Mobile number validation

<input id="txtPhoneNo" type="text" onkeypress="return isNumber(event)" />

<input type="button" value="Submit" onclick="ValidateNo();">

<script type="text/javascript">

function isNumber(evt) {

evt = (evt) ? evt : window.event;

var charCode = (evt.which) ? evt.which : evt.keyCode;

if (charCode > 31 && (charCode < 48 || charCode > 57)) {

alert("Please enter only Numbers.");

return false;

}

return true;

}

function ValidateNo() {

var phoneNo = document.getElementById('txtPhoneNo');

if (phoneNo.value == "" || phoneNo.value == null) {

alert("Please enter your Mobile No.");

return false;

}

if (phoneNo.value.length < 10 || phoneNo.value.length > 10) {

alert("Mobile No. is not valid, Please Enter 10 Digit Mobile No.");

return false;

}

alert("Success ");

return true;

}

</script>

function IsMobileNumber(txtMobId) {

var mob = /^[1-9]{1}[0-9]{9}$/;

var txtMobile = document.getElementById(txtMobId);

if (mob.test(txtMobile.value) == false) {

alert("Please enter valid mobile number.");

txtMobile.focus();

return false;

}

return true;

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Mobile No. Validation</title>

<script type="text/javascript">

function validateMobNo(mobno){

var mobno2;

var flag=false;

var mlen= mobno.length;

//alert(mobno.substr(3,mobno.length-3));

if(mobno.charAt(0)!='+' && mlen==10){

mobno2="+91-"+mobno;

alert("1>>your mobile wants to be in "+mobno2);

flag=true;

}

else if(mobno.charAt(0)=='+'){

if(mobno.substr(0,3)=='+91' && mobno.length==13){

mobno2=mobno.substr(0,3)+"-"+mobno.substr(3,mobno.length-3);

alert("2>>your mobile wants to be in "+mobno2);

flag=true;

}

}

else if(mobno.indexOf("-")<0&&mobno.length==12 && mobno.substr(0,2)=='91'){

mobno2=mobno.substr(0,2)+"-"+mobno.substr(3,mobno.length-2);

alert("3>>your mobile wants to be in "+mobno2);

flag=true;

}

else

alert("Please correct your mobile No");

if(flag==true)

document.mobvalidate.mobno.value=mobno2;

else

document.mobvalidate.mobno.focus()

return flag;

}

</script>

</head>

<body>

<form name="mobvalidate">

<input type="text" id="mobno" />

<input type="button" value="VALIDATE" onclick="validateMobNo(mobno.value)" />

</form>

</body>

</html>

receive sms

<http://www.ozekisms.com/video/start_application_with_sms_part1/start_application_with_sms_part1.html>

<http://www.ozekisms.com/index.php?owpn=571&sms_api=Application_starter_SMS>

Send Mail From Java

<http://www.javatpoint.com/example-of-sending-attachment-with-email-using-java-mail-api>

**package** com.mkyong.common;  
  
**import** java.util.Properties;  
  
**import** javax.mail.Message;  
**import** javax.mail.MessagingException;  
**import** javax.mail.PasswordAuthentication;  
**import** javax.mail.Session;  
**import** javax.mail.Transport;  
**import** javax.mail.internet.InternetAddress;  
**import** javax.mail.internet.MimeMessage;  
  
**public** **class** SendMailTLS {  
  
 **public** **static** **void** main(String[] args) {  
  
 **final** String username = "username@gmail.com";  
 **final** String password = "password";  
  
 Properties props = **new** Properties();  
 props.put("mail.smtp.auth", "true");  
 props.put("mail.smtp.starttls.enable", "true");  
 props.put("mail.smtp.host", "smtp.gmail.com");  
 props.put("mail.smtp.port", "587");  
  
 Session session = Session.getInstance(props,  
  **new** javax.mail.Authenticator() {  
 **protected** PasswordAuthentication getPasswordAuthentication() {  
 **return** **new** PasswordAuthentication(username, password);  
 }  
  });  
  
 **try** {  
  
 Message message = **new** MimeMessage(session);  
 message.setFrom(**new** InternetAddress("from-email@gmail.com"));  
 message.setRecipients(Message.RecipientType.TO,  
 InternetAddress.parse("to-email@gmail.com"));  
 message.setSubject("Testing Subject");  
 message.setText("Dear Mail Crawler,"  
 + "**\n\n** No spam to my email, please!");  
  
 Transport.send(message);  
  
 System.out.println("Done");  
  
 } **catch** (MessagingException e) {  
 **throw** **new** RuntimeException(e);  
 }  
 }  
}

Method 2

package crunchify.com.tutorial;

import java.util.Properties;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.AddressException;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

/\*\*

\* @author Crunchify.com

\*

\*/

public class CrunchifyJavaMailExample {

static Properties mailServerProperties;

static Session getMailSession;

static MimeMessage generateMailMessage;

public static void main(String args[]) throws AddressException, MessagingException {

generateAndSendEmail();

System.out.println("\n\n ===> Your Java Program has just sent an Email successfully. Check your email..");

}

public static void generateAndSendEmail() throws AddressException, MessagingException {

//Step1

System.out.println("\n 1st ===> setup Mail Server Properties..");

mailServerProperties = System.getProperties();

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

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

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

System.out.println("Mail Server Properties have been setup successfully..");

//Step2

System.out.println("\n\n 2nd ===> get Mail Session..");

getMailSession = Session.getDefaultInstance(mailServerProperties, null);

generateMailMessage = new MimeMessage(getMailSession);

generateMailMessage.addRecipient(Message.RecipientType.TO, new InternetAddress("test1@crunchify.com"));

generateMailMessage.addRecipient(Message.RecipientType.CC, new InternetAddress("test2@crunchify.com"));

generateMailMessage.setSubject("Greetings from Crunchify..");

String emailBody = "Test email by Crunchify.com JavaMail API example. " + "<br><br> Regards, <br>Crunchify Admin";

generateMailMessage.setContent(emailBody, "text/html");

System.out.println("Mail Session has been created successfully..");

//Step3

System.out.println("\n\n 3rd ===> Get Session and Send mail");

Transport transport = getMailSession.getTransport("smtp");

// Enter your correct gmail UserID and Password (XXXApp Shah@gmail.com)

transport.connect("smtp.gmail.com", "<-- your gmail USERNAME -->", "<-- your gmail PASSWORD -->");

transport.sendMessage(generateMailMessage, generateMailMessage.getAllRecipients());

transport.close();

}

}

***Password Generator***

**import** java.util.\*;  
  
**class** RandomNumbers {  
 **public** **static** **void** main(String[] args) {  
   **int** c;  
   Random t = **new** Random();  
  
   *// random integers in [0, 1000000]*  
  
   **for** (c = 1; c <= 10; c++) {  
     System.out.println(t.nextInt(1000000);  
   }  
 }  
}

***Method 2: alpha numeric***

import java.util.Random;

public class jishnu {

private static final String CHAR\_LIST =

    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";

private static final int RANDOM\_STRING\_LENGTH = 10;// length of the string

/\*\*

\* This method generates random string

\* @return

\*/

public String generateRandomString(){

    StringBuffer randStr = new StringBuffer();

    for(int i=0; i<RANDOM\_STRING\_LENGTH; i++){

        int number = getRandomNumber();

        char ch = CHAR\_LIST.charAt(number);

        randStr.append(ch);

    }

    return randStr.toString();

}

/\*\*

\* This method generates random numbers

\* @return int

\*/

private int getRandomNumber() {

    int randomInt = 0;

    Random randomGenerator = new Random();

    randomInt = randomGenerator.nextInt(CHAR\_LIST.length());

    if (randomInt - 1 == -1) {

        return randomInt;

    } else {

        return randomInt - 1;

    }

}

public static void main(String a[]){

    jishnu msr = new jishnu();

    System.out.println(msr.generateRandomString());

    }

}

***Timer***

<http://www.journaldev.com/1050/java-timer-and-timertask-example-tutorial>

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | package com.journaldev.threads;  import java.util.Date;  import java.util.Timer;  import java.util.TimerTask;  public class MyTimerTask extends TimerTask {     @Override     public void run() {         System.out.println("Timer task started at:"+new Date());         completeTask();         System.out.println("Timer task finished at:"+new Date());     }     private void completeTask() {         try {             //assuming it takes 20 secs to complete the task             Thread.sleep(20000);         } catch (InterruptedException e) {             e.printStackTrace();         }     }       public static void main(String args[]){         TimerTask timerTask = new MyTimerTask();         //running timer task as daemon thread         Timer timer = new Timer(true);         timer.scheduleAtFixedRate(timerTask, 0, 10\*1000);         System.out.println("TimerTask started");         //cancel after sometime         try {             Thread.sleep(120000);         } catch (InterruptedException e) {             e.printStackTrace();         }         timer.cancel();         System.out.println("TimerTask cancelled");         try {             Thread.sleep(30000);         } catch (InterruptedException e) {             e.printStackTrace();         }     }  } |

Notice that one thread execution will take 20 seconds but Timer object is scheduled to run the task every 10 seconds. Here is the output of the program:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | TimerTask started  Timer task started at:Wed Dec 26 19:16:39 PST 2012  Timer task finished at:Wed Dec 26 19:16:59 PST 2012  Timer task started at:Wed Dec 26 19:16:59 PST 2012  Timer task finished at:Wed Dec 26 19:17:19 PST 2012  Timer task started at:Wed Dec 26 19:17:19 PST 2012  Timer task finished at:Wed Dec 26 19:17:39 PST 2012  Timer task started at:Wed Dec 26 19:17:39 PST 2012  Timer task finished at:Wed Dec 26 19:17:59 PST 2012  Timer task started at:Wed Dec 26 19:17:59 PST 2012  Timer task finished at:Wed Dec 26 19:18:19 PST 2012  Timer task started at:Wed Dec 26 19:18:19 PST 2012  TimerTask cancelled  Timer task finished at:Wed Dec 26 19:18:39 PST 2012 |

continuous loading

$(document).load(function(){

   setTimeout(function(){

   $.ajax({

     method: "GET",

     url: "some.php"

   });

   }, 1000);

});

<!DOCTYPE html>

<html>

<head>

<script>

function validateForm() {

var x = document.forms["myForm"]["fname"].value;

if (x == null || x == "") {

alert("Name must be filled out");

return false;

}

}

</script>

</head>

<body>

<form name="myForm" action="demo\_form.asp"

onsubmit="return validateForm()" method="post">

Name: <input type="text" name="fname">

<input type="submit" value="Submit">

</form>

</body>

</html>

<html>

<head>

<title>Mobile number validation using regex</title>

<script type="text/javascript">

function validate() {

var mobile = document.getElementById("mobile").value;

var pattern = /^[\s()+-]\*([0-9][\s()+-]\*){6,20}$/;

if (pattern.test(mobile)) {

alert("Your mobile number : "+mobile);

return true;

}

alert("It is not valid mobile number");

return false;

}

</script>

</head>

<body>

Enter Mobile No. :

<input type="text" name="mobile" id="mobile" />

<input type="submit" value="Check" onclick="validate();" />

</body>

</html>