



<https://verifone.cloud/docs/oakos/writing-application/making-payment-and-printing-receipt>

Updated: 01-Dec-2020

Making a payment and printing a receipt

INFO: Prerequisite

It is recommended to read the [Writing an application](#) and familiarize yourself with the [OakOS API](#) before continuing.

Talking to a Verifone P400 terminal

OakOS has built into the OS a proxy to handle the communications to the terminal. All that needs to be sent is a JSON payload to our PSDK proxy and the terminal sale will be triggered and report back to the status.

A typical cart might be collected on the client-side of the application like this:

JSON request

```
{
  "items": [
    {
      "name": "Item 1",
      "subtotal": "20.2",
      "taxRate": "0.085",
      "total": "21.916999999999998"
    },
    {
      "name": "Item 2",
      "subtotal": "20.2",
      "taxRate": "0.085",
      "total": "21.916999999999998"
    }
  ],
  "cart": {
    "total": "40.4",
    "tax": "3.434",
    "taxRate": "0.085",
    "grandTotal": "43.833999999999996"
  },
  "terminalIp": "192.168.86.43"
}
```

Note that all values should be converted to a string. This entire cart, including the items array is sent to the app-payment demo [print-receipt.js](#) script to illustrate how to build a printer receipt into a PDF for printing. The only fields that are necessary for sending a payment to the Verifone terminal proxy looks like this:

Payment Request Payload

```
{
  "card": {
    "total": "40.4",
    "tax": "3.434",
    "taxRate": "0.085",
    "grandTotal": "43.833999999999996"
  },
  "terminalIp": "192.168.86.43"
}
```

can be sent directly to the proxy by using a node exec command:

payment.js

```
const { exec } = require('child_process')

let data =
`{"card":{"total":"40.4","tax":"3.434","taxRate":"0.085","grandTotal":"43.833999999999996"},
${process.env.TERMINAL_IP}}`

async function sendCard(payload) {
  await exec(
    `curl --header "Content-Type: application/json" --request POST --data '${data}' localhost:8003` (err, stdout, stderr) => {
    if (err) {
      // node couldn't execute the command
      console.log(`Error: ${err}`);
      return;
    }
    // the *entire* stdout and stderr (buffered)
    console.log(`stdout: ${stdout}`);
    console.log(`stderr: ${stderr}`);
  })
}
```

This code can be found at [test/payment.js](#) of the app-payment project

Installing the demo application

Using the OakOS Dashboard, the [app-payment](#) POC can be used to test running a payment. This app also demonstrates the printing ability.