```
const moment = require('moment');
const logger = require('../../services/logger');
const _ = require('lodash');
var nodemailer = require("nodemailer");
var smtpTransport = nodemailer.createTransport({
  host: 'sdcsmtp02.sentosa.com.sg',
  port: 25,
  tls: {
    rejectUnauthorized: false
  }
});

var smtpTransportBackup = nodemailer.createTransport({
  host: 'sdcsmtp01.sentosa.com.sg',
  port: 25,
  tls: {
    rejectUnauthorized: false
  }
});

var fs = require('fs')
require.extensions['.txt'] = function(module, filename) {
  module.exports = fs.readFileSync(filename, 'utf8');
};
const terms = require('./email_template.txt')
var sg = require('sendgrid')('SG.CS75JJdmT2-
p4RwDlEVePA.EnsVDy2t21G6OfRFDJ3XmVjUmorX_uIDSYFVfCQKTLw');

exports.smtpMail = (recipient, order, latest_order, permutation) => {
  // permutation indicates whether if it's no wot (-1), wot only (1), or both wot
and normal attractions (0)
  let pins_html = ''
  let barcodes_html = ''
  if(permutation === -1) { // no wot
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-
size: 14px">' + order.pincode.pin + '</strong></td><td></td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><img src="https://
ospapim.azure-api.net/open/barcode?pin=' + order.pincode.pin + '"
alt="tickets pincode"></td><td></td></tr></table></tr>'
  } else if(permutation === 0) { // both
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-
size: 14px">' + order.pincode.pin + '</strong></td><td></td></
tr><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' +
order.pincode.pin + '" alt="tickets pincode"></td><td></td></tr></table></
```

```
td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><strong
style="font-size: 14px">' + order.pincode.wot + '</strong><td><td><td></
tr><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' +
order.pincode.wot + '" alt="wot pincode"></td><td></td></tr></table></td></
tr>'
  } else { // only wot
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-
size: 14px">' + order.pincode.wot + '</strong></td><td></td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><img src="https://
ospapim.azure-api.net/open/barcode?pin=' + order.pincode.wot + '"
alt="tickets pincode"></td><td></td></tr></table></tr>'
  }
  let payment = order.payment_mode
  let email_wrapper = terms

  email_wrapper = email_wrapper.replace(/{{pins}}/g, pins_html)
                  .replace(/{{barcodes}}/g, barcodes_html)
                  .replace(/{{purchase_date}}/g,
moment(latest_order.date_time).format("DD-MM-YYYY"))
                  .replace(/{{receipt_number}}/g, order.receipt_number)
                  .replace(/{{orders}}/g, generateEmailOrderSummary(order))
                  .replace(/{{order_total}}/g, (parseInt(order.total)/
100.0).toFixed(2))
                  .replace(/{{order_totals}}/g, (parseFloat(((order.total*7)/
(100.0)))/100.0).toFixed(2))
                  .replace(/{{name}}/g, latest_order.name)
                  .replace(/{{email}}/g, latest_order.email)
                  .replace(/{{mobile}}/g, latest_order.contact)
                  .replace(/{{payment}}/g, latest_order.payment_mode)

  let mailOptions = {
    from: 'Sentosa Sales <sales@sentosa.com.sg>',
    to: recipient,
    subject: '[OSP] Thank you for your order',
    html: email_wrapper
  }

  smtpTransport.sendMail(mailOptions, function(err, response) {
    if(err) {
      console.log(err)
      logger.log('smtp', '', 'error sending email to ' + recipient + ' with messsage '
+ err, 'error', {}, null, err);
      // attempt backup smtp
      smtpTransportBackup.sendMail(mailOptions, function(err2,
backup_response) {
        if(err2) {
          console.log(err2)
          logger.log('smtp', '', 'error sending email via backup to ' + recipient + ' with messsage ' + err2, 'error', {}, null, err2);
        } else {
          console.log(backup_response.message)
          logger.log('smtp', '', 'sent email via backup to ' + recipient + ' success with message: ' + backup_response.message, 'info', {}, null, {});
        }
      })
    } else {
      console.log(response.message)
      logger.log('smtp', '', 'sent email to ' + recipient + ' success with message: ' + response.message, 'info', {}, null, {});
    }
  })
}

exports.confirmEmail = (recipient, order, latest_order, permutation) => {
  // permutation indicates whether if it's no wot (-1), wot only (1), or both wot and normal attractions (0)
  let pins_html = ''
  let barcodes_html = ''
  if(permutation === -1) { // no wot
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-size: 14px">' + order.pincode.pin + '</strong></td><td></td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' + order.pincode.pin + '" alt="tickets pincode"></td><td></td></tr></table></tr>'
  } else if(permutation === 0) { // both
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-size: 14px">' + order.pincode.pin + '</strong></td><td></td></tr><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' + order.pincode.pin + '" alt="tickets pincode"></td><td></td></tr></table></td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><strong style="font-size: 14px">' + order.pincode.wot + '</strong><td><td><td></tr><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' + order.pincode.wot + '" alt="wot pincode"></td><td></td></tr></table></td></tr>'
  } else { // only wot
    pins_html = '<tr><td><table width="100%"><tr><td><strong style="font-size: 14px">' + order.pincode.wot + '</strong></td><td></td></tr>'
    barcodes_html = '<tr><td><table width="100%"><tr><td><img src="https://ospapim.azure-api.net/open/barcode?pin=' + order.pincode.wot + '" alt="tickets pincode"></td><td></td></tr></table></tr>'
  }
  let payment = order.payment_mode
  let email_wrapper = terms
   var recipient='pavani.racham@accenture.com'
  email_wrapper = email_wrapper.replace(/{{pins}}/g, pins_html)
                               .replace(/{{barcodes}}/g, barcodes_html)
                               .replace(/{{purchase_date}}/g,
moment(latest_order.date_time).format("DD-MM-YYYY"))
                               .replace(/{{receipt_number}}/g, order.receipt_number)
                               .replace(/{{orders}}/g, generateEmailOrderSummary(order))
                               .replace(/{{order_total}}/g, (parseInt(order.total)/
100.0).toFixed(2))
                               .replace(/{{order_totals}}/g, (parseFloat(((order.total*7)/
(100.0)))/100.0).toFixed(2))
                               .replace(/{{name}}/g, latest_order.name)
                               .replace(/{{email}}/g, latest_order.email)
                               .replace(/{{mobile}}/g, latest_order.contact)
                               .replace(/{{payment}}/g, latest_order.payment_mode)
  var sg_request = sg.emptyRequest({
    method: 'POST',
    path: '/v3/mail/send',
    body: {
      personalizations: [
        {
          to: [{
            email: recipient
          }],
          subject: '[OSP] Thank you for your order'
        }
      ],
      from: {
        email: 'noreply@sentosa.com.sg'
      },
      content: [{
        type: 'text/html',
        value: email_wrapper
      }]
    }
  });
  sg.API(sg_request, function(err, response) {
    if(err) {
      console.log("Error response received")
      logger.log('smtp', '', 'error sending email to ' + recipient + ' with messsage '
+ err, 'error', {}, null, err);
    } else {
      logger.log('smtp', '', 'sent email to ' + recipient + ' success with message: ' +
response.message, 'info', {}, null, {});
    }
    console.log(response.statusCode);
    console.log(response.body);
    console.log(response.headers);
  })
}

const generateEmailOrderSummary = (orders) => {
  let order_types = ''
  let attraction_title
  _.filter(orders.attractions, (attraction) => {
    let line_items = ''
    let line_total = 0
    _.filter(attraction.tickets, (ticket) => {
      let item_name = ticket.description
      _.filter(ticket.types, (type) => {
        let item_price = type.price
        if(typeof type.promo_code !== 'undefined')  {
          item_price = type.new_price
        }
        line_total += (_.round(item_price / 100, 2) * type.quantity)
        // special handler for wot tickets (display date time)
        if(attraction.id === 490) {
          line_items += '<tr> <td>' + item_name + '</td><td>Ticket Type: ' + 
type.type + '<br/>Schedule: ' + moment(ticket.date).format("DD-MM-YYYY") + 
', ' + ticket.time + '</td><td>$' + (item_price / 100.0).toFixed(2) + '</td><td>' + 
type.quantity + '</td><td>$' + (item_price / 100.0 * type.quantity).toFixed(2) + 
'</td></tr>'
        } else {
          line_items += '<tr> <td>' + item_name + '</td><td>Ticket Type: ' + 
type.type + '</td><td>$' + (item_price / 100.0).toFixed(2) + '</td><td>' + 
type.quantity + '</td><td>$' + (item_price / 100.0 * type.quantity).toFixed(2) + 
'</td></tr>'
        }
      })
    })
    order_types += '<table width="100%" cellpadding="5px"><tr> <td>
<strong>Attractions - ' + attraction.title + '</strong> </td></tr><tr 
style="color: white; background-color: #38cef7; font-weight: bold">
<td>Item</td><td>Details</td><td>Qty Price</td><td>Qty</
td><td>Amount</td></tr>' + line_items + '</table><table 
width="100%"><tr><td  width="90%" style="display: block; border-top: 1px 
solid #eff0f2; text-align: right"> <strong>SUBTOTAL (S$):</strong> </td><td 
width="10%" style="display: block; border-top: 1px solid #eff0f2; text-align: 
right"><strong>' + line_total.toFixed(2) + '</strong> </td></tr></table>'
```

```
  })
  return order_types
}
```