EG71 API Documentation

CGI Request Common Format

```JSON
{
  "execute": 1,
  "core": "<module_name>",
  "function": "get|set|del",
  "values": [{ "base": "<config_section>" }],
  "id": <sequence_number>
}
```

CGI Response Common Format

```JSON
{
  "id": <sequence_number>,
  "model": "EG71",
  "pn": "<part_number>",
  "oem": "0000",
  "rtver": "71.0.0.2",
  "status": 0,
  "result": [{ "get": [...], "grey": 0 }]
}
```

REST API Response Common Format

```JSON
{
  "errCode": 0,
  "errMsg": "success",
```

```
  ...business data
}
```

⚠ CGI Rate Limiting

The CGI backend has rate limiting — **consecutive requests must be spaced ≥500ms apart**, otherwise a 503 Service Temporarily Unavailable is returned\.

Common Requests on Page Initialization \(Triggered on Every Page Switch\)

Request	Description
`POST /islogin`	Check login status, return device info
`POST /cgi` core=yruo_usermanagement base=security	Get user permissions
`POST /cgi` core=yruo_usermanagement base=check_pass	Check if default password
`POST /cgi` core=yruo_wizard base=yruo_wizard	Wizard status
`POST /cgi` core=yruo_system base=general	System base configuration
`GET /api/general-info/interface/with-access-info`	Get access interface info
`POST /cgi` core=yruo_status base=dashboard	Dashboard status

1\. Authentication Module \(Login\)

1\.1 Check Login Status

```Plain Text
POST /islogin
Authorization: Bearer login=<user>;<hash>  (include when token available)
```

Not logged in response \(status: \-2\):

```JSON
{
  "id": -1, "model": "EG71", "rtver": "71.0.0.2", "status": -2,
  "result": [{ "login": "false", "ysrole": 0, "timeout": 0, "upgrade_error":
0, "lora_port": 443 }]
}
```

Logged in response \(status: 0\):

```JSON
{
  "id": -1, "model": "EG71", "pn": "L08GL3100000CN0010310100", "sms": 1,
"oem": "0000", "rtver": "71.0.0.2", "status": 0,
  "result": [{ "login": "true", "ysrole": 4, "timeout": 3600,
"upgrade_error": 0, "lora_port": 443 }]
}
```

- `ysrole`: 4 = Admin

- `timeout`: 3600 = Session timeout \(seconds\)

1\.2 CGI Login \(Obtain td Token\)

```Plain Text
POST /cgi
Content-Type: application/json
```

```JSON
```

```
{
  "execute": 1,
  "core": "user",
  "function": "login",
  "values": [{
    "username": "admin",
    "password": "<AES-128-CBC encrypted, Base64>"
  }],
  "id": 5
}
```

Password encryption parameters:

- Algorithm: AES\-128\-CBC

- Key: `4829173051647823`

- IV: `7603912845091736`

- Padding: PKCS7

- Output: Base64

Success response \(status: 0\):

```JSON
{
  "status": 0,
  "result": [{
    "ysrole": 4,
    "ystimeout": 3600,
    "ysexpires": 3599,
    "username": "admin",
    "td": "f6ad9ca5703b0bc84546280d1e38902f"
  }]
}
```

- `td` value is the CGI token hash, used in subsequent requests as `Bearer login=admin;<td>`

Failure response \(status: \-2\):

```JSON
{
  "status": -2,
  "result": [{ "locktime": 0, "chance": 4 }]
}
```

- `chance` decreasing = Remaining attempts

1\.3 JWT Login \(REST API Token\)

```Plain Text
POST /api/internal/login
Content-Type: application/json
```

```JSON
{
  "username": "admin",
  "password": "<AES-128-CBC encrypted, Base64>"
}
```

Success response:

```JSON
{
  "jwt":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJsb3JhLWFwcC1zZXJ2ZXIiLCJleHAiOjE3ODEzNDQ5MTEsImlzcyI6ImxvcmEtYXBwLXNlcnZlciIsIm5iZiI6MTc4MTI1ODUxMSwic3ViIjoidXNlciIsInVzZXJuYW1lIjoiYWRtaW4ifQ.OUq26C9fvFh7Bp3TBC3PfkW12iRNPecefNymLV8WErs"
}
```

- JWT payload: `aud=lora-app-server`, `iss=lora-app-server`, `sub=user`, `username=admin`

- Validity period: \~24h

1\.4 Logout

```Plain Text
POST /logout
Authorization: Bearer login=admin;<td>
```

1\.5 Token Storage \(localStorage\)

Key	Value	Purpose
`token`	JWT string	`/api/*` request authentication
`cgi_token`	`login=admin;<td>`	`/cgi` request authentication
`EG7X.lang`	`{"key":"EG7X.lang","value":"EN"}`	UI language
`EG7X.theme`	`{"key":"EG7X.theme","value":"light"}`	Theme

2\. Data Acquisition

Route: `/data-services/equipment-data`

2\.1 Get Device List

```Plain Text
POST /api/dsdevices/device
Authorization: Bearer <JWT>
```

```JSON
{
  "protocolType": [],
  "deviceType": [],
  "deviceNameOrEui": "",
  "status": "",
  "pageSize": 10,
  "page": 1
}
```

Filter parameters:

- `protocolType`: Protocol type array \(`lorawan`, `bacnet ip`, `bacnet mstp`, `modbus tcp`, `modbus rtu`, `knx tp`\)

- `deviceType`: Device model array

- `deviceNameOrEui`: Name or EUI search

- `status`: Status filter \(`online`, `offline`, `never_online`\)

Response:

```JSON
{
  "devTotalCount": "14",
  "deviceResult": [{
    "id": "39",
    "name": "EM300-TH-868M",
    "deviceType": "EM300-TH",
    "protocolType": "lorawan",
    "signal": "0",
```

```
    "updatedAt": "-",
    "status": "never_online",
    "objectCount": "30",
    "payloadCodecID": "744",
    "accessNetworkId": "1",
    "accessNetworkName": "LoRaWAN",
    "snr": 0, "rssi": 0, "sf": 0,
    "deviceEui": "24e124136f490557"
  }]
}
```

2\.2 Add LoRaWAN Device

```Plain Text
POST /api/urdevices
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "Test-Device",
  "protocolType": "lorawan",
  "payloadCodecID": "707",
  "description": "",
  "devEUI": "24e1247250000088",
  "profileID": "24415a71-b1de-45c0-ac20-9a2590f35d17",
  "fPort": 1,
  "appKey": "2B7E151628AED2A6ABF7158809CF4F3C",
  "devAddr": "",
  "appSKey": "",
  "nwkSKey": "",
  "fCntUp": 0,
  "fCntDown": 0,
  "timeout": 1440,
  "skipFCntCheck": true,
  "isSave": true
}
```

Response:

```JSON
{ "errCode": 200, "errMsg": "", "deviceId": "41" }
```

Field descriptions:

- `isSave`: `true`=save, `false`=validate only \(Next Step pre\-check\)

- `payloadCodecID`: Codec ID for the device model \(obtained from `/api/payloadcodecs-short`\)

- `profileID`: LoRaWAN Profile ID \(obtained from `/api/urprofiles`\)

- `devEUI` prefix must match the Codec's `devEUIPrefix`

- OTAA devices fill `appKey`; ABP devices fill `nwkSKey` \+ `appSKey` \+ `devAddr`

2\.3 Add BACnet/Modbus/KNX Device

BACnet/IP and BACnet MS/TP

```Plain Text
POST /api/dsdevices/device/bacnet/create
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "TestBACnet",
  "deviceId": "12345",
  "payloadCodecID": "0",
  "protocolType": "bacnet ip",
  "accessNetworkId": "3",
  "description": "",
  "isSave": true
```

```
}
```

- BACnet MS/TP: Change `protocolType` to `"bacnet mstp"`, `accessNetworkId` points to RS485 access network

- `deviceId` = BACnet Instance Number

- Response: `{"errCode":0,"errMsg":"success","deviceId":"43"}`

Modbus TCP / RTU / RTU over TCP

```Plain Text
POST /api/dsdevices/device/modbus/create
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "TestModbus",
  "payloadCodecID": "0",
  "protocolType": "modbus tcp",
  "accessNetworkId": "8",
  "description": "",
  "ipAddress": "192.168.1.100",
  "ipPort": 502,
  "slaveId": 1,
  "isSave": true
}
```

- Modbus RTU: `protocolType`=`"modbus rtu"`, **no** `ipAddress`/`ipPort`, `accessNetworkId` points to RS485

- Modbus RTU over TCP: `protocolType`=`"modbus rtu over tcp"`, has `ipAddress`/`ipPort`

- Response: `{"errCode":0,"errMsg":"success","deviceId":"44"}`

KNX/TP

```Plain Text
POST /api/dsdevices/device/knx/create
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "TestKNX",
  "protocolType": "knx tp",
  "payloadCodecID": "0",
  "accessNetworkId": "6",
  "description": "",
  "mac": "1.1.100",
  "isSave": true
}
```

- `mac` = KNX physical address \(e\.g\., `1.1.100`\)

2\.3b Update BACnet/Modbus/KNX Device

```Plain Text
PUT /api/dsdevices/device/bacnet/update/{id}    (BACnet)
PUT /api/dsdevices/device/modbus/update/{id}    (Modbus)
PUT /api/dsdevices/device/knx/update/{id}       (KNX)
```

2\.4 Update Device

```Plain Text
PUT /api/urdevices/{devEUI}
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "Test-Device",
  "protocolType": "lorawan",
  "payloadCodecID": "707",
  "description": "Updated description",
  "devEUI": "24e1247250000088",
  "profileID": "24415a71-b1de-45c0-ac20-9a2590f35d17",
  "fPort": 10,
  "appKey": "2B7E151628AED2A6ABF7158809CF4F3C",
  "devAddr": "", "appSKey": "", "nwkSKey": "",
  "fCntUp": 0, "fCntDown": 0,
  "timeout": 1440,
  "skipFCntCheck": false,
  "isSave": true
}
```

Response: `{"code": 200, "error": ""}`

2\.5 Delete Device

```Plain Text
POST /api/dsdevices/device/delete
Authorization: Bearer <JWT>
```

```JSON
{
  "deviceIds": ["41"]
}
```

2\.6 Get Device Object List

```Plain Text
```

```
GET /api/dsdevices/objects/{deviceId}?pageSize=10&page=1
Authorization: Bearer <JWT>
```

Response:

```JSON
{
  "objectsResult": [
    {
      "id": "361",
      "name": "Battery",
      "currentValue": "-",
      "gradient": 1,
      "offset": 0,
      "unit": "%",
      "updateAt": "-",
      "updateTimes": "0",
      "associatedObjects": [],
      "enable": true,
      "linearEnable": false,
      "type": "NUMBER",
      "usedByForward": true
    },
    {
      "id": "362",
      "name": "Clear History",
      "type": "BOOL",
      "usedByForward": false
    }
  ],
  "dsDeviceTotalCount": "35"
}
```

2\.7b Add BACnet/Modbus/KNX Object

Modbus Object

```Plain Text
POST /api/dsdevices/objects/modbus/pc-create
Authorization: Bearer <JWT>
```

```JSON
{
  "deviceId": 44,
  "objectName": "Temperature",
  "registerType": "holding_register",
  "registerAddress": 40001,
  "registerQuantity": 1,
  "dataFormat": "uint16",
  "objectDescription": "Test register",
  "collectionInterval": 60
}
```

- `registerType`: `holding_register`, `input_register`, `coil`, `discrete`

- `dataFormat` \(holding/input\): `uint16`, `int16_ab`, `int16_ba`, `uint32_abcd`, `int32_abcd`, `float32`, `float64`, etc\.

- `dataFormat` \(coil\): `bool`; \(discrete\): `flag`

BACnet Object

```Plain Text
POST /api/dsdevices/objects/bacnet/pc-create
Authorization: Bearer <JWT>
```

```JSON
{
  "deviceId": 43,
  "objectName": "RoomTemp",
  "objectType": "analog_input",
  "objectInstanceNr": 100,
  "interval": 60,
  "unit": "",
  "unitId": -1
```

```
}
```

- `objectType`: `analog_input`, `analog_output`, `analog_value`, `binary_input`, `binary_output`, `binary_value`, `multistate_input`, `multistate_output`, `multistate_value`

KNX Object

```Plain Text
POST /api/dsdevices/objects/knx/pc-create
Authorization: Bearer <JWT>
```

```JSON
{
  "deviceId": 48,
  "objectName": "RoomTemp",
  "dataPoint": "9.001",
  "groupAddress": "1/1/1",
  "interval": 600,
  "flag": 1,
  "unit": "",
  "unitId": -1
}
```

- `dataPoint`: KNX DPT \(e\.g\., `9.001`=temperature, `1.001`=switch\)

- `groupAddress`: KNX group address \(e\.g\., `1/1/1`\)

Protocol Object List Query

Protocol	Method	URL
Modbus	GET	`/api/dsdevices/objects/modbus/{deviceId}?pageSize=10&page=1`
BACnet	GET	`/api/dsdevices/objects/bacnet/{deviceId}?pageSize=10&page=1`

|KNX|POST|`/api/dsdevices/objects/knx` \(body:
`{"deviceId":"32","pageSize":9999,"page":1}`\)|

Protocol Object Update

Protocol	Method	URL
Modbus	PUT	`/api/dsdevices/objects/modbus/simple/{objectId}`
BACnet	PUT	`/api/dsdevices/objects/bacnet/simple/{objectId}`
KNX	PUT	`/api/dsdevices/objects/knx/simple/{objectId}`

> ⚠ Field names are strictly case\-sensitive: `deviceId` \(not
`deviceID`\), `objectIdList` \(not `objects`\)
>
>

```Plain Text
POST /api/dsdevices/objects/create
Authorization: Bearer <JWT>
```

```JSON
{
  "deviceId": "39",
  "objectIdList": ["36185", "36196"]
}
```

Response: `{"errCode":0,"errMsg":""}`

- `deviceId`: Device ID \(string\)

- `objectIdList`: Object ID string array \(obtained from
`/api/dsdevices/objects/addlist/{deviceId}`\)

2\.8 Get Addable Object List

```Plain Text
GET /api/dsdevices/objects/addlist/{deviceId}
Authorization: Bearer <JWT>
```

Response:

```JSON
{
  "result": [
    { "id": "36185", "name": "Battery", "select": true, "associatedObjects":
[] },
    { "id": "36196", "name": "Clear History", "select": true,
"associatedObjects": [] },
    {
      "id": "36194", "name": "Fetch History (End Time)", "select": true,
      "associatedObjects": [
        { "id": "36193", "name": "fetch_history.start_time" },
        { "id": "36194", "name": "fetch_history.end_time" }
      ]
    }
  ],
  "errCode": 0, "errMsg": ""
}
```

2\.9 Get Addable Object List by Codec ID

```Plain Text
GET /api/dsdevices/objects/payloadcodecs/{codecId}
Authorization: Bearer <JWT>
```

2\.10 Update/Enable/Delete Device Object

```Plain Text
PUT /api/dsdevices/objects/update/{objectId}     (Update)
PUT /api/dsdevices/objects/enable                 (Batch enable/disable)
POST /api/dsdevices/objects/delete                (Delete)
```

2\.11 Copy Device Objects

```Plain Text
POST /api/dsdevices/device/objects/get-copyable-list   (Get copyable
sources)
POST /api/dsdevices/device/objects/copy                (Execute copy)
```

2\.12 IO Device Operations

```Plain Text
GET /api/dsdevices/device/io                           (Get IO list)
PUT /api/dsdevices/device/io/update/{id}               (Update IO)
PUT /api/dsdevices/device/io/copy/{id}                 (Copy IO)
PUT /api/dsdevices/device/io/reset-counter/{deviceId}  (Reset counter)
PUT /api/dsdevices/device/io/force-output/{id}         (Force output)
```

2\.13 Get Access Network List

```Plain Text
```

```
GET /api/access-network
Authorization: Bearer <JWT>
```

Response:

```JSON
{
  "accessNetworks": [
    { "id": "1", "name": "LoRaWAN", "protocol": "lorawan", "interface":
"EU868" },
    { "id": "2", "name": "1", "protocol": "bacnet mstp", "interface":
"rs485-1" },
    { "id": "3", "name": "2", "protocol": "bacnet ip", "interface": "eth0"
},
    { "id": "4", "name": "Modbus", "protocol": "modbus rtu", "interface":
"rs485-2" },
    { "id": "5", "name": "WLAN", "protocol": "bacnet ip", "interface":
"wlan0" },
    { "id": "8", "name": "Modbus1", "protocol": "modbus tcp", "interface":
"" }
  ],
  "accessNetworkTotalCount": "8"
}
```

2\.14 Get RS485 Configuration

```Plain Text
GET /api/access-network/rs485
```

Response:

```JSON
{
  "errCode": 0, "errMsg": "success",
  "result": {
```

```
    "rs485-1": { "id": "1", "baudRate": 9600, "dataBits": 8, "stopBits": 1,
"parity": 0, "dip": false },
    "rs485-2": { "id": "2", "baudRate": 9600, "dataBits": 8, "stopBits": 1,
"parity": 0, "dip": false }
  }
}
```

2.15 Get Interface Occupancy Status

```Plain Text
GET /api/access-network/interface/occupied
```

Response:

```JSON
{
  "errCode": 0, "errMsg": "success",
  "result": {
    "ethList": ["eth0", "eth1", "cellular0", "wlan0"],
    "rs485List": ["rs485-1", "rs485-2"],
    "BacnetOccupiedList": ["eth0", "wlan0", "eth1"],
    "rs485OccupiedList": ["rs485-2", "rs485-1"]
  }
}
```

2.16 Get UR Profile List

```Plain Text
GET /api/urprofiles?limit=9999&offset=0&organizationID=1
```

Response:

```JSON
{
  "totalCount": "10",
  "disable": false,
  "result": [{
    "profile": {
      "profileID": "c61728f0-a941-40ad-b3e6-7c3f1645b42b",
      "supportsClassB": false, "supportsClassC": false,
      "macVersion": "1.0.2", "regParamsRevision": "B",
      "rfRegion": "EU868", "supportsJoin": false
    },
    "name": "ClassA-ABP",
    "organizationID": "1", "networkServerID": "1",
    "using": true, "isDefault": true
  }]
}
```

Built\-in Profiles:

Name	Class	Access Mode	profileID
ClassA\-ABP	A	ABP	c61728f0\-\.\.\.
ClassA\-OTAA	A	OTAA	24415a71\-\.\.\.
ClassB\-ABP	B	ABP	ebc1b444\-\.\.\.
ClassB\-OTAA	B	OTAA	2e3f16e4\-\.\.\.

2\.17 Get Codec Short List

```Plain Text
GET /api/payloadcodecs-short?order=asc&offset=0&limit=9999
```

Response:

```JSON
```

````
{
  "totalCount": "159",
  "result": [{
    "id": "707",
    "name": "AM102",
    "type": "default",
    "devEUIPrefix": "24e124725",
    "protocol": "lorawan",
    "deviceProfileIDs": ["24415a71-b1de-45c0-ac20-9a2590f35d17"],
    "objectCount": 48
  }]
}
```



---



## 3\. Device Library Module \(Device Library / Data Parsing Library\)



**Route**: `/data-services/data-parsing-library`



### 3\.1 Get Codec Settings



```Plain Text
GET /api/payloadcodecs-setting
```



**Response**:

```JSON
{
  "onlineUpgradeEnabled": true,
  "version": "1.5.38",
  "updatedAt": "2026-05-20T04:31:15.863925+01:00",
  "upgradeType": "online"
````

```
}
```

3\.2 Get Built\-in Device Library List

```Plain Text
GET /api/payloadcodecs?order=asc&type=default&limit=10&offset=0
```

Response:

```JSON
{
  "totalCount": "132",
  "result": [{
    "id": "836",
    "name": "CTH01",
    "description": "Intelligent power monitoring terminal",
    "templateID": "0",
    "devEUIPrefix": "24e124637",
    "encoderScript": "/** ... */",
    "decoderScript": "...",
    "objectSpecs": [...],
    "protocol": "lorawan"
  }]
}
```

3\.3 Get Custom Device Library List

```Plain Text
GET /api/payloadcodecs?order=asc&type=custom&limit=10&offset=0
```

Response:

```JSON
{
  "totalCount": "27",
  "result": [{
    "id": "14994",
    "name": "LoRaTest",
    "description": "Smart Fan Coil Thermostat",
    "templateID": "831",
    "devEUIPrefix": "24e124406",
    "encoderScript": "...",
    "protocol": "lorawan"
  }]
}
```

3\.4 Get Codec Details

```Plain Text
GET /api/payloadcodecs/{id}
```

Returns complete codec info \(encoderScript, decoderScript, objectSpecs, etc\.\)\.

3\.5 Create Custom Codec

```Plain Text
POST /api/payloadcodecs
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "CustomDevice",
```

```
  "description": "Custom device codec",
  "type": "custom",
  "protocol": "lorawan",
  "devEUIPrefix": "",
  "templateID": "0",
  "encoderScript": "function Encoder(obj, fPort) { return []; }",
  "decoderScript": "function Decoder(bytes, port) { var decoded = {}; return
decoded; }",
  "objectSpecs": [{
    "objectID": "temperature",
    "name": "Temperature",
    "dataType": "number",
    "unit": "°C",
    "description": ""
  }]
}
```

3\.6 Update/Delete Custom Codec

```Plain Text
PUT /api/payloadcodecs/{id}          (Update)
DELETE /api/payloadcodecs/{id}       (Delete)
```

3\.7 Import/Export Custom Codec

```Plain Text
POST /api/payloadcodecs-export-custom       (Batch export)
GET  /api/payloadcodecs-import-custom       (Import)
POST /api/payloadcodecs-import              (General import)
```

3\.8 Codec Upgrade

```Plain Text
POST /api/payloadcodecs-upgrade             (Online/offline upgrade)
```

3\.9 Codec Test

```Plain Text
POST /api/payloadcodecs-test                (Encode/decode test)
```

3\.10 Check If Object Is Used

```Plain Text
POST /api/payloadcodecs-object-used
```

4\. Data Forwarding Module

Route: `/data-services/data-forwarding`

4\.1 Get Forwarding Rule List

```Plain Text
GET /api/dsforward
Authorization: Bearer <JWT>
```

Response:

```JSON
{
  "totalCount": "301",
  "dataServiceForward": [
    { "id": "4", "name": "NeoMind", "type": "MQTT", "deviceCount": "9",
"objectCount": "147", "status": "enable" },
    { "id": "3", "name": "123", "type": "HTTP", "deviceCount": "1",
"objectCount": "6", "status": "enable" },
    { "id": "2", "name": "eth2 modbus tcp", "type": "Modbus", "deviceCount":
"8", "objectCount": "127", "status": "enable" },
    { "id": "1", "name": "44网段", "type": "BACnet", "deviceCount": "2",
"objectCount": "21", "status": "enable" }
  ]
}
```

Forwarding types: MQTT, HTTP, Modbus, BACnet

4\.2 MQTT Forwarding

4\.2\.1 Create MQTT Forwarding

```Plain Text
POST /api/dsforward/mqtt
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "TestMQTT",
  "server": "broker.example.com",
  "port": 1883,
  "clientID": "",
```

```
  "connectTimeout": 30,
  "keepAliveInterval": 60,
  "dataRetransmission": true,
  "userCredentials": false,
  "userName": "",
  "password": "",
  "useTLS": false,
  "uplinkTopic": "eg71/uplink",
  "uplinkQoS": 0,
  "uplinkRetain": false,
  "metadataEnable": true,
  "metadataDetails": ["deviceName", "deviceID", "objectName"],
  "globalObjectEnable": false,
  "globalObjectDetails": [],
  "status": "enable",
  "description": "",
  "payloadFormat": 0
}
```

4\.2\.2 Get MQTT Forwarding Details

```Plain Text
GET /api/dsforward/mqtt/{forwardId}
```

Response:

```JSON
{
  "id": "4", "enable": true, "name": "NeoMind",
  "server": "101.37.88.247", "port": 1883, "clientID": "",
  "connectStatus": false, "connectTimeout": 30, "keepAliveInterval": 60,
  "dataRetransmission": true,
  "metadataEnable": true,
  "metadataDetails": ["deviceName", "deviceID", "objectName",
"applicationID", "applicationName", "gatewaySN", "gatewayTime"],
  "globalObjectEnable": true,
  "globalObjectDetails": ["DevEUI", "FPort", "Frequency", "RSSI", "SNR",
"DataRate", "FrameCount"],
```

```
  "userCredentials": true, "userName": "admin", "password": "password",
  "useTLS": false, "tlsMode": 0,
  "uplinkTopic": "abc", "uplinkQoS": 0, "uplinkRetain": false,
  "downlinkTopic": "", "downlinkQoS": 0,
  "mcDownlinkTopic": "", "mcDownlinkQoS": 0,
  "onlineTopic": "", "onlineQoS": 0, "onlineRetain": false,
  "offlineTopic": "", "offlineQoS": 0, "offlineRetain": false,
  "ackTopic": "", "ackQoS": 0, "ackRetain": false,
  "errorTopic": "", "errorQoS": 0, "errorRetain": false,
  "requestTopic": "", "requestQoS": 0,
  "responseTopic": "", "responseQoS": 0, "responseRetain": false,
  "willFunction": false, "willSubject": "", "willQoS": 0,
"willRetentionFlag": false, "willMessages": "",
  "description": "", "payloadFormat": 0
}
```

4\.2\.3 Update/Delete MQTT Forwarding

```Plain Text
PUT /api/dsforward/mqtt/{forwardId}           (Update)
DELETE /api/dsforward/{forwardId}             (Delete, general)
PUT /api/dsforward/mqtt/connect/{forwardId}   (Test connection)
```

4\.3 HTTP Forwarding

4\.3\.1 Create HTTP Forwarding

```Plain Text
POST /api/dsforward/http
Authorization: Bearer <JWT>
```

```JSON
{
```

```
  "name": "TestHTTP",
  "dataUpURL": "https://example.com/api/data",
  "onlineNotificationURL": "",
  "offlineNotificationURL": "",
  "ackNotificationURL": "",
  "errorNotificationURL": "",
  "headers": [],
  "metadataEnable": true,
  "metadataDetails": ["deviceName", "deviceID", "objectName"],
  "globalObjectEnable": false,
  "globalObjectDetails": [],
  "enable": true,
  "description": ""
}
```

4\.3\.2 Get HTTP Forwarding Details

```Plain Text
GET /api/dsforward/http/{forwardId}
```

Response:

```JSON
{
  "id": "3", "name": "123",
  "headers": [],
  "dataUpURL": "", "onlineNotificationURL": "", "offlineNotificationURL":
"",
  "ackNotificationURL": "", "errorNotificationURL": "",
  "metadataEnable": true,
  "metadataDetails": ["deviceName", "deviceID", "objectName"],
  "globalObjectEnable": true,
  "globalObjectDetails": ["Frequency", "RSSI", "SNR", "DataRate",
"FrameCount"],
  "description": "", "enable": true
}
```

4\.3\.3 Update HTTP Forwarding

```Plain Text
PUT /api/dsforward/http/{forwardId}
```

4\.4 Modbus Forwarding \(Uses serverId\)

4\.4\.1 Create Modbus Forwarding

```Plain Text
POST /api/protocol/modbus_server/add
Authorization: Bearer <JWT>
```

```JSON
{
  "name": "TestModbus",
  "description": "",
  "port": 503,
  "slaveId": "10",
  "interface": "eth1",
  "ipAddr": "192.168.44.114",
  "connectType": "modbus_tcp",
  "globalObjectsEnable": false,
  "globalObjectsDetails": ["Frequency", "RSSI", "SNR", "DataRate",
"FrameCount", "Status"]
}
```

> ⚠ `port` is **number**, `slaveId` is **string**; port must not conflict with existing Modbus forwarding
>

>

Response: `{"errCode":0,"errMsg":"","serverId":"11"}`

4\.4\.2 Get Modbus Forwarding Details

```Plain Text
POST /api/protocol/modbus_server/get
```

```JSON
{ "serverId": "2" }
```

Response:

```JSON
{
  "id": "2", "enable": true, "name": "eth2 modbus tcp", "description": "",
  "port": 502, "slaveId": "3",
  "interface": "eth1", "ipAddr": "192.168.44.114",
  "connectType": "modbus_tcp",
  "globalObjectsEnable": false,
  "globalObjectsDetails": ["Frequency", "RSSI", "SNR", "DataRate",
"FrameCount", "Status"],
  "objectNum": 127
}
```

4\.4\.3 Update/Delete Modbus Forwarding

```Plain Text
POST /api/protocol/modbus_server/set         (Update)
POST /api/protocol/modbus_server/del         (Delete)
```

```

### 4\.5 BACnet Forwarding \(Uses serverId\)

#### 4\.5\.1 Create BACnet Forwarding

```Plain Text
POST /api/protocol/bacnet_server/add
```

4\.5\.2 Get BACnet Forwarding Details

```Plain Text
POST /api/protocol/bacnet_server/get
```

```JSON
{ "serverId": "1" }
```

Response:

```JSON
{
  "id": "1", "enable": true, "name": "44网段", "description": "",
  "deviceName": "EG71-6438F39949850005", "deviceId": 3753801,
  "ipNatAddr": "", "ipPort": 47808, "interface": "eth1",
  "bbmdEnable": false, "bbmdType": "", "bbmdIp": "", "bbmdPort": 47808,
"bbmdToLive": 60000, "bbmdBdt": [],
  "globalObjectsEnable": false,
  "globalObjectsDetails": ["Frequency", "RSSI", "SNR", "DataRate",
"FrameCount", "Status"],
  "objectNum": 21
}
```
```

```

#### 4\.5\.3 Update/Delete BACnet Forwarding

```Plain Text
POST /api/protocol/bacnet_server/set         (Update)
POST /api/protocol/bacnet_server/del         (Delete)
```

4\.6 Forwarding Device Management \(MQTT/HTTP Types\)

> ⚠ **Important: Write APIs use ****`forwardServiceId`**** \(not ****`forwardId`****\)**
>
> Read operations \(list/addlist\) use `forwardId`, write operations \(create/delete/enable\) use `forwardServiceId`
>
>

4\.6\.1 Get Device List in Forwarding

```Plain Text
POST /api/dsforward/devices/list
```

```JSON
{ "forwardId": "4", "pageSize": 10, "page": 1 }
```

Response:

```JSON
```
```

```
{
  "total": "1",
  "deviceList": [{
    "deviceId": "20",
    "deviceName": "24e124136e445120",
    "eui": "24e124136e445120",
    "deviceType": "None"
  }]
}
```

4\.6\.2 Add Device to Forwarding

```Plain Text
POST /api/dsforward/devices/create
```

```JSON
{
  "forwardServiceId": "4",
  "deviceId": "16",
  "deviceName": "DI-1"
}
```

Response: `{}` \(empty object = success\)

4\.6\.3 Get Addable Device List for Forwarding

```Plain Text
POST /api/dsforward/devices/addlist
```

```JSON
{ "forwardId": "4", "deviceNameOrEui": "", "pageSize": 10, "page": 1 }
```

Response:

```JSON
{
  "total": "0",
  "deviceList": []
}
```

4\.6\.4 Delete Device from Forwarding

```Plain Text
POST /api/dsforward/devices/delete
```

```JSON
{
  "forwardServiceId": "4",
  "deviceIds": ["16"]
}
```

4\.7 Forwarding Object Management \(MQTT/HTTP Types\)

> ⚠ **Same as 4\.6: Write uses ****`forwardServiceId`****, read uses ****`forwardId`****
>
>

4\.7\.1 Get Object List in Forwarding

```Plain Text
POST /api/dsforward/objects
```

```JSON
{ "forwardId": "4", "pageSize": 10, "page": 1 }
```

Response:

```JSON
{
  "objectTotalCount": "147",
  "dsDeviceTotalCount": "9",
  "dsDeviceObjectList": [{
    "deviceId": "1",
    "deviceName": "AO-1",
    "deviceEnable": true,
    "deviceType": "io",
    "devEui": "",
    "ioType": "AO",
    "payloadCodecId": "0",
    "dsDeviceObjectList": [{
      "objectId": "1",
      "objectName": "Present Value",
      "reference": [],
      "enable": true,
      "currentValue": "0",
      "updateAt": "2026-06-11 04:11:58",
      "select": false,
      "description": "",
      "originName": "Present Value",
      "isGlobalObject": false,
      "id": "79",
      "devObjectEnable": true
    }],
    "allowSelectEmptyDevice": false
  }]
}
```

4\.7\.2 Add Object to Forwarding

A single request adds both device and object \(device is auto\-created if not in forwarding\):

```Plain Text
POST /api/dsforward/objects/add
```

```JSON
{
  "forwardServiceId": "4",
  "isSelectAll": false,
  "filterObjectList": [],
  "fsDeviceList": [
    {
      "deviceId": "16",
      "onlyAddEmptyDevice": false,
      "objectList": [
        {
          "objectId": "24",
          "objectName": "Present Value",
          "description": ""
        }
      ]
    }
  ],
  "filterDeviceList": [],
  "objectId": null
}
```

Response: `{"errCode":0,"errMsg":""}`

Field descriptions:

- `fsDeviceList[].deviceId` — Device ID in the system

- `fsDeviceList[].objectList[].objectId` — Object ID within the device \(obtained from addlist\)

- `isSelectAll: true` — Add all selectable objects

4\.7\.3 Get Addable Object List

```Plain Text
POST /api/dsforward/objects/addlist
```

```JSON
{ "forwardId": "4", "deviceNameOrEui": "", "objectName": "", "pageSize": 10, "page": 1 }
```

4\.7\.4 Enable/Disable Forwarding Object

```Plain Text
POST /api/dsforward/objects/enable
```

```JSON
{
  "forwardServiceId": "4",
  "objectIds": ["308"],
  "enable": false
}
```

4\.7\.5 Delete Forwarding Object

```Plain Text
POST /api/dsforward/objects/delete
```

```

```JSON
{
  "forwardServiceId": "4",
  "objectIds": ["308"]
}
```


4\.7\.6 Update Forwarding Object Name/Description


```Plain Text
PUT /api/dsforward/objects/{forwardObjectId}
```

```JSON
{
  "objectName": "New Name",
  "description": "Updated description"
}
```

```Plain Text
POST /api/dsforward/objects/enable            \(Enable/Disable\)

POST /api/dsforward/objects/delete            \(Delete\)

PUT  /api/dsforward/objects/\{id\}              \(Update object
name/description\)
```

```Plain Text

### 4.8 Modbus Forwarding Object Management
```

```Plain Text
POST /api/protocol/modbus_object/get          \(Added object list\)

POST /api/protocol/modbus_object/getall       \(Addable object list\)

POST /api/protocol/modbus_object/add          \(Add object\)

POST /api/protocol/modbus_object/set          \(Update object\)
```
```

```
POST /api/protocol/modbus\_object/del          \(Delete object\)

POST /api/protocol/modbus\_object/switch       \(Enable/Disable\)

POST /api/protocol/modbus\_object/export       \(Export\)

```Plain Text

**Modbus object response example** (Verified):
```json
{
  "id": "2",
  "enable": true,
  "payloadCodecObjectId": "0",
  "name": "Present Value",
  "devObjId": "1",
  "devObjName": "Present Value",
  "registerAddr": 0,
  "registerType": "holding_register",
  "registerNum": 2,
  "dataType": "float32_dcba",
  "unit": "V",
  "unitId": "5",
  "description": "",
  "value": "0",
  "updateTime": "2026-06-11 04:11:58",
  "isGlobalObject": false
}
```


registerType options: `coil`, `discrete`, `input_register`, `holding_register`

dataType options: `flag`, `uint16_ba`, `uint32_dcba`, `float32_dcba`, etc\.

4\.9 BACnet Forwarding Object Management


```Plain Text
```

```
POST /api/protocol/bacnet_object/get          (Added normal objects)
POST /api/protocol/bacnet_object/getall       (Addable objects)
POST /api/protocol/bacnet_object/getNc        (Notification Class objects)
POST /api/protocol/bacnet_object/getNcInstanceId
POST /api/protocol/bacnet_object/add          (Add normal object)
POST /api/protocol/bacnet_object/addNc        (Add NC object)
POST /api/protocol/bacnet_object/set          (Update normal object)
POST /api/protocol/bacnet_object/setNc        (Update NC object)
POST /api/protocol/bacnet_object/del          (Delete normal object)
POST /api/protocol/bacnet_object/delNc        (Delete NC object)
POST /api/protocol/bacnet_object/switch       (Enable/Disable)
POST /api/protocol/bacnet_object/export       (Export)
```

BACnet object response example

```JSON
{
  "id": "271",
  "enable": true,
  "name": "111.Battery",
  "devObjId": "192",
  "devObjName": "Clear History",
  "type": "Binary-Output",
  "instanceId": 0,
  "unit": "no-units",
  "unitId": "95",
  "covEnable": 0,
  "covIncrement": "1.0",
  "polarity": 0,
  "relinquishDefault": "0",
  "activeText": "yes",
  "inactiveText": "no",
  "eventDetectionEnable": false,
  "notificationClass": -1,
  "highLimit": "", "lowLimit": "", "deadband": "",
  "limitEnable": 3, "eventEnable": 5,
  "notifyType": 0, "alarmValue": 0,
  "value": "-",
  "updateTime": "-"
}
```

BACnet type options: `Analog-Input`, `Analog-Output`, `Analog-Value`, `Binary-Input`, `Binary-Output`, `Binary-Value`, `Multi-State-Input`, `Multi-State-Output`, `Multi-State-Value`

5\. Network Configuration Module

Route: `/network/interfaces`

5\.1 Get WAN Configuration

```Plain Text
POST /cgi
Authorization: Bearer login=admin;<td>
```

```JSON
{
  "execute": 1, "core": "yruo_wan", "function": "get",
  "values": [{ "base": "yruo_wan" }], "id": 2
}
```

Response:

```JSON
{
  "status": 0,
  "result": [{ "get": [{
    "type": "yruo_wan",
```

```
    "index": "xWI3QOoxni7t2JO386ovEO9JNs31DFs",
    "value": {
      "enable_wan": 1, "port": "eth1",
      "protocol": 0, "mtu": 1500, "enable_nat": 1,
      "pri_dns": "8.8.8.8", "sec_dns": "223.5.5.5",
      "static": {
        "ip_address": "192.168.44.114", "netmask": "255.255.255.0",
        "gateway": "192.168.44.1", "multi_ip": []
      },
      "dhcp": { "mtu": 1500 },
      "pppoe": { "mtu": 1500 }
    }
  }]}]
}
```

- `protocol`: 0=Static, 1=DHCP, 2=PPPoE

5\.2 Get LAN Configuration

```Plain Text
POST /cgi  core=yruo_lan, base=yruo_lan
```

Response:

```JSON
{
  "result": [{ "get": [
    { "type": "yruo_lan", "index": "eth0", "value": {
      "port": "eth0", "ip_address": "192.168.1.1", "netmask":
"255.255.255.0", "mtu": 1500, "multi_ip": []
    }},
    { "type": "yruo_lan", "index": "Bridge0", "value": {
      "port": "Bridge0", "ip_address": "192.168.3.1", "netmask":
"255.255.255.0", "mtu": 1500, "multi_ip": []
    }}
  ]}]
```

```
}
```

5\.3 Get Bridge Configuration

```Plain Text
POST /cgi  core=yruo_bridge, base=yruo_bridge
```

Response:

```JSON
{
  "result": [{ "get": [{
    "type": "yruo_bridge", "index": "Bridge0",
    "value": {
      "enable_bridge": false, "name": "Bridge0", "protocol": 0,
      "stp": false, "mtu": 1500, "enable_nat": false,
      "pri_dns": "", "sec_dns": "",
      "static": { "ip_address": "192.168.3.1", "netmask": "255.255.255.0",
"gateway": "", "multi_ip": [] },
      "dhcp": { "use_peer_dns": false }
    }
  }]}]
}
```

5\.4 Get Port Status

```Plain Text
POST /cgi  core=yruo_port, base=yruo_port
```

Response:

```JSON
{
  "result": [{ "get": [
    { "type": "yruo_port", "index": "eth1", "value": { "port": "eth1",
"status": 1, "property": 2, "speed": 0, "duplex": 0, "conferred": 0 }},
    { "type": "yruo_port", "index": "eth0", "value": { "port": "eth0",
"status": 1, "property": 1, "speed": 0, "duplex": 0, "conferred": 0 }}
  ]}]
}
```

- `property`: 1=LAN, 2=WAN

- `status`: 1=Connected

5\.5 Get DHCP Server Configuration

```Plain Text
POST /cgi  core=yruo_dhcpserver, base=yruo_dhcpserver
```

Response:

```JSON
{
  "result": [{ "get": [
    { "type": "yruo_dhcpserver", "index": "wlan0", "value": {
      "enable": true, "relay_enabled": false,
      "network": "wlan0", "network_array": ["wlan0", "eth0", "Bridge0"],
      "start_ip": "192.168.2.100", "end_ip": "192.168.2.199",
      "mask": "255.255.255.0", "lease": 1440,
      "wins": "", "pri_dns": "8.8.8.8", "sec_dns": "",
      "static_ip_map": []
    }},
    { "type": "yruo_dhcpserver", "index": "eth0", "value": {
      "enable": true, "network": "eth0",
      "start_ip": "192.168.1.100", "end_ip": "192.168.1.199",
      "mask": "255.255.255.0", "lease": 1440,
```

```
      "pri_dns": "192.168.1.1", "sec_dns": "223.5.5.5"
    }},
    { "type": "yruo_dhcpserver", "index": "eth1", "value": { "enable": false
}},
    { "type": "yruo_dhcpserver", "index": "Bridge0", "value": { "enable":
false }}
  ]}]
}
```

5\.6 Get All Interface IP Addresses

```Plain Text
GET /api/general-info/interface/IpAddress
```

Response:

```JSON
{
  "errCode": 0, "errMsg": "success",
  "result": [
    { "interfaceName": "eth1", "ipAddress": "192.168.44.114", "netMask":
"255.255.255.0", "isUp": true },
    { "interfaceName": "eth0", "ipAddress": "192.168.1.1", "netMask":
"255.255.255.0", "isUp": true },
    { "interfaceName": "wlan0", "ipAddress": "192.168.2.1", "netMask":
"255.255.255.0", "isUp": true },
    { "interfaceName": "cellular0", "ipAddress": "10.61.96.142", "netMask":
"255.255.255.0", "isUp": true },
    { "interfaceName": "docker0", "ipAddress": "172.17.0.1", "netMask":
"255.255.0.0", "isUp": true }
  ]
}
```

5\.7 Modify Network Configuration \(CGI set\)

Change `function` from `get` to `set`, provide the complete configuration object in `values`:

```JSON
{
  "execute": 1, "core": "yruo_wan", "function": "set",
  "values": [{
    "base": "yruo_wan",
    "index": "<wan_index from get>",
    "value": {
      "enable_wan": 1, "port": "eth1", "protocol": 0,
      "mtu": 1500, "enable_nat": 1,
      "pri_dns": "8.8.8.8", "sec_dns": "223.5.5.5",
      "static": {
        "ip_address": "192.168.44.114", "netmask": "255.255.255.0",
        "gateway": "192.168.44.1", "multi_ip": []
      },
      "dhcp": { "mtu": 1500, "use_peer_dns": true },
      "pppoe": { "username": "", "password": "", "mtu": 1500 }
    }
  }]
}
```

Same applies to `yruo_lan`, `yruo_bridge`, `yruo_dhcpserver`, etc\.

6\. VPN Configuration Module

Route: `/network/vpn`

6\.1 OpenVPN Client

```Plain Text
POST /cgi  core=yruo_vpn_openvpn_client, base=yruo_vpn_openvpn_client
```

Response \(3 client slots\):

```JSON
{
  "result": [{ "get": [
    { "type": "yruo_vpn_openvpn_client", "index": 1, "value": {
      "remote_ip": "", "protocol": 0, "config_mode": 0,
      "port": 1194, "interface_type": 0, "authentication": 0,
      "cipher": 0, "local_virtual_ip": "", "local_virtual_mask": "",
      "remote_virtual_ip": "", "username": "", "password": "",
      "tls_authentication": 0, "enable_nat": 1, "default_route": 0,
      "comp": 1, "ping_interval": 60, "ping_restart": 300,
      "mtu": 1500, "fragment": 1500, "log_level": 0,
      "expert_options": "", "enable": 0, "remote_subnet": []
    }}
  ]}]
}
```

Field descriptions:

Field	Description	Options
`protocol`	Transport protocol	0=UDP, 1=TCP
`config_mode`	Config mode	0=Page configuration, 1=File configuration
`interface_type`	Interface type	0=Tun, 1=Tap
`authentication`	Authentication method	0=None
`cipher`	Cipher	0=None
`comp`	Compression	1=None, other=LZO
`enable`	Enable	0=Disable, 1=Enable
`tls_authentication`	TLS authentication	0=Disable
`enable_nat`	NAT	0=Disable, 1=Enable
`default_route`	Default route	0=Disable, 1=Enable

6\.2 OpenVPN Server

```Plain Text
POST /cgi  core=yruo_vpn_openvpn_server, base=yruo_vpn_openvpn_server
```

Response:

```JSON
{
  "result": [{ "get": [{
    "type": "yruo_vpn_openvpn_server", "index": 1, "value": {
      "listen_ip": "", "protocol": 0, "config_mode": 0,
      "port": 1194, "interface_type": 0, "authentication": 0,
      "cipher": 0, "local_virtual_ip": "", "local_virtual_mask": "",
      "remote_virtual_ip": "",
      "client_virtual_subnet": "10.8.0.0", "client_virtual_mask":
"255.255.255.0",
      "tls_authentication": 0, "enable_nat": 1, "default_route": 0,
      "comp": 1, "ping_interval": 60, "ping_restart": 150,
      "mtu": 1500, "fragment": 1500,
      "client_to_client": 1, "client_dup": 0, "crl": 0,
      "renegotiation_time": 3600, "max_clients": 16,
      "log_level": 0, "expert_options": "", "enable": 0,
      "local_subnet": [], "account": [], "client_subnet": []
    }
  }]}]
}
```

Server\-specific fields:

Field	Description
`client_to_client`	Client\-to\-client communication: 0=Disable, 1=Enable
`client_dup`	Duplicate client: 0=Not allowed
`crl`	Certificate revocation list

`renegotiation_time`	Renegotiation time \(seconds\)
`max_clients`	Max clients
`account`	User account list
`local_subnet`	Local subnet list
`client_subnet`	Client subnet list

6\.3 VPN Certificates

```Plain Text
POST /cgi  core=yruo_vpn_certificate, base=yruo_vpn_certificate
```

Response:

```JSON
{
  "result": [{ "get": [
    { "type": "yruo_vpn_openvn_server_certificate", "index": 1, "value": {}
},
    { "type": "yruo_vpn_openvpn_client_certificate", "index": 2, "value": {}
},
    { "type": "yruo_vpn_openvpn_client_certificate", "index": 3, "value": {}
},
    { "type": "yruo_vpn_openvpn_client_certificate", "index": 4, "value": {}
},
    { "type": "yruo_vpn_ipsec_certificate", "index": 5, "value": {} },
    { "type": "yruo_vpn_ipsec_certificate", "index": 6, "value": {} },
    { "type": "yruo_vpn_ipsec_certificate", "index": 7, "value": {} }
  ]}]
}
```

6\.4 IPSec

```Plain Text
POST /cgi  core=yruo_vpn_ipsec, base=yruo_vpn_ipsec
```

```

**Response** \(3 slots\):

```JSON
{
  "type": "yruo_vpn_ipsec", "index": 1, "value": {
    "remote_ip": "", "local_subnet": "", "local_subnet_mask": "",
    "remote_subnet": "", "remote_subnet_mask": "",
    "lifetime": 3600, "pfs_group": 0, "mode": 0, "protocol": 0,
    "sa_algorithm": 0, "version": 1,
    "authentication_type_local": 0, "secrets_local": "",
    "dpd_interval": 30, "dpd_timeout": 150,
    "negotiation_mode": 0, "local_id_type": 0, "remote_id_type": 0,
    "local_id": "", "remote_id": "",
    "enable_xauth": 0, "enable_comp": 0,
    "encryption_algorithm": 2, "authentication_algorithm": 0,
    "dh_group": 0, "ike_lifetime": 10800,
    "ipsec_over_vpn": 0, "enable": 0
  }
}
```

6\.5 L2TP

```Plain Text
POST /cgi  core=yruo_vpn_l2tp, base=yruo_vpn_l2tp
```

Response \(3 slots\):

```JSON
{
  "type": "yruo_vpn_l2tp", "index": 1, "value": {
    "remote_ip": "", "user": "", "password": "",
    "auth_method": 0, "enable_nat": 1, "default_route": 0,
    "enable_mppe": 0, "usepeerdns": 0,
    "ac_comp": 0, "p_comp": 0, "asyncmap": "ffffffff",
```
```

```
    "mtu": 1436, "mru": 1500,
    "echo_interval": 60, "max_retries": 0,
    "local_ip": "", "peer_ip": "",
    "expert_opts": "", "remote_subnet": "", "remote_mask": "",
    "key": "", "enable": 0
  }
}
```

6\.6 PPTP

```Plain Text
POST /cgi  core=yruo_vpn_pptp, base=yruo_vpn_pptp
```

Response \(3 slots\):

```JSON
{
  "type": "yruo_vpn_pptp", "index": 1, "value": {
    "remote_ip": "", "user": "", "password": "",
    "auth_method": 0, "enable_nat": 1, "default_route": 0,
    "enable_mppe": 0, "usepeerdns": 0,
    "ac_comp": 0, "p_comp": 0, "asyncmap": "ffffffff",
    "mtu": 1436, "mru": 1500,
    "echo_interval": 60, "max_retries": 0,
    "local_ip": "", "peer_ip": "",
    "expert_opts": "", "remote_subnet": "", "remote_mask": "",
    "enable": 0
  }
}
```

6\.7 WireGuard

```Plain Text
```

```
POST /cgi  core=yruo_vpn_wireguard, base=yruo_vpn_wireguard
```

> Current device returns `status: -1, "Object not found"` — firmware 71\.0\.0\.2 may not yet include the WireGuard module\.
>
>

6\.8 Modify VPN Configuration \(CGI set\)

Change `function` from `get` to `set`, keep the complete value object:

```JSON
{
  "execute": 1, "core": "yruo_vpn_openvpn_client", "function": "set",
  "values": [{
    "base": "yruo_vpn_openvpn_client",
    "index": 1,
    "value": { /* complete get response + modifications */ }
  }]
}
```

7\. Other Useful APIs

7\.1 Dashboard Status

```Plain Text
POST /cgi  core=yruo_status, base=dashboard
```

Returns complete device status: model, sn, firmware_ver, hardware_ver, local_time, uptime, cpu_load, memory_total/free, flash_total/free, dev_eui, gateway_id, cpu_temperature, hostname, device_count, lorawan_band, cell_status, etc\.

7\.2 System Base Configuration

```Plain Text
POST /cgi  core=yruo_system, base=general
```

CGI read:

```JSON
{ "execute": 1, "core": "yruo_system", "function": "get", "values": [{ "base": "general" }] }
```

CGI modify:

```JSON
{ "execute": 1, "core": "yruo_system", "function": "set", "values": [{ ... }] }
```

Returns: hostname, web_login_timeout, http/https/telnet/ssh port and enable status, remote/local access control, etc\.

7\.3 Access Interface Info

```Plain Text
GET /api/general-info/interface/with-access-info
```

Response:

```JSON
{
  "errCode": 0, "errMsg": "success",
  "interfaceName": "eth1", "interfaceType": "wan",
  "protocol": "https", "warning": false
}
```

7\.4 LoRaWAN Network Configuration

```Plain Text
POST /cgi  core=yruo_loragw, base=ns_general    (Get)
POST /cgi  core=yruo_loragw, function=set        (Set)
```

7\.5 User Management

```Plain Text
POST /cgi  core=yruo_usermanagement, base=security        (Change password)
POST /cgi  core=yruo_usermanagement, base=check_pass      (Check default
password)
```

8\. Complete API Route Table

REST API Endpoints

Method	URL	Description
POST	`/api/internal/login`	JWT login
POST	`/api/dsdevices/device`	Device list
POST	`/api/dsdevices/device/delete`	Delete device
POST	`/api/dsdevices/device/bacnet/create`	Create BACnet device
POST	`/api/dsdevices/device/modbus/create`	Create Modbus device
POST	`/api/dsdevices/device/knx/create`	Create KNX device
PUT	`/api/dsdevices/device/bacnet/{id}`	Update BACnet device
PUT	`/api/dsdevices/device/modbus/{id}`	Update Modbus device
PUT	`/api/dsdevices/device/knx/{id}`	Update KNX device
GET	`/api/dsdevices/objects/{deviceId}`	Device object list
POST	`/api/dsdevices/objects/create`	Add device object
PUT	`/api/dsdevices/objects/update/{objectId}`	Update device object
PUT	`/api/dsdevices/objects/enable`	Enable/Disable device object
POST	`/api/dsdevices/objects/delete`	Delete device object
GET	`/api/dsdevices/objects/addlist/{deviceId}`	Addable object list
GET	`/api/dsdevices/objects/payloadcodecs/{codecId}`	Codec object list
POST	`/api/dsdevices/objects/copy`	Copy objects
POST	`/api/dsdevices/objects/bacnet/create`	Create BACnet object
POST	`/api/dsdevices/objects/modbus/create`	Create Modbus object
POST	`/api/dsdevices/objects/knx/create`	Create KNX object
GET	`/api/dsdevices/device/io`	IO device list
PUT	`/api/dsdevices/device/io/update/{id}`	Update IO
POST	`/api/dsdevices/device/is-used-in-forward-service/{deviceId}`	Check forwarding reference
POST	`/api/urdevices`	Create LoRaWAN device
PUT	`/api/urdevices/{devEUI}`	Update LoRaWAN device
GET	`/api/urdevices/simple/{id}`	Device details
GET	`/api/urprofiles`	Profile list
POST	`/api/urprofiles`	Create Profile
POST	`/api/urprofiles/lns`	Get Profile by EUI
GET	`/api/access-network`	Access network list
GET	`/api/access-network/rs485`	RS485 configuration
GET	`/api/access-network/bacnet`	BACnet configuration
GET	`/api/access-network/modbus`	Modbus configuration
GET	`/api/access-network/knx`	KNX configuration
GET	`/api/access-network/interface/occupied`	Interface occupancy status

```
|GET|`/api/payloadcodecs`|Codec list \(type=default/custom\)|
|GET|`/api/payloadcodecs/{id}`|Codec details|
|POST|`/api/payloadcodecs`|Create custom codec|
|PUT|`/api/payloadcodecs/{id}`|Update codec|
|DELETE|`/api/payloadcodecs/{id}`|Delete codec|
|GET|`/api/payloadcodecs-short`|Codec short list|
|GET|`/api/payloadcodecs-setting`|Codec settings|
|POST|`/api/payloadcodecs-test`|Encode/decode test|
|POST|`/api/payloadcodecs-export-custom`|Export custom codec|
|GET|`/api/payloadcodecs-import-custom`|Import custom codec|
|POST|`/api/payloadcodecs-object-used`|Check object usage|
|GET|`/api/dsforward`|Forwarding rule list|
|POST|`/api/dsforward/mqtt`|Create MQTT forwarding|
|GET|`/api/dsforward/mqtt/{id}`|MQTT details|
|PUT|`/api/dsforward/mqtt/{id}`|Update MQTT forwarding|
|PUT|`/api/dsforward/mqtt/connect/{id}`|Test MQTT connection|
|POST|`/api/dsforward/http`|Create HTTP forwarding|
|GET|`/api/dsforward/http/{id}`|HTTP details|
|PUT|`/api/dsforward/http/{id}`|Update HTTP forwarding|
|DELETE|`/api/dsforward/{id}`|Delete forwarding rule|
|POST|`/api/dsforward/devices/list`|Forwarding device list \(body:
`forwardId`\)|
|POST|`/api/dsforward/devices/create`|Add device to forwarding \(body:
**`forwardServiceId`**\)|
|POST|`/api/dsforward/devices/addlist`|Addable device list \(body:
`forwardId`\)|
|POST|`/api/dsforward/devices/delete`|Delete forwarding device \(body:
**`forwardServiceId`**\)|
|POST|`/api/dsforward/objects`|Forwarding object list \(body: `forwardId`\)|
|POST|`/api/dsforward/objects/add`|Add object to forwarding \(body:
**`forwardServiceId`**\)|
|POST|`/api/dsforward/objects/addlist`|Addable object list \(body:
`forwardId`\)|
|POST|`/api/dsforward/objects/enable`|Enable/Disable forwarding object
\(body: **`forwardServiceId`**\)|
|POST|`/api/dsforward/objects/delete`|Delete forwarding object \(body:
**`forwardServiceId`**\)|
|PUT|`/api/dsforward/objects/{id}`|Update forwarding object|
|POST|`/api/protocol/modbus_server/add`|Create Modbus forwarding|
|POST|`/api/protocol/modbus_server/get`|Modbus forwarding details|
|POST|`/api/protocol/modbus_server/set`|Update Modbus forwarding|
|POST|`/api/protocol/modbus_server/del`|Delete Modbus forwarding|
|POST|`/api/protocol/modbus_object/get`|Modbus forwarding objects|
|POST|`/api/protocol/modbus_object/getall`|Modbus addable objects|
|POST|`/api/protocol/modbus_object/add`|Add Modbus object|
```

```
|POST|`/api/protocol/modbus_object/set`|Update Modbus object|
|POST|`/api/protocol/modbus_object/del`|Delete Modbus object|
|POST|`/api/protocol/modbus_object/switch`|Enable/Disable Modbus object|
|POST|`/api/protocol/modbus_object/export`|Export Modbus objects|
|POST|`/api/protocol/bacnet_server/add`|Create BACnet forwarding|
|POST|`/api/protocol/bacnet_server/get`|BACnet forwarding details|
|POST|`/api/protocol/bacnet_server/set`|Update BACnet forwarding|
|POST|`/api/protocol/bacnet_server/del`|Delete BACnet forwarding|
|POST|`/api/protocol/bacnet_object/get`|BACnet forwarding objects|
|POST|`/api/protocol/bacnet_object/getall`|BACnet addable objects|
|POST|`/api/protocol/bacnet_object/getNc`|BACnet NC objects|
|POST|`/api/protocol/bacnet_object/add`|Add BACnet object|
|POST|`/api/protocol/bacnet_object/addNc`|Add BACnet NC object|
|POST|`/api/protocol/bacnet_object/set`|Update BACnet object|
|POST|`/api/protocol/bacnet_object/setNc`|Update BACnet NC object|
|POST|`/api/protocol/bacnet_object/del`|Delete BACnet object|
|POST|`/api/protocol/bacnet_object/delNc`|Delete BACnet NC object|
|POST|`/api/protocol/bacnet_object/switch`|Enable/Disable BACnet object|
|POST|`/api/protocol/bacnet_object/export`|Export BACnet objects|
|POST|`/api/dsdevices/bacnet/scan`|BACnet scan|
|POST|`/api/dsdevices/bacnet/scan/stop`|Stop scan|
|POST|`/api/dsdevices/bacnet/scan/config`|Scan configuration|
|GET|`/api/gateways`|Gateway list|
|GET|`/api/general-info/interface/with-access-info`|Access interface info|
|GET|`/api/general-info/interface/IpAddress`|Interface IP addresses|
|GET|`/api/general-info/weblog`|Web log|



### CGI Endpoints \(POST /cgi\)



|Core|Base|Description|
|---|---|---|
|`user`|\-|Login|
|`yruo_system`|`general`|System configuration|
|`yruo_status`|`dashboard`|Dashboard status|
|`yruo_usermanagement`|`security`, `check_pass`|User management|
|`yruo_wizard`|`yruo_wizard`|Wizard status|
|`yruo_loragw`|`ns_general`|LoRaWAN network configuration|
|`yruo_wan`|`yruo_wan`|WAN configuration|
|`yruo_lan`|`yruo_lan`|LAN configuration|
|`yruo_bridge`|`yruo_bridge`|Bridge configuration|
|`yruo_port`|`yruo_port`|Port status|
```

```
|`yruo_dhcpserver`|`yruo_dhcpserver`|DHCP server|
|`yruo_vpn_openvpn_client`|`yruo_vpn_openvpn_client`|OpenVPN Client|
|`yruo_vpn_openvpn_server`|`yruo_vpn_openvpn_server`|OpenVPN Server|
|`yruo_vpn_certificate`|`yruo_vpn_certificate`|VPN certificates|
|`yruo_vpn_ipsec`|`yruo_vpn_ipsec`|IPSec|
|`yruo_vpn_wireguard`|`yruo_vpn_wireguard`|WireGuard|
|`yruo_vpn_l2tp`|`yruo_vpn_l2tp`|L2TP|
|`yruo_vpn_pptp`|`yruo_vpn_pptp`|PPTP|



---



## 9\. Frontend Route Table



|Route|Module|
|---|---|
|`/dashboard`|Dashboard|
|`/data-services/equipment-data`|Equipment Data|
|`/data-services/equipment-data/add-device`|Add Device Manually|
|`/data-services/equipment-data/scan-device`|Scan Devices|
|`/data-services/equipment-data/device-object`|Device Object Detail|
|`/data-services/equipment-data/multicast-form`|Multicast Config|
|`/data-services/equipment-data/fuota-form`|FUOTA Config|
|`/data-services/data-forwarding`|Data Forwarding|
|`/data-services/data-forwarding/forwarding-form`|Create/Edit Forwarding
Rule|
|`/data-services/data-forwarding/http-mqtt-object-list`|HTTP/MQTT Object
List|
|`/data-services/data-forwarding/bacnet-object-list`|BACnet Object List|
|`/data-services/data-forwarding/modbus-object-list`|Modbus Object List|
|`/data-services/data-parsing-library`|Device Library|
|`/data-services/data-parsing-library/default-parsing-detail`|Built\-in
Codec Detail|
|`/data-services/data-parsing-library/custom-parsing-detail`|Custom Codec
Detail|
|`/data-services/data-flow`|Data Flow|
|`/network`|Network Overview|
|`/network/interfaces`|Network Interfaces|
|`/network/firewall`|Firewall|
|`/network/ddns`|DDNS|
|`/network/backup`|Link Failover|
```

```
|`/network/vpn`|VPN|
|`/platform`|Platform Management|
|`/system/setting`|System Settings|
|`/system/user`|User Management|
|`/system/serve`|Service Management|
|`/system/maintenance`|System Maintenance|
|`/system/log`|System Log|
|`/system/snmp`|SNMP|
|`/system/events`|Event Management|
|`/app/python`|Python App|
|`/app/nodered`|Node\-RED|
|`/upgrade`|Firmware Upgrade|
|`/guide`|Wizard|
```