**PAYGTL\_LORA\_BLE**

**Common URL:** Address:PortNumber/PAYGTL\_LORA\_BLE

**url:** common URL/

**Method:**

**Request:**

**Response:**

**1. Login**

a) for customer login

**url:** common URL/login

**Method:** Post

**Request:**

{

"userID" : "HAN0012",

"password" : "Admin",

"source" : "mobile"

}

**Response:**

{

    "result": "Success",

    "Message": "Successfully Logged In",

    "userDetails": {

        "roleID": 3,

        "blockID": 11,

        "email": "kvk9889@gmail.com",

        "mobileNumber": "9398348954",

        "communityID": 5,

        "userName": "K VIMAL KUMAR",

        "ID": 19,

        "customerID": 12,

        "CRNNumber": "HAN0012",

        "communityName": "Finaltesting",

        "pendingCommandType": -1, // if any command is pending its respective id will be sent here else -1

        "pendingTransactionID": -1, // if any command is pending its respective transactionid (else -1)

along with Extra Parameter with key as “dataFrame” & value of

String type ”0A0C007b32020000230000002817” will be sent which

should be directly sent to the device after connecting via Bluetooth.

        "blockName": "finaltest",

        "meterID": "70b3d5f83000157b"

    }

}

b) for technical support login

**Request:**

{

"userID" : "Techsupport",

"password" : "Admin",

"source" : "mobile"

}

**Response:**

{

    "result": "Success",

    "Message": "Successfully Logged In"

}

**2. ReadMeter**

**url:** common URL/datafrommobile

**Method:** Post

**Request:**

{

"source" : "mobile",

"meterID" : "70b3d5f830004f56",

"reading" : 0.0,

"meterType" : 3,

"balance" : 30,

"tariffAmount" : 50,

"emergencyCredit" : 20,

"batteryVoltage" : 3.59,

"tamperStatus" : 0, // 0 or 1 or 2 or 3

“tamperTimeStamp” : “2020-04-20 13:34:40”

“doorOpenTimeStamp” : “2020-04-20 13:34:40”

"valveStatus" : 0, // 0 or 1

"creditStatus" : 0, // 0 or 1 based on formula given in Notes point 3

"lowBattery" : 0,

"minutes" : 0,

"vacation" : 0,

"timeStamp" : "2020-04-20 13:34:40"

}

**Response:**

{

    "result": "Success",

    "Message": "Data Inserted Successfully"

}

**3. Fetch Customer Details for doing topup**

**url:** common URL/topupdetails/{CRNNumber}

**Method:** GET

**Response:**

{

    "topupdetails": {

        "meterID": "70b3d5f83000157b",

        "currentBalance": 110.0,

        "IoTTimeStamp": "05:15:2020,23:49:50",

        "tariffName": "Tariff1",

        "emergencyCredit": 50.0,

        "alarmCredit": 15.0,

        "tariff": 30.0,

        "tariffID": 1,

        "reconnectionCharges": 0,

        "fixedCharges": 25,

        "noOfMonths": 1

    }

}

**4. Topup**

a. to retrieve list of Topups

**url:** common URL/status/3/{CRNNumber}/0

**Method:** GET

**Response:**

{

    "transactionID": 0,

    "data": [

        {

            "transactionID": 23,

            "communityName": "Finaltesting",

            "blockName": "finaltest",

            "houseNumber": "49/38",

            "CRNNumber": "HAN0012",

            "firstName": "K VIMAL",

            "lastName": "KUMAR",

            "meterID": "70b3d5f83000157b",

            "amount": "200",

            "emergencyCredit": "50.0",

            "alarmCredit": "15.0",

            "transactionDate": "2020-05-15 20:45:14",

            "Status": "Failed",

"razorPayOrderID": "order\_EyuowyRemKydQ8",

            "razorPayPaymentID": "pay\_Eyup5nKuxGRwVG",

            "razorPayRefundID": "---",

            "RazorPayRefundStatus": "---",

            "modeOfPayment": "Online",

            "paymentStatus": "PAID",

            "modeOfPayment": "Cash",

            "transactedByUserName": "finaltest",

            "transactedByRoleDescription": "Admin"

        }

    ]

}

b. to add a topup

**url:** common URL/topup

**Method:** POST

**Request:**

{

    "source" : "mobile", // if connected with device send source as ‘mobile’ else ‘web’

    "CRNNumber" : "HAN0014",

    "meterID" : "70b3d5f830004fb5",

    "amount" : 300,

    "modeOfPayment" : "Online",

    "transactedByID" : 1, //this is the value of ID param received at the time of login (highlighted

in blue color in page 1)

    "transactedByRoleID" : 3 //fixed value

}

**Response:**

{

    "result": "Success",

    "Message": "Order Created Successfully. Proceed to CheckOut",

    "checkoutDetails": {

        "key": "rzp\_test\_PK0HJHPshOFmX5",

        "amount": 30000,

        "currency": "INR",

        "order\_id": "order\_F0WB07fgLuIhYO",

        "buttonText": "Proceed to Pay With Razorpay",

        "name": "HANBIT",

        "description": "Recharge of INR 300/- for CRN: HAN0014.",

        "image": "http://localhost:8080/PAYGTL\_LORA\_BLE/common/images/hanbit1.png",

        "prefill": {

            "name": "Mamatha Hanbit",

            "email": "vmamatha4@gmail.com",

            "contact": "7659934563"

        },

        "theme": {

            "color": "#125e8c"

        },

        "transactionID": 53

    },

    "paymentMode": "Online"

}

c. to proceed to checkout

**url:** common URL/checkout

**Method:** POST

**Request:**

{

"razorpay\_order\_id" : " order\_EyuowyRemKydQ8",

"razorpay\_payment\_id" : " pay\_Eyup5nKuxGRwVG ",

"razorpay\_signature" : "bb68d77f48f27ad1c90df76ef6f0838e35d0a933cbc3781eb06e5ff67d626aad ",

"transactionID": 23 // received in the response json from the above api

// if payment fails/ razorpay error occurs u need to append the error in the same json. The error will be the same format sent by razorpay. Kindly refer to the error entity in razorpay doc website.

}

**Response:**

{

    "result": "Success",

    "Message": "Payment Captured Successfully",

}

**5. Vacation**

a. to retrieve list of vacations

**url:** common URL/vacation/3/{CRNNumber}/0

**Method:** GET

**Response:**

{

    "data": [

{

            "communityName": "Finaltesting",

            "blockName": "finaltest",

            "houseNumber": "49/38",

            "CRNNumber": "HAN0012",

            "firstName": "K VIMAL",

            "lastName": "KUMAR",

            "meterID": "70b3d5f83000157b",

            "startDate": "05:15:2020,23:46:00",

            "endDate": "05:15:2020,23:50:00",

            "startDateForEdit": "2020-05-15 23:46:00",

            "endDateForEdit": "2020-05-15 23:50:00",

            "registeredDate": "05:15:2020,23:43:50",

            "vacationID": 15,

            "vacationName": "demo",

            "mode": "add",

            "status": "Failed"

        },

        {

            "communityName": "Finaltesting",

            "blockName": "finaltest",

            "houseNumber": "49/38",

            "CRNNumber": "HAN0012",

            "firstName": "K VIMAL",

            "lastName": "KUMAR",

            "meterID": "70b3d5f83000157b",

            "startDate": "05:15:2020,23:40:00",

            "endDate": "05:15:2020,23:45:00",

            "startDateForEdit": "2020-05-15 23:40:00",

            "endDateForEdit": "2020-05-15 23:45:00",

            "registeredDate": "05:15:2020,23:37:52",

            "vacationID": 14,

            "vacationName": "finals",

            "mode": "add",

            "status": "Failed"

        }

    ]

}

b. to add a vacation

**url:** common URL/vacation/add

**Method:** POST

**Request:**

{

"source" : "mobile", // if connected with device send source as ‘mobile’ else ‘web’

"CRNNumber" : ”HAN0006”,

"vacationName": "out of Station",

"startDateTime" : "2020-04-21 13:00",

"endDateTime" : "2020-04-21 13:15",

"startDay" : 2, // day numbering starts with 0 (i.e Sun = 0) and ends with 6 (i.e. Sat = 6)

"endDay" : 2, // day numbering starts with 0 (i.e Sun = 0) and ends with 6 (i.e. Sat = 6)

"status" : 2 // if connected with device send Status as 2 else 0

}

**Response:**

{

    "result": "Success" ,

"Message": "Vacation Request Inserted Successfully"

}

b. to edit a vacation

**url:** common URL/vacation/edit/{vacationID}

**Method:** POST

**Request:**

{

"source" : "mobile", // if connected with device send source as ‘mobile’ else ‘web’

"CRNNumber" : ”HAN0006”,

"vacationName": "out of Station",

"startDateTime" : "2020-04-21 13:00",

"endDateTime" : "2020-04-21 13:15",

"startDay" : 2, // day numbering starts with 0 (i.e Sun = 0) and ends with 6 (i.e. Sat = 6)

"endDay" : 2, // day numbering starts with 0 (i.e Sun = 0) and ends with 6 (i.e. Sat = 6)

"status" : 2 // if connected with device send Status as 2 else 0

}

**Response:**

{

    "result": "Success",

"Message": " Vacation Update request Inserted Successfully "

}

b. to delete a vacation

**url:** common URL/vacation/delete/{source}/{vacationID}

// if connected with device send source as ‘mobile’ else ‘web’

**Method:** POST

**Response:**

{

    "result":  "Success",

"Message":  “Vacation Delete request Inserted Successfully"

}

**6. User Consumption Reports**

**url:** common URL/userconsumptionreports

**Method:** POST

**Request:**

{

    "CRNNumber" : "HAN0006",

    "fromDate" : "2020-04-29 16:40:30",

    "toDate" : "2020-05-30 23:40:30"

}

**Response:**

{

    "reading": 0.0,

    "balance": 0.0,

    "battery": 0.0,

    "tariff": 0.0,

    "alarmCredit": 0.0,

    "emergencyCredit": 0.0,

    "data": [

        {

            "CRNNumber": "HAN0006",

            "meterID": "70b3d5f830000a68",

            "reading": 140.0,

            "balance": 1910.0,

            "battery": 2.24,

            "tariff": 10.0,

            "alarmCredit": 0.0,

            "emergencyCredit": 0.0,

            "dateTime": "05:02:2020,22:12:36"

        },

        {

            "CRNNumber": "HAN0006",

            "meterID": "70b3d5f830000a68",

            "reading": 142.0,

            "balance": 1910.0,

            "battery": 2.22,

            "tariff": 10.0,

            "alarmCredit": 0.0,

            "emergencyCredit": 0.0,

            "dateTime": "05:02:2020,22:19:42"

        }

    ]

}

**7. Feedback/Complaints**

**url:** common URL/feedback/add

**Method:** POST

**Request:**

{

"CRNNumber": "HAN0012",

"feedback": "test1",

"description" : "testing1" // optional

}

**Response:**

{

    "result": "Success",

    "Message": "Feedback/Complaint Submitted Successfully"

}

**Note:**

List of feedbacks/complaints to be provided in the drop down:

1. Bill in Time Not Updated
2. InSuffient Pressure
3. Leakages
4. Meter Not Working
5. Meter Run Fast
6. Mismatch B/W Mechanical & Digital Reading
7. New Connection Request
8. Others
9. Failure

List of Command codes:

1. 40 = Valve open
2. 0 = valve close
3. 3 = Clear Meter
4. 1 = Clear Tamper
5. 5 = Set RTC
6. 6 = Set Meter Index
7. 10 = Set Tariff

Formula to Check credit status:

The balance in the meter should not be less than the tariff assigned to the meter multiplied by specific constant (multiplier)\*

i.e. if tariff amount in the meter is 10 and available balance be 30 and specific constant (multiplier) be 3 then the credit status will be 1 (as 3 (specific constant or multiplier) X 10 (tariff amount) <= 30 (available balance)) else 0.

\*The specific constant or multiplier should be editable