onomondo OTA / onomondo-softsim-cli-todo

(Haddad Rafik LCS 14-05-2024)

based on https://github.com/onomondo/onomondo-softsim-cli

with OTA GW SMPP https://github.com/onomondo/smpp-gateway

Module provides SMPP External Short Messaging Entity (ESME) behaviour ’out of the box’.

It also contains Express routers for HTTP to SMPP gateway and Prometheus metrics.

Based on JSON FORMAT to OTA system

```
{
  "password": "passw0rd",
  "destination": "111111111",
  "text": "hello",
  "encoding": "ucs2",
  "udh": {
    "port": 37273
  }
}
```

Add authentication and encryption mechanisms to secure downloads and updates.

This simple project can be improved and adapted according to the specific needs of your OTA system.

here an example using nodejs (Page 2 server nodejs)

Use small tools to assist with provisioning your SoftSIM-enabled device both before and during production.(onomondo)

This command retrieves a specified number of profiles from the Onomondo API using the SoftSIM API credentials generated on our platform.

This SoftSIM is based for smart material NORDIC modem (specific for soft/ esim or isim)

using management profiles:

Update History:

- Keep track of applied updates, including firmware versions, installation dates, and the results of each update (success or failure).

Device Management:

- Store information about each connected device, such as unique identifiers, current firmware versions, and other device-specific parameters.

Logs and Diagnostics:

- Record error logs and diagnostic data for each update, which can be useful for troubleshooting and continuous system improvement.

```
const express = require('express');

const multer = require('multer');

const path = require('path');

const app = express();

const PORT = 3000;

// Configuration du stockage des fichiers

const storage = multer.diskStorage({

    destination: (req, file, cb) => {

        cb(null, 'public/firmware');

    },

    filename: (req, file, cb) => {

        cb(null, file.originalname);

    }

});

const upload = multer({ storage: storage });

app.post('/upload', upload.single('firmware'), (req, res) => {

    res.send('Firmware uploaded successfully');

});

app.get('/firmware/latest', (req, res) => {

    const firmwarePath = path.join(__dirname, 'public/firmware/latest-firmware.bin');

    res.download(firmwarePath);

});

app.listen(PORT, () => {

    console.log(`Server is running on http://localhost:${PORT}`);
});
```