

```

////////////////////////////////////
//                                                                    //
//                                                                    //
//                                API Gestió dels aparcaments        //
//                                Servidor UdG                        //
//                                                                    //
////////////////////////////////////

require("dotenv").config();

const express = require("express");//Servidor HTTP

const axios = require("axios").default;//Client HTTP per enviar al TTN
//const schedule = require('node-schedule');

//Executem el servidor i client
const app = express();
const port = 40300;

//Base de dades MySQL
var mysql      = require('mysql');

var connection = mysql.createConnection({
  host      : 'localhost',
  user      : 'root',
  password  : '',
  database  : 'mydb'
});

connection.connect(error => {
  if (error) throw error;
  console.log('Database server running!!!');
});

//Declara l'estructura de rebuda com a JSON
app.use(express.json());

//Pàgina principal

```

```

app.get("/", (req, res) => res.send(`
  <html>
    <head><title>Success!</title></head>
    <body>
      <h1>Benvingut al servidor de gestió </h1>
      
    </body>
  </html>
`
));

// Base de dades de Proves, Trastejar aquí
app.post("/provadb", async (req, res) =>{
  const sql = 'INSERT INTO Sensor_Cotxel SET ?';
  const cotxeObj = {
    //Data: req.body.uplink_message.received_at,
    Data: new Date(req.body.uplink_message.received_at),
    DevEUI: req.body.end_device_ids.device_id,
    Parking_status: req.body.uplink_message.decoded_payload.Parking_status,
    Battery_Voltage: req.body.uplink_message.decoded_payload.Battery_Voltage,
    Direction: req.body.uplink_message.decoded_payload.Direction,
    Frame_type: req.body.uplink_message.decoded_payload.Frame_type,
    Sens_type: req.body.uplink_message.decoded_payload.Sens_type,
    Temp: req.body.uplink_message.decoded_payload.Temp,
    X_Axis: req.body.uplink_message.decoded_payload.X_Axis,
    Y_Axis: req.body.uplink_message.decoded_payload.Y_Axis,
    Z_Axis: req.body.uplink_message.decoded_payload.Z_Axis,
  }
  //Guardem a la DB Sensor_Cotxel
  connection.query(sql, cotxeObj, error =>{
    if (error) throw error;
    console.log('Sensor cotxe rebut');
  });

  const sql2 = 'SELECT DevEUI FROM Sensor_Cotxel ORDER BY Data DESC LIMIT 5';
  let variable = {};

```

```

connection.query(sql2, (error, result) =>{
  if (error) throw error;
  variable = {result};
  console.log('Sensor cotxe rebut');
  res.status(200).send(variable);
});
console.log('-----');

})

//Post missatges rebuts de la API LoRa, TTN V3
app.post("/sensor", async (req, res) =>{

  //Mirem si és sensor Cotxe o actuator LED
  if (req.body.uplink_message.hasOwnProperty('decoded_payload')){
    let sensor = req.body.uplink_message.decoded_payload.Type_Sens;
    if (sensor == 1){
      //Preparem dades del sensor cotxe
      const sql = 'INSERT INTO Sensor_Cotxe SET ?';
      const cotxeObj = {
        Data: new Date(req.body.uplink_message.received_at),
        DevEUI: req.body.end_device_ids.device_id,
        Parking_status:
req.body.uplink_message.decoded_payload.Parking_status,
        Battery_Voltage:
req.body.uplink_message.decoded_payload.Battery_Voltage,
        Direction: req.body.uplink_message.decoded_payload.Direction,
        Frame_type: req.body.uplink_message.decoded_payload.Frame_type,
        Sens_type: req.body.uplink_message.decoded_payload.Sens_type,
        Temp: req.body.uplink_message.decoded_payload.Temp,
        X_Axis: req.body.uplink_message.decoded_payload.X_Axis,
        Y_Axis: req.body.uplink_message.decoded_payload.Y_Axis,
        Z_Axis: req.body.uplink_message.decoded_payload.Z_Axis,
      }
      //Guardem a la Taula Sensor_Cotxe
      connection.query(sql, cotxeObj, error =>{
        if (error) throw error;
        console.log('Sensor cotxe ' + cotxeObj.DevEUI + ' rebut');
      });
    }
  }
});

```

```

});

//Actualitzem la Taula Gestio_Cotxe

const sql2 = 'UPDATE Gestio_Cotxe SET Parking_Status = ' +
mysql.escape(cotxeObj.Parking_Status) + ' WHERE DevEUI_cotxe =' +
mysql.escape(cotxeObj.DevEUI);

//Guardem a la DB Gestio_Cotxe
connection.query(sql2, error =>{
  if (error) throw error;
  console.log('Sensor cotxe ' + cotxeObj.DevEUI + ' actualitzat');
});

}

else if (sensor == 2){

  let ledEUI =req.body.end_device_ids.device_id;

  const ledObj = { Estat_led:
req.body.uplink_message.decoded_payload.LED }

  //console.log(ledObj);

  //Preparem dades del sensor cotxe

  const sql = 'UPDATE Gestio_Cotxe SET Estat_led =' +
mysql.escape(ledObj.Estat_led) + ' WHERE DevEUI_led =' + mysql.escape(ledEUI);

  //Guardem a la Taula Gestio_Cotxe
  connection.query(sql, error =>{
    if (error) throw error;
    console.log('Actuador ' + ledEUI + ' led actualitzat');
  });
}

}

res.status(200).send();
})

//Post Creació Postman dels sensors
app.post("/registre", async (req, res) =>{

  //Mirem si ja existeix

```

```

let cotxe = req.body.EUICotxe;

let c_cotxe = 'A';

let led = req.body.EUILED;

let c_led = 'A';

//Al no ser async espera a recopilar l'informació i llavors decideix
function Logica(c_cotxe, c_led){
  if (c_cotxe !== cotxe & c_led !== led) {
    //Preparem dades dela DB gestió cotxe
    const sql2 = `INSERT INTO Gestio_Cotxe SET ?`;
    const cotxeObj = {
      DevEUI_cotxe: req.body.EUICotxe,
      Parking_status: 0,
      Downlinks_sent: 0,
      DevEUI_led:req.body.EUILED,
      Estat_led:0,
    }
    //Guardem a la DB Sensor_Cotxe
    connection.query(sql2, cotxeObj, error =>{
      if (error) throw error;
      console.log('REGISTRAT!');
    });
  }
  else if ( c_cotxe == cotxe & c_led == led ) {
    console.log('El sensor del cotxe i led ja estan registrats!');
  }
  else if ( c_cotxe == cotxe & c_led !== led ) {
    console.log('Sensor cotxe ja registrat!');
  }
  else if ( c_cotxe !== cotxe & c_led == led ) {
    console.log('Actuador LED ja registrat!');
  }
  else {
    console.log('Error desconegut');
  }
}

```

```

//Preguntem a la base de dades si ja existeix algun camp
const sql = 'SELECT * FROM Gestio_Cotxe WHERE DevEUI_cotxe = ' +
mysql.escape(cotxe) + 'OR DevEUI_led = ' + mysql.escape(led);
connection.query(sql, (error, result) =>{
    if (error) throw error;
    if (result.length > 0){
        c_cotxe = result[0].DevEUI_cotxe;
        c_led = result[0].DevEUI_led;
    }
    Logica(c_cotxe,c_led);
});

res.status(200).send();
})

//Downlink LoRa Proves
// 1 = AQ== ----- 0 = AA==
app.post("/downlink", async (req, res) =>{
    const sended = 'AA==';
    axios({
        headers: {'Authorization': 'Bearer
NNSXS.5SFWX4EHPY67ECSZHX26BVPRIPDVN7ZZIZV77KA.DPW4CBGI3TU2GF3BOY2DY7OOWPKBCDFX
LHTUONZFOLYNZE25AYZA',
        'Content-Type': 'application/json',
        'User-Agent': 'proves-cotxe/v3'},
        method: 'post',
        url: 'https://eul.cloud.thethings.network/api/v3/as/applications/proves-
cotxe/webhooks/api-webhook-udg/devices/' + 'eui-70b3d57ed004dalc' +
'/down/push',
        data: {"downlinks": [{
            "frm_payload": sended,
            "f_port": 15,
            "priority":"NORMAL"
        }]}
    });
    res.status(200).send();
})

```

```

//-----Funcions repetitives de gestió lògica-----//

// Aquesta funció mira la taula on hi han registrats tots el Sensors i envia
l'ordre per aquells necessaris d'activar el LED

const BuffRev = setInterval(RevDevEUI, 30000); //Cada 30 segons
const BuffRevDesac = setInterval(RevDevEUIDesac, 30000); //Cada 30 segons

//Funció que revisa i crea la llista que envia els downlinks d'ACTIVACIÓ
function RevDevEUI(){
    let indxbuff = 0;
    let arrsensor = [];

    const sql = 'SELECT * FROM Gestio_Cotxe WHERE Parking_Status = 1 AND
Downlinks_sent = 0 AND Estat_led = 0 ';

    connection.query(sql, (error, result) =>{
        if (error) throw error;
        if (result.length > 0){
            indxbuff = result.length;
            arrsensor = result;
            //console.log(arrsensor[0]);
            //console.log(indxbuff);
            //console.log(arrsensor);
        }

        BufferSeleccio(indxbuff, arrsensor);
    });
};

//Funció que mira si ja s'ha enviat Downlink o si el led està ACTIVAT
function BufferSeleccio(index, llista){
    let arrcotxe = []; // Array cotxes
    let arrdwnlk = []; // Array leds
    let arrdate = []; // Array dates dels cotxes

    //Creem les llistes per buscar temps i preparar downlinks
    for (let i = 0; i < index; i++) {
        arrcotxe[i] = llista[i].DevEUI_cotxe;
        arrdwnlk[i] = llista[i].DevEUI_led;
    }
}

```

```

};

//console.log(arrcotxe);

// Un cop sabem la llista dels sensors busquem quins han sobrepassat el
temps

for (let i = 0; i < index; i++) {
    let eui = arrcotxe[i];

    const sql = 'SELECT Data FROM Sensor_Cotxe WHERE DevEUI = ' +
mysql.escape(eui) + ' AND Parking_status = 1 ORDER BY Data DESC LIMIT 1';

    connection.query(sql, (error, result) =>{
        if (error) throw error;
        if (result.length > 0){
            //let index = result.length;
            arrdate[i] = result[0].Data;

            const act = new Date(); // Hora actual
            const t_sens = new Date(arrdate[i]); // Hora del sensor

            //console.log((act - t_sens));
            let minuts = (act - t_sens)/60000;

            //Mirem si han passat els minuts
            if (minuts > 1){
                console.log('Downlink enviat');
                //console.log(arrcotxe[i]);
                //console.log(arrdwnlk[i]);

                const sended = 'AQ==';

                axios({
                    headers: {'Authorization': 'Bearer',
                        'Content-Type': 'application/json',
                        'User-Agent': 'proves-cotxe/v3'},
                    method: 'post',
                    url:
'https://eu1.cloud.thethings.network/api/v3/as/applications/proves-
cotxe/webhooks/api-webhook-udg/devices/' + arrdwnlk[i] + '/down/push',
                    data: {"downlinks": [{
                        "frm_payload": sended,

```



```

        "f_port": 15,

        "priority": "NORMAL"
    }]
}

});

//Finalment actualitzem el valor Downlink

const sql = 'UPDATE Gestio_Cotxe SET Downlinks_sent = 1 WHERE
DevEUI_led =' + mysql.escape(arrdwlk[i]);

//Guardem a la DB Gestio_Cotxe
connection.query(sql, error =>{
    if (error) throw error;

    console.log('Downlink pel actuador ' + arrdwlk[i] + '
actualitzat');
});

});

}

});

}

/*const date = new Date();
console.log(date -Date.now());*/

};

// Funció que revisa i crea els downlinks de DESACTIVACIÓ
//Aquesta funció no contempla el cas PS = 0 AND D_s = 1 AND E_1 = 0 Espera que
el LED dongui resposta E_1 = 1
function RevDevEUIdesac(){
    let indxbuff = 0;
    let arrsensor = [];

    const sql = 'SELECT * FROM Gestio_Cotxe WHERE Parking_Status = 0 AND
Downlinks_sent = 1 AND Estat_led = 1 ';

    connection.query(sql, (error, result) =>{
        if (error) throw error;

```

```

    if (result.length > 0){
        indxbuff = result.length;
        arrsensor = result;
        //console.log(arrsensor[0]);
        //console.log(indxbuff);
        //console.log(arrsensor);
    }

    BufferSeleccioDesac(indxbuff, arrsensor);
});
};

//Funció eliminatòria i downlink desactivació
function BufferSeleccioDesac(index, llista){
    let arrcotxe = []; // Array cotxes i leds associats
    let arrdwnlnk = []; // Array leds
    //let arrdate = []; // Array dates dels cotxes
    //Creem les llistes per buscar temps i preparar downlinks
    for (let i = 0; i < index; i++) {
        arrcotxe[i] = llista[i].DevEUI_cotxe;
        arrdwnlnk[i] = llista[i].DevEUI_led;
        const sended = 'AA==';
        axios({
            headers: {'Authorization': 'Bearer',
                'Content-Type': 'application/json',
                'User-Agent': 'proves-cotxe/v3'},
            method: 'post',
            url: 'https://eu1.cloud.thethings.network/api/v3/as/applications/proves-cotxe/webhooks/api-webhook-udg/devices/' + arrdwnlnk[i] + '/down/push',
            data: {"downlinks": [{
                "frm_payload": sended,
                "f_port": 15,
                "priority": "NORMAL"
            }]}
        });

        //Finalment actualitzem el valor Downlink
        const sql = 'UPDATE Gestio_Cotxe SET Downlinks_sent = 0 WHERE DevEUI_led = ' + mysql.escape(arrdwnlnk[i]);
    }
}

```

```

        //Guardem a la DB Gestio_Cotxe
        connection.query(sql, error =>{
            if (error) throw error;

            console.log('Downlink Desactivació per l"actuador ' + arrdwlnk[i] + '
actualitzat');

            });

        };

};

//Funció integrada a RevDevEUI que envia els downlinks d'activació i
RevDevEUIdesac per desactivacions

RevDevEUI();

RevDevEUIdesac();

//-----//

// Errors no descrits a l'aplicació
app.use((error, req, res, next) => {
    res.status(500)
    res.send({error: error})
    console.error(error.stack)
    next(error)
})

// Missatge d'avís, per la connexió del port
app.listen(port, () =>
    console.log(`Aplicació executada a la direcció http://localhost:${port}`)
);

```