustodianBean();
82
83         custodian.setIcegateId(resultSet2.getString(1));
84         custodian.setCustodianCode(resultSet2.getString(3));
85         custodian.setAgencyName(resultSet2.getString(4));
86
87
88         Integer memberParentNum = resultSet2.getInt(2);
89         if (memberParentNum != 0){
90             custodian.setCustodianType("Parent");
91             custodian.setParentIcegateId("N/A");
92         }else{
93             custodian.setCustodianType("Child");
94             preparedStatement3 = connection.prepareStatement("select ICEGATE_ID from MEMBER where MEMBER_NUM = ?");
95             preparedStatement3.setInt(1, memberParentNum);
96             resultSet3 = preparedStatement3.executeQuery();
97             while(resultSet3.next()){
98                 custodian.setParentIcegateId(resultSet3.getString(1));
99             }
100         }
101     }
102
103     preparedStatement4 = connection.prepareStatement("select distinct(ICES_LOC_ID) from ICES_LOCATION where ICES_LOC_NUM in (select ICES_LOC_NUM from MEMBER_LOC_REL where MEMBER_NUM = ?)");
104     preparedStatement4.setInt(1, resultSet2.getInt(5));
105     resultSet4 = preparedStatement4.executeQuery();
106     locationCds = new ArrayList<String>();
```

```

97         while(resultSet3.next()){
98             custodian.setParentIcegateId(resultSet3.getString(1));
99         }
100     }
101
102     preparedStatement4 = connection.prepareStatement("select distinct(ICES_LOC_ID) from ICES_LOCATION where ICES_LOC_NUM in (select ICES_LOC_NUM from MEMBER_LOC_REL where MEMBER_NUM = ?)");
103     preparedStatement4.setInt(1, resultSet2.getInt(5));
104     resultSet4 = preparedStatement4.executeQuery();
105     locationcds = new ArrayList<String>();
106     while(resultSet4.next()){
107         locationcd = new String();
108         locationcd = resultSet4.getString(1);
109         locationcds.add(locationcd);
110     }
111     custodian.setLocationCode(locationcds);
112     custodians.add(custodian);
113 }
114
115 return new EmailService().sendEmailToUser(custodianCount, previousDate, emailProp, custodians);
116
117 }
118
119 } catch (Exception e) {
120     e.printStackTrace();
121     return "Error while connecting to the database";
122 } finally {
123     try {
124         if (resultSet != null) {
125             resultSet.close();
126         }
127         if (preparedStatement != null) {
128             preparedStatement.close();
129         }
130         if (connection != null) {
131             connection.close();
132         }
133     }
134 } catch (Exception e) {
135     e.printStackTrace();
136     throw new Exception("DB Exception while DB operation");
137 }
138 }
139 }
140
141 }
142 }
143

```



```
EmailService.java MainClass.java
1 package in.gov.icegate.custodian.service;
2
30 import java.util.Date;
20
21 /**
22  * @Developer: Abhinav Tyagi
23  *
24  *
25  */
26
27 public class EmailService {
28     private static Logger logger = Logger.getLogger(EmailService.class);
29
30
31     public String sendEmailToUser(Integer custodianCount, Date previousDate, Properties emailpro, List<CustodianBean> custodians) throws Exception {
32         final String userName = emailpro.getProperty("mail.from");
33         String toEmailId = emailpro.getProperty("mail.to"); //email to
34         String ccEmailId = emailpro.getProperty("mail.cc");
35         String emailSubject = emailpro.getProperty("mail.subject");
36         Session session = Session.getInstance(emailpro, null);
37         try {
38
39             System.out.println("Inside sendEmailToUser method");
40             logger.info("Inside sendEmailToUser method");
41
42             MimeMessage msg = new MimeMessage(session);
43             msg.setFrom(new InternetAddress(userName));
44             InternetAddress[] address = InternetAddress.parse(toEmailId);
45             msg.setRecipients(Message.RecipientType.TO, address);
46             InternetAddress[] addressCC = InternetAddress.parse(ccEmailId);
47             msg.setRecipients(Message.RecipientType.CC, addressCC);
48
49
50
51             msg.setSubject(emailSubject);
52             MimeBodyPart mbp1 = new MimeBodyPart();
53             StringBuilder sb = new StringBuilder();
54
55             sb.append("Dear Team," + "\n\nAs on Date: " + previousDate + ", Registered Custodians at ICEGATE Portal: " + custodianCount +
56                 "\n\nCustodian Details:");
57
58             if(custodianCount > 0){
59                 for(CustodianBean custodian : custodians){
60                     sb.append( "\n\n1. ICEGATE_ID: " + custodian.getIcegateId() +
61                         "\n2. Location Code: " + custodian.getLocationCode() +
62                         "\n3. Parent/Child: " + custodian.getCustodianType() +
63
```

Writablr

Smart Ins


```

53 MimeBodyPart mbp1 = new MimeBodyPart();
54 StringBuilder sb = new StringBuilder();
55
56 sb.append("Dear Team," + "\n\nAs on Date: " + previousDate + ", Registered Custodians at ICEGATE Portal: " + custodianCount +
57 "\n\nCustodian Details:");
58
59 if(custodianCount > 0){
60     for(CustodianBean custodian : custodians){
61         sb.append( "\n\n1. ICEGATE_ID: " + custodian.getIcegateId() +
62 "\n2. Location Code: " + custodian.getLocationCode() +
63 "\n3. Parent/Child: " + custodian.getCustodianType() +
64 "\n4. Custodian Code: " + custodian.getCustodianCode() +
65 "\n5. Agency Name: " + custodian.getAgencyName() +
66 "\n6. PARENT ICEGATE ID: " + custodian.getParentIcegateId() +
67 "\n-----");
68     }
69
70 }else if(custodianCount == 0){
71     sb.append("\n\nCustodian details are not available.");
72 }
73
74 sb.append("\n\nThanks," + "\n\nICEGATE L3 JAVA TEAM");
75 mbp1.setText(sb.toString());
76
77 Multipart mp = new MimeMultipart();
78 mp.addBodyPart(mbp1);
79 msg.setContent(mp);
80 msg.setSentDate(new Date());
81
82 System.out.println("Before sending mail To: " + toEmailId);
83 logger.info("Before sending mail To: " + toEmailId);
84
85 Transport.send(msg);
86
87 System.out.println("Email sent sucessfully To: " + toEmailId);
88 logger.info("Email sent sucessfully To: " + toEmailId);
89 return "OK";
90
91 } catch (Exception e) {
92     System.out.println(e.getMessage());
93     // logger.info(e);
94     e.printStackTrace();
95     return "Error while sending email";
96 }
97 }
98 }
99

```

Writable

Smart Insert