ARTIGO: 11580
Send Mail Mobile Mind

NodeJS

```
var Client = require('node-rest-client').Client
var hmacsha1 = require('hmacsha1')
var apiKey = 'apy key code'
var apiEndPoint = 'http://sendmail.mobilemind.com.br/api/send'

var entity = {
  'subject': 'my subject'
  'body': 'my html body'
  'fromName': "My from name"
  'to': to
  'application' : "my app name"
}


var data = new Buffer(JSON.stringify(entity)).toString("hex")

var signatureHash = hmacsha1(apiEndPoint, data)

var args = {
  data: entity,
  headers: {
    "Content-Type": "application/json",
    "X-Hub-Signature": signatureHash
  }
}

client.post(api_url, args, function(data, response){

   if(response.statusCode == 200)
     // check error
   esle
    // success

})
```

Go Lang

```
import (
  "encoding/json"
  "encoding/hex"
  "encoding/base64"
  "crypto/sha1"
  "crypto/hmac"
  "io/ioutil"
  "net/http"
  "strings"
  "bytes"
  "ftm"
)

const (
  ApiKey = "api key"
  ApiEndPoint = "http://sendmail.mobilemind.com.br/api/send"
)

func SendEmail(){
  entity := map[string]string{}
  entity["subject"] = "my subject"
  entity["to"] = "to@email.com"
  entity["application"] = "aplication name"
  entity["body"] = "my html body"

  jsonData, _ := json.Marshal(entity)

  signatureHash := GenerateHash(jsonData)

  data := bytes.NewBuffer(jsonData)

  client := &http.Client{}
  req, _ := http.NewRequest("POST", ApiEndPoint, data)

  req.Header.Add("Content-Type", "application/json")
  req.Header.Add("X-Hub-Signature", signatureHash)
```

```
  r, _ := client.Do(req)

  response, _ := ioutil.ReadAll(r.Body)

  fmt.Println(string(response))

}

func GenerateHash(body []byte) string {
  mac := hmac.New(sha1.New, []byte(ApiKey))

  bodyHex := []byte(hex.EncodeToString(body))

  mac.Write(bodyHex)
  rawBodyMAC := mac.Sum(nil)
  computedHash := base64.StdEncoding.EncodeToString(rawBodyMAC)

  return computedHash
}
```

Groovy

```
import groovy.json.JsonOutput
import groovy.json.JsonSlurper
import javax.crypto.spec.SecretKeySpec
import javax.crypto.Mac
import java.net.URL

def data = [
  subject: 'app test',
  body: 'app test',
  fromName: "Mobile Mind",
  to: "ricardo@mobilemind.com.br"
]

def jsonData = JsonOutput.toJson(data)

def appId = "app id"
def serverEndpoint = "http://sendmail.mobilemind.com.br/api/send"
def appKey = "app key"

def jsonHex = jsonData.bytes.encodeHex()

SecretKeySpec signingKey = new SecretKeySpec(appKey.bytes, "HmacSHA1");
Mac mac = Mac.getInstance("HmacSHA1");
mac.init(signingKey);

// compute the hmac on input data bytes
byte[] rawHmac = mac.doFinal("${jsonHex}".bytes);
def calculedSignature = rawHmac.encodeBase64()

def baseUrl = new URL(serverEndpoint)
def connection = baseUrl.openConnection()

connection.addRequestProperty("Content-Type", "application/json")
connection.addRequestProperty("X-Hub-Application", appId)
connection.addRequestProperty("X-Hub-Signature", "${calculedSignature}")

connection.with {
  doOutput = true
  requestMethod = 'POST'
  outputStream.withWriter { writer ->
    writer << jsonData
  }

  def jsonSlurper = new JsonSlurper()
  def result = jsonSlurper.parseText(content.text)

  println content.text

  if(result["status"] == "success")
      println result["id"] // sms id

}
```

Python

```
#-*- coding: utf8 -*-
import requests
import json
import urllib, cStringIO, base64
from datetime import datetime, time, timedelta
from gzip import GzipFile
import gzip, os
```

```
from binascii import hexlify
import base64, hmac, hashlib
from collections import OrderedDict

class EmailService():

  apiKey = "apy_key"
  apiUrl = "http://sendmail.mobilemind.com.br/api/send"
  apiApp = "app_name"

  def send(self, subject, body, to):

    entity = OrderedDict([
      ("subject", subject),
      ("body", body),
      ("fromName", "My App"),
      ("to", to),
      ("application", self.apiApp)
    ])

    json_dump = json.dumps(entity, sort_keys=False)

    msg = json_dump.encode('hex')

    signature = hmac.new(self.apiKey, msg=msg, digestmod=hashlib.sha1).digest()

    signature_hash = base64.b64encode(signature)

    headers = {
      "Content-Type": "application/json",
      "X-Hub-Signature": signature_hash
    }

    try:
      response = requests.post(self.apiUrl, data=json.dumps(entity), headers=headers)
      print("%s - %s", (response.status_code, response.json()))
    except Exception, e:
  raise Exception("%s - %s", (response.status_code, response.json()))
```