SMS Sending Through API

Reffered link

<https://api.msg91.com/apidoc/samplecode/java-sample-code-send-sms-xml.php>

[**http://www.asaptech.net/Bulk-SMS-API.html**](http://www.asaptech.net/Bulk-SMS-API.html)

<https://api.msg91.com/apidoc/basic/route-balance.php>

<http://api.msg91.com/user/#delivery_reports_text>

. <https://viasocket.com/collection/msg91-api-integration/5/send-sms-via-msg91/T26A6X72?utm_source=msg91&utm_campaign=soktbutton&utm_medium=referral>

<https://docs.msg91.com/collection/msg91-api-integration/5/send-sms-v2/TZ2IXQHS>

<https://api.msg91.com/apidoc/basic/route-balance.php>

Source code :

**package** ssms;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.net.HttpURLConnection;

**import** java.net.URL;

**import** java.net.URLConnection;

**import** java.net.URLEncoder;

**import** java.util.Properties;

**public** **class** sms23{

**public** **static** **void** main(String args[]) {

//Your authentication key

String authkey = "244090AuosZRagd5bcedcf5";

//Multiple mobiles numbers separated by comma

String mobiles = "9440511264";//,7680808070";

//Sender ID,While using route4 sender id should be 6 characters long.

//String senderId = "611341";

String sender="MSGIND";//"611342";//"8428516197";

//Your message to send, Add URL encoding here.

String message = "Bye";

//define route

String route="4";

String campaign="API";

String countryCode="91";

Properties systemSettings = System.*getProperties*();

systemSettings.put("proxySet", "true");

systemSettings.put("http.proxyHost", "172.16.0.10");

systemSettings.put("http.proxyPort", "8100");

/\*URL u = new URL("http://www.google.com");

HttpURLConnection con = (HttpURLConnection)u.openConnection();\*/

/\*System.out.println(con.getResponseCode() + " : " + con.getResponseMessage());

System.out.println(con.getResponseCode() == HttpURLConnection.HTTP\_OK);\*/

//Prepare Url

URLConnection myURLConnection=**null**;

URL myURL=**null**;

BufferedReader reader=**null**;

//encoding message

String encoded\_message=URLEncoder.~~encode~~(message);

//Send SMS API

String mainUrl="http://api.msg91.com/api/v2/sendsms?";

//Prepare parameter string

StringBuilder sbPostData= **new** StringBuilder(mainUrl);

sbPostData.append("&message="+encoded\_message);

sbPostData.append("&authkey="+authkey);

sbPostData.append("&mobiles="+mobiles);

sbPostData.append("&route="+route);

sbPostData.append("&sender="+sender);

sbPostData.append("&country="+countryCode);

sbPostData.append("&campaign="+campaign);

//final string

mainUrl = sbPostData.toString();

System.***out***.println("mainUrl\n"+mainUrl);

**try**

{

//prepare connection

myURL = **new** URL(mainUrl);

System.***out***.println(myURL);

myURLConnection = myURL.openConnection();

System.***out***.println("success");

myURLConnection.connect();

reader= **new** BufferedReader(**new** InputStreamReader(myURLConnection.getInputStream()));

//reading response

String response;

**while** ((response = reader.readLine()) != **null**)

//print response

System.***out***.println(response);

//finally close connection

reader.close();

}

**catch** (IOException e)

{

e.printStackTrace();

}

}

}

Output

------------

mainUrl

http://api.msg91.com/api/v2/sendsms?&message=Bye&authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&route=4&sender=MSGIND&country=91&campaign=API

http://api.msg91.com/api/v2/sendsms?&message=Bye&authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&route=4&sender=MSGIND&country=91&campaign=API

success

{"message":"5bcf06dd00d716397c15f281","type":"success"}

mainUrl

http://api.msg91.com/api/v2/sendsms?authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&message=hi+am+msg+%21%21%21&route=4&sender=611341

http://api.msg91.com/api/v2/sendsms?message=ho&authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&route=4&sender=8428516197&country=91

mainUrl

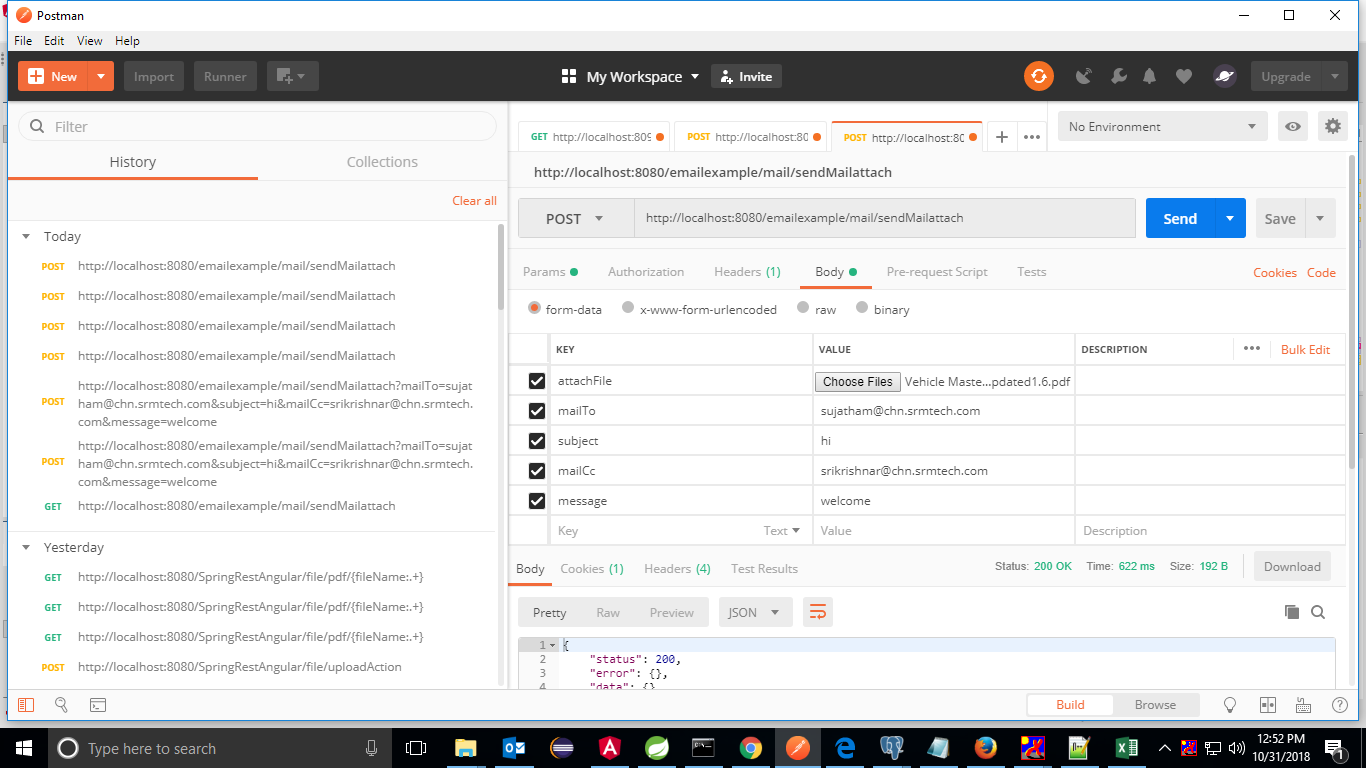
http://api.msg91.com/api/v2/sendsms?&message=hi+am+msg+%21%21%21authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&route=4&sender=8428516197&country=91

<http://api.msg91.com/api/v2/sendsms?message=hi&authkey=244090AuosZRagd5bcedcf5&mobiles=9440511264&route=4&sender=8428516197&country=91>

Login details

[Sujathaanand456@gmail.com](mailto:Sujathaanand456@gmail.com)

Sujatha678



##Mailing Properties

mail.userName=sujatham@chn.srmtech.com

mail.password= srm12345

mail.fromAddress= sujatham@chn.srmtech.com

mail.host=mail.smtp.host

mail.hostname=intranetmail.srmtech.com

mail.port=mail.smtp.port

mail.portNo=25

mail.mailauth=mail.smtp.auth

mail.t=true

mail.starttls=mail.smtp.starttls.enable

mail.sslfactory=mail.smtp.ssl.socketFactory

mail.auth=mail.smtp.auth

mailserviceimpl

package com.srmt.erp.service;

import java.io.IOException;

import java.io.InputStream;

import java.security.GeneralSecurityException;

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.AddressException;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.core.io.InputStreamSource;

import org.springframework.mail.javamail.MimeMessageHelper;

import org.springframework.stereotype.Component;

import org.springframework.web.multipart.MultipartFile;

import com.srmt.erp.common.BadRequestException;

import com.srmt.erp.common.ResponseAPI;

import com.srmt.erp.model.UserMailDetails;

import com.sun.mail.util.MailSSLSocketFactory;

/\*

\*

\*File Name : MailApp.java

\*Description : MailApp.java Implementation for Mailling Bussinessprocess

\*Created Date : 23/10/2018

\*Created By : Murali

\*

\*/

@Component

public class MailServiceImpl {

@Value("${mail.userName}")

public String userName;

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

public String password;

@Value("${mail.fromAddress}")

public String fromAddress;

@Value("${mail.starttls}")

public String starttls;

@Value("${mail.t}")

public String t;

@Value("${mail.host}")

public String host;

@Value("${mail.hostname}")

public String hostname;

@Value("${mail.port}")

public String port;

@Value("${mail.portNo}")

public String portNo;

@Value("${mail.mailauth}")

public String auth;

@Value("${mail.sslfactory}")

public String sslfactory;

/\*

\* MailVo Details sendMail() which process mail business

\*

\* @return ResponseAPI

\*/

public ResponseAPI sendMail(UserMailDetails mailVo,MultipartFile attachFile) throws MessagingException {

ResponseAPI apiResonse = new ResponseAPI();

Properties objProps = new Properties();

objProps.put(host, hostname);

objProps.put(port, portNo);

objProps.put(auth, t);

objProps.put(starttls, t);

MailSSLSocketFactory sf;

try {

sf = new MailSSLSocketFactory();

sf.setTrustAllHosts(true);

objProps.put(sslfactory, sf);

} catch (GeneralSecurityException e) {

e.printStackTrace();

}

Session session = Session.getInstance(objProps, new javax.mail.Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(userName, password);

}

});

MimeMessage msg1 = new MimeMessage(session);

try {

InternetAddress address = new InternetAddress();

address.setAddress(fromAddress);

msg1.setFrom(address);

InternetAddress[] addressTo = new InternetAddress[mailVo.getMailTo().length];

for (int i = 0; i < mailVo.getMailTo().length; i++) {

addressTo[i] = new InternetAddress(mailVo.getMailTo()[i]);

}

msg1.setRecipients(Message.RecipientType.TO, addressTo);

InternetAddress[] addressCc = null;

if (mailVo.getMailCc() != null && mailVo.getMailCc().length > 0) {

addressCc = new InternetAddress[mailVo.getMailCc().length];

for (int i = 0; i < mailVo.getMailCc().length; i++) {

addressCc[i] = new InternetAddress(mailVo.getMailCc()[i]);

}

}

if (addressCc != null) {

msg1.setRecipients(Message.RecipientType.CC, addressCc);

}

msg1.setSubject(mailVo.getSubject());

}

catch (AddressException addressExcep) {

throw new BadRequestException("Address Wrong ");

} catch (Exception exception) {

throw new BadRequestException("Details Wrong");

}

MimeMessageHelper messageHelper=new MimeMessageHelper(msg1,true, "UTF-8");

messageHelper.setTo(mailVo.getMailTo());

messageHelper.setSubject(mailVo.getSubject());

messageHelper.setText(mailVo.getMessage());

messageHelper.setFrom(fromAddress);

String attachName = attachFile.getOriginalFilename();

if (!attachFile.equals("")) {

messageHelper.addAttachment(attachName, new InputStreamSource() {

@Override

public InputStream getInputStream() throws IOException {

return attachFile.getInputStream();

}

});

}

Transport.send(msg1);

return apiResonse;

}

public ResponseAPI sendMail(UserMailDetails mailVo) throws MessagingException {

ResponseAPI apiResonse = new ResponseAPI();

Properties objProps = new Properties();

objProps.put(host, hostname);

objProps.put(port, portNo);

objProps.put(auth, t);

objProps.put(starttls, t);

MailSSLSocketFactory sf;

try {

sf = new MailSSLSocketFactory();

sf.setTrustAllHosts(true);

objProps.put(sslfactory, sf);

} catch (GeneralSecurityException e) {

e.printStackTrace();

}

Session session = Session.getInstance(objProps, new javax.mail.Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(userName, password);

}

});

MimeMessage msg1 = new MimeMessage(session);

try {

InternetAddress address = new InternetAddress();

address.setAddress(fromAddress);

msg1.setFrom(address);

InternetAddress[] addressTo = new InternetAddress[mailVo.getMailTo().length];

for (int i = 0; i < mailVo.getMailTo().length; i++) {

addressTo[i] = new InternetAddress(mailVo.getMailTo()[i]);

}

msg1.setRecipients(Message.RecipientType.TO, addressTo);

InternetAddress[] addressCc = null;

if (mailVo.getMailCc() != null && mailVo.getMailCc().length > 0) {

addressCc = new InternetAddress[mailVo.getMailCc().length];

for (int i = 0; i < mailVo.getMailCc().length; i++) {

addressCc[i] = new InternetAddress(mailVo.getMailCc()[i]);

}

}

if (addressCc != null) {

msg1.setRecipients(Message.RecipientType.CC, addressCc);

}

msg1.setSubject(mailVo.getSubject());

}

catch (AddressException addressExcep) {

throw new BadRequestException("Address Wrong ");

} catch (Exception exception) {

throw new BadRequestException("Details Wrong");

}

MimeMessageHelper messageHelper=new MimeMessageHelper(msg1,true, "UTF-8");

messageHelper.setTo(mailVo.getMailTo());

messageHelper.setSubject(mailVo.getSubject());

messageHelper.setText(mailVo.getMessage());

messageHelper.setFrom(fromAddress);

Transport.send(msg1);

return apiResonse;

}

}

Model

**package** com.srmt.erp.model;

/\*

\*

\*File Name : UserMailDetails.java

\*Description : UserMailDetails is Model which contain's MailingProperties

\*Created By : Murali

\*

\*/

**public** **class** UserMailDetails {

**private** String[] mailTo;

**private** String[] mailCc;

**private** String subject;

**private** String message;

**public** String[] getMailTo() {

**return** mailTo;

}

**public** **void** setMailTo(String[] mailTo) {

**this**.mailTo = mailTo;

}

**public** String getSubject() {

**return** subject;

}

**public** **void** setSubject(String subject) {

**this**.subject = subject;

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

/\*\*

\* Getter method for mailCc

\*

\* **@return** the mailCc (String[])

\*/

**public** String[] getMailCc() {

**return** mailCc;

}

/\*\*

\* Setter method for mailCc

\*

\* **@param** mailCc the mailCc to set

\*/

**public** **void** setMailCc(String[] mailCc) {

**this**.mailCc = mailCc;

}

/\*\*

\* Getter method for attachment

\*

\* **@return** the attachment (String)

\*/

}

Controller

package com.srmt.erp.controller;

import java.security.GeneralSecurityException;

import javax.mail.MessagingException;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.multipart.MultipartFile;

import com.srmt.erp.common.ResponseAPI;

import com.srmt.erp.model.UserMailDetails;

import com.srmt.erp.service.MailServiceImpl;

/\*

\*

\*File Name : SimpleMailController.java

\*Description : SimpleMailController Which acts as Rest Controller

\*Created By : Murali

\*

\*/

@RestController

@RequestMapping("/mail")

public class SimpleMailController {

@Autowired

private MailServiceImpl mail;

/\*

\* @RequestBody UserMailDetails Details

\* @return String

\*/

@PostMapping("/sendMailattach")

public ResponseEntity<ResponseAPI> sendMailBy(@RequestParam("mailTo") String[] mailTo,@RequestParam("subject") String subject,@RequestParam("message") String message,@RequestParam("mailCc") String[] mailCc,@RequestParam MultipartFile attachFile) throws GeneralSecurityException, MessagingException {

UserMailDetails mailVo=new UserMailDetails();

mailVo.setMailTo(mailTo);

mailVo.setMailCc(mailCc);

mailVo.setSubject(subject);

mailVo.setMessage(message);

ResponseAPI apiResponse = mail.sendMail(mailVo,attachFile);

return ResponseEntity.status(apiResponse.getStatus()).body(apiResponse);

}

@PostMapping("/sendMail")

public ResponseEntity<ResponseAPI> sendMailBy(@RequestParam("mailTo") String[] mailTo,@RequestParam("subject") String subject,@RequestParam("message") String message,@RequestParam("mailCc") String[] mailCc) throws GeneralSecurityException, MessagingException {

UserMailDetails mailVo=new UserMailDetails();

mailVo.setMailTo(mailTo);

mailVo.setMailCc(mailCc);

mailVo.setSubject(subject);

mailVo.setMessage(message);

ResponseAPI apiResponse = mail.sendMail(mailVo);

return ResponseEntity.status(apiResponse.getStatus()).body(apiResponse);

}

}

package com.srmt.erp;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

/\*

\*@Author:Murali

\*

\*

\*

\*/

@SpringBootApplication

public class SampleErpMailApplication extends SpringBootServletInitializer{

@Override

protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

return application.sources(SampleErpMailApplication.class);

}

public static void main(String[] args) {

SpringApplication.run(SampleErpMailApplication.class, args);

}

}

Common folder

**package** com.srmt.erp.common;

**import** java.util.List;

/\*

\*

\*File Name : BadRequestException.java

\*Description : BadRequestException for Mail api

\*Created By : Murali

\*

\*/

**public** **class** BadRequestException **extends** RuntimeException{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** List<ErrorMessage> details;

**public** BadRequestException(String message) {

**super**(message);

}

**public** BadRequestException(String message,List<ErrorMessage> details) {

**super**(message);

**this**.details = details;

}

/\*\*

\* **@return** the details

\*/

**public** List<ErrorMessage> getDetails() {

**return** details;

}

}

**package** com.srmt.erp.common;

**import** org.springframework.http.HttpStatus;

/\*

\*

\*File Name : ErrorCode.java

\*Description : Custom errors and ErrorCodes

\*Created By : Murali

\*

\*/

**public** **enum** ErrorCode {

***OK***(HttpStatus.***OK***), //200

***INVALID\_ARGUMENT***(HttpStatus.***BAD\_REQUEST***), //400

***NOT\_FOUND***(HttpStatus.***NOT\_FOUND***), //404

***INTERNAL\_SERVER\_ERROR***(HttpStatus.***INTERNAL\_SERVER\_ERROR***),//500

;

**private** HttpStatus statusCode;

/\*\*

\* **@param** statusCode

\*/

**private** ErrorCode(HttpStatus statusCode) {

**this**.statusCode = statusCode;

}

**public** Integer value() {

**return** statusCode.value();

}

**public** HttpStatus statusCode() {

**return** statusCode();

}

}

package com.srmt.erp.common;

import java.util.List;

import com.fasterxml.jackson.annotation.JsonInclude;

/\*

\*

\*File Name : Error.java

\*Description : represents API errors.

\*Created By : Murali

\*

\*/

@JsonInclude(JsonInclude.Include.NON\_NULL)

public class ErrorMessage {

private String code;

private String message;

private List<ErrorMessage> details;

/\*\*

\*

\*/

public ErrorMessage() {

super();

}

/\*\*

\* @param code

\* @param message

\*/

public ErrorMessage(String code, String message) {

super();

this.code = code;

this.message = message;

}

/\*\*

\* @return the code

\*/

public String getCode() {

return code;

}

/\*\*

\* @param code the code to set

\*/

public void setCode(String code) {

this.code = code;

}

/\*\*

\* @return the message

\*/

public String getMessage() {

return message;

}

/\*\*

\* @param message the message to set

\*/

public void setMessage(String message) {

this.message = message;

}

/\*\*

\* @return the details

\*/

public List<ErrorMessage> getDetails() {

return details;

}

/\*\*

\* @param details the details to set

\*/

public void setDetails(List<ErrorMessage> details) {

this.details = details;

}

/\* (non-Javadoc)

\* @see java.lang.Object#toString()

\*/

@Override

public String toString() {

return "Error [code=" + code + ", message=" + message + ", details=" + details + "]";

}

}

package com.srmt.erp.common;

import java.util.HashMap;

import java.util.Objects;

/\*

\*

\*File Name : APIResponse.java

\*Description : this class represents the general response structure of Email api

\*Created By : Murali

\*

\*/

public class ResponseAPI {

private Integer status;

private ErrorMessage error;

private Object data;

/\*\*

\* @param status

\* @param error

\* @param data

\*/

public ResponseAPI() {

super();

this.status = ErrorCode.OK.value();

this.error = new ErrorMessage();

this.data = new HashMap<String,Object>();

}

/\* (non-Javadoc)

\* @see java.lang.Object#hashCode()

\*/

@Override

public int hashCode() {

return Objects.hash(status,error,data);

}

/\* (non-Javadoc)

\* @see java.lang.Object#equals(java.lang.Object)

\*/

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

ResponseAPI other = (ResponseAPI) obj;

return Objects.equals(status, other.status) && Objects.equals(error, other.error) && Objects.equals(data, other.data);

}

/\*\*

\* @return the status

\*/

public Integer getStatus() {

return status;

}

/\*\*

\* @param status the status to set

\*/

public void setStatus(Integer status) {

this.status = status;

}

/\*\*

\* @return the error

\*/

public ErrorMessage getError() {

return error;

}

/\*\*

\* @param error the error to set

\*/

public void setError(ErrorMessage error) {

this.error = error;

}

/\*\*

\* @return the data

\*/

public Object getData() {

return data;

}

/\*\*

\* @param data the data to set

\*/

public void setData(Object data) {

this.data = data;

}

/\* (non-Javadoc)

\* @see java.lang.Object#toString()

\*/

@Override

public String toString() {

return "APIResponse [status=" + status + ", error=" + error + ", data=" + data + "]";

}

}