Application Description

Basically, the Android application needs to render sections and commands based on the actions retrieved from Permissions endpoint. Each command generates hexadecimal payload that the application have to send to a selected Bluetooth device. So, the application needs to have a setting section where save a selected Bluetooth device.

Possible commands are retrieved from Permissions endpoint. Each user has different permissions, so each user could have different available commands.
So, overall, first action that the application have to do, it's the login by using Login endpoint.
If username and password sent to login endpoint are correct, the JSON response contains a Token that needs to be stored somewhere in the application.

After the login gets success, the application has to request permissions to the Permissions endpoint. The JSON response contains some actions. Each action is a group of commands. In the application groups can be arranged in tabs or menu items or what you think is better to arrange them like sections.
Each command has a "label", a hexadecimal "payload" and, optionally, a list of "parameters". If there are parameters, payload contains many {HOLDERS} as parameters.
Each parameter has a type and, if the it requires more fields, it can have "label", "value" and/or "required" fields.
"label" is, of course, the label of the field, "required" is a regular expression that validate the inserted data in the field and "value" means as the inserted data in the field has to be transformed when replaced in his payload {HOLDER}. Each {HOLDER} has in the parenthesis his corresponding parameter key.

The Permissions endpoint should be queried every 3 hours, to be sure permissions haven't changed or session isn't expired. When session is expired or user is not authorized, http code is different than 200. If user is logged in and session expires, the application has to logout him automatically.

Of course, the application needs to provide manual logout too.

End Points

Type: Login

URL: http://apm.integraaposta.com/gestione/api/login

Request-Headers:

    Method: POST
    Content-Type: application/x-www-form-urlencoded
    Request-Body: u={{USERNAME}}&p={{PASSWORD}}&lat={{LATITUDE}}&lng={{LONGITUDE}}

Request-Parameters:

    {{USERNAME}}=test@integraa.it
    {{PASSWORD}}=Test123!
    {{LATITUDE}}=[latitude when send login access]
    {{LONGITUDE}}=[longitude when send login access]

Response-Headers:

    Content-Type: application/json

Response -Parameters:

    Token(to store somewhere)

Type: Permissions

URL: http://apm.integraaposta.com/gestione/api/waterPermissions

Request- Headers:

    Method: GET
    Token: {{TOKEN}}

Request-Parameters:

    {{TOKEN}}=Retrievied from Login Response

Response-Headers:

    Content-Type: application/json

Response -Parameters:

```
{
    actions: {
        GROUP_KEY: {
            label: “GROUP_LABEL”,
            items: {
                COMMAND_KEY: {
                    label: “COMMAND_LABEL”,
                    payload: “COMMAND_PAYLOAD”
                    parameters: {
                        PARAMENT_KEY: {
                            label: “PARAMETER_LABEL”,
                            type: “PARAMENTER_TYPE”,
                            value: “PARAMENTER_VALUE”,
                            required: “PARAMENTER_REGEXP”
                        }
                    }
                }
            }
        }
    }
}
```

Command Parameter Type

Type: text

Application needs to render a text field where user can type some text

Type: int

Application needs to render a text field where user can type only integer numbers and, if min/max are present, numbers have to be limited in according them.

Type: checksum

Application doesn't need to render anything but his {HOLDER} need to be replaced with his corresponding checksum. The checksum algorithm is the "CheckSum8 Modulo 256", check site like https://www.scadacore.com/tools/programming-calculators/online-checksum-calculator/. The checksum is calculated on all bytes before his {HOLDER}. For example, checksum for "6810FFFFFFFF0011110404A0170055{CHK}16" is "AA", so final payload is "6810FFFFFFFF0011110404A0170055AA16".

Command Parameter Value

Value: equal

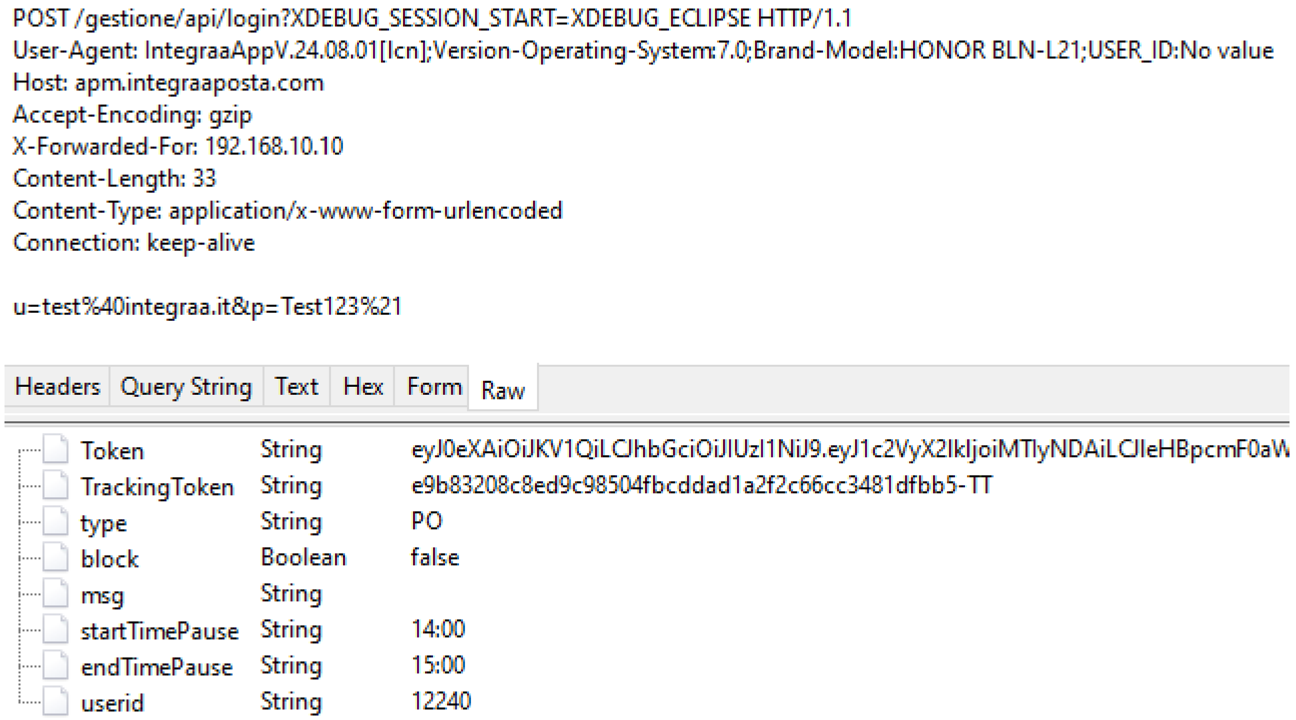
Inserted data needs to be replaced in the payload as well as the user inserted it in the field

Value: IP

Inserted data needs to be replaced as shown below (4 bytes length).
Inserted IP: {Int_PART1}.{Int_PART2}.{Int_PART3}.{Int_PART4}
Replaced {HOLDER}: {IntToHex_PART1}{IntToHex_PART2}{IntToHex_PART3}{IntToHex_PART4}
For example, 021.042.063.084 is converted to 152A3F54

Value: int4

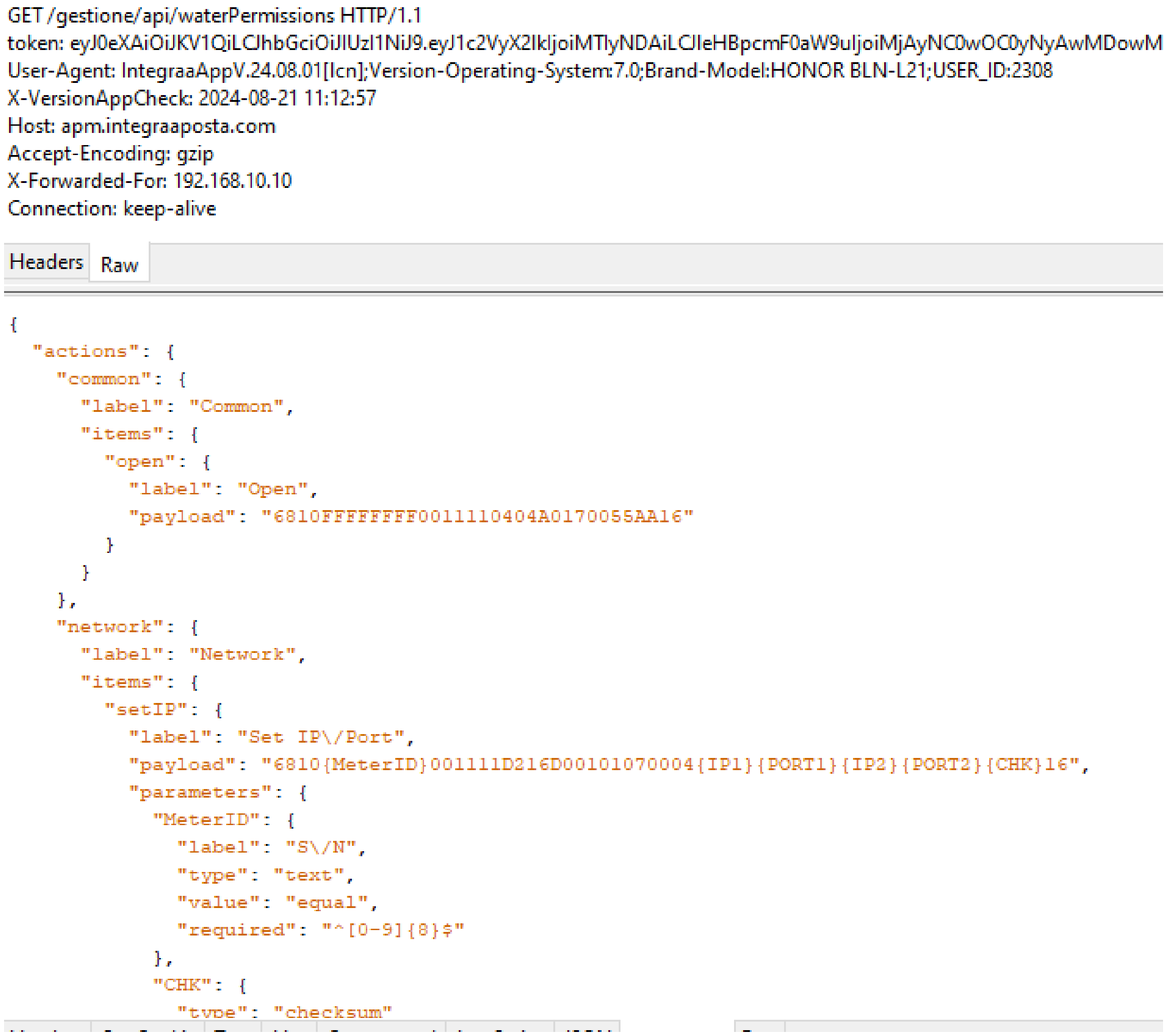
Inserted integer needs to be replaced as fixed 4 bytes length.
For example, 5013 is converted to 00001395



```
POST /gestione/api/login?XDEBUG_SESSION_START=XDEBUG_ECLIPSE HTTP/1.1
User-Agent: IntegraaAppV.24.08.01[lcn];Version-Operating-System:7.0;Brand-Model:HONOR BLN-L21;USER_ID:No value
Host: apm.integraaposta.com
Accept-Encoding: gzip
X-Forwarded-For: 192.168.10.10
Content-Length: 33
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

u=test%40integraa.it&p=Test123%21
```

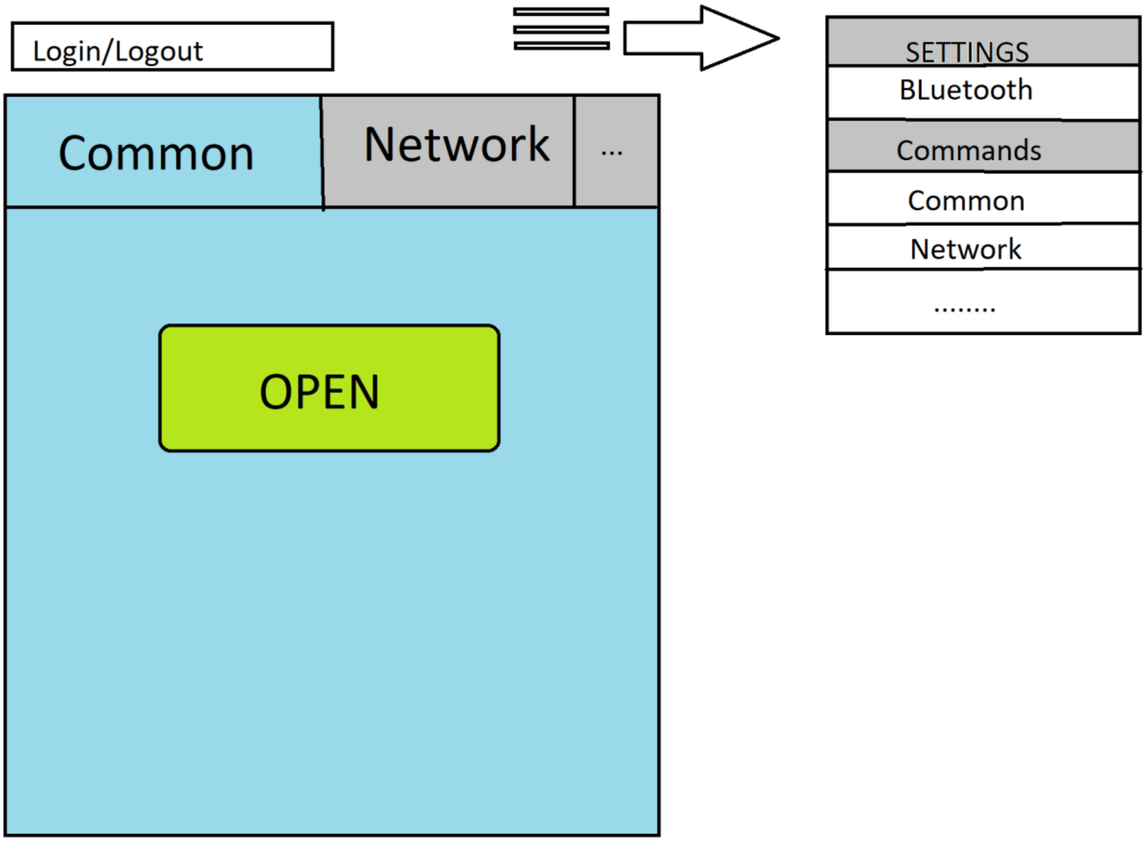
```
Token           String    eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiMTIyNDAiLCJleHBpcmF0aW
TrackingToken   String    e9b83208c8ed9c98504fbcddad1a2f2c66cc3481dfbb5-TT
type            String    PO
block           Boolean   false
msg             String
startTimePause  String    14:00
endTimePause    String    15:00
userid          String    12240
```



```
GET /gestione/api/waterPermissions HTTP/1.1
token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoiMTIyNDAiLCJleHBpcmF0aW9uIjoiMjAyNC0wOC0yNyAwMDowM
User-Agent: IntegraaAppV.24.08.01[lcn];Version-Operating-System:7.0;Brand-Model:HONOR BLN-L21;USER_ID:2308
X-VersionAppCheck: 2024-08-21 11:12:57
Host: apm.integraaposta.com
Accept-Encoding: gzip
X-Forwarded-For: 192.168.10.10
Connection: keep-alive
```

```
{
  "actions": {
    "common": {
      "label": "Common",
      "items": {
        "open": {
          "label": "Open",
          "payload": "6810FFFFFFFF0011110404A0170055AA16"
        }
      }
    },
    "network": {
      "label": "Network",
      "items": {
        "setIP": {
          "label": "Set IP\/Port",
          "payload": "6810{MeterID}001111D216D00101070004{IP1}{PORT1}{IP2}{PORT2}{CHK}16",
          "parameters": {
            "MeterID": {
              "label": "S\/N",
              "type": "text",
              "value": "equal",
              "required": "^[0-9]{8}$"
            },
            "CHK": {
              "type": "checksum"
```

Draft Application Picture


Login/Logout
Common
Network
...
OPEN
SETTINGS
BLuetooth
Commands
Common
Network
....