

# Special Documentation Tankvision Tank Scanner NXA820

API definition for ERP Integration





## Table of contents

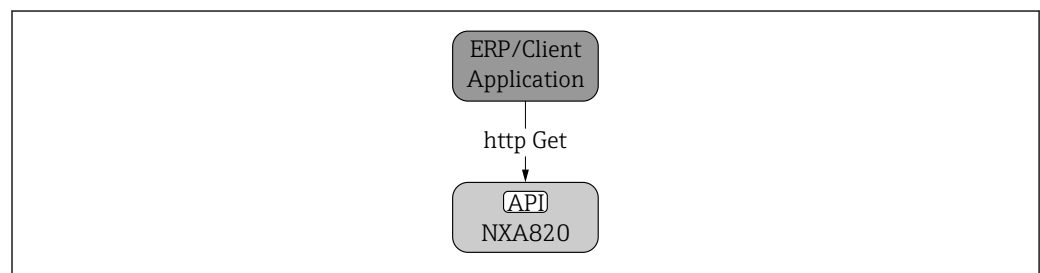
<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Scope</b> .....	<b>4</b>
<b>3</b>	<b>General comments</b> .....	<b>5</b>
3.1	RESTful API .....	5
3.2	Encryption .....	5
3.3	Data format & encoding .....	5
3.4	API Versioning .....	5
3.5	Throttling of requests (Rate Limiting) .....	5
<b>4</b>	<b>Rest api documentation</b> .....	<b>6</b>
4.1	Use Case 1 – Integration of Inventory Data in the Business System .....	6
	4.1.1 Tank information .....	6
	4.1.2 Tank parameter .....	7
4.2	Use Case 2 – Reconciliation of inventories & warehouse management .....	10
	4.2.1 Group information .....	11
	4.2.2 Group parameters .....	13
<b>5</b>	<b>References</b> .....	<b>17</b>
5.1	TankInfo.json .....	17
5.2	TankParam.json .....	19
5.3	GroupInfo.json .....	21
5.4	GroupParam.json .....	23

# 1 Introduction

The Tankvision Tank Scanner NXA820 has a dedicated Rest API interface for third-party applications like ERP/SAP. Purpose of this document is to describe in detail how end users can integrate the Tank Scanner NXA820 to third-party client applications.

# 2 Scope

The scope of this document is to present a brief description and examples of the Tank Scanner NXA820 API services and the mechanisms by which a third-party client application can retrieve the expected data.



A0053300

1 Inventory Synchronization

The device supports the following workflows:

Workflow: Inventory Synchronization

- Use case 1: Integration of Inventory Data in the Business System
- Use Case 2: Reconciliation of inventories & warehouse management

## **3 General comments**

### **3.1 RESTful API**

The API services are based on HTTP RESTful, and it supports only the GET method.

### **3.2 Encryption**

The API does not support any encrypted https protocol. Only http is supported.

### **3.3 Data format & encoding**

API response is in JSON format. The encoding used for request and response is UTF-8 format.

### **3.4 API Versioning**

The device does not support any versioning. Outputs from these APIs are fixed and do not change with software version.

### **3.5 Throttling of requests (Rate Limiting)**

To ensure smooth operation of the device, number of requests to these APIs should be limited.

## 4 Rest api documentation

### 4.1 Use Case 1 – Integration of Inventory Data in the Business System

Tank Scanner NXA820 provides a Rest interface for 'Tank information' and 'Tank parameter'.

#### 4.1.1 Tank information

Receive tank related information like Tank name, Tank id and Tank status.

##### URL

<GET /GWTHandler.esp?DATATYPE=99&TYPE=TANKINFO>

##### URL parameters

Key	Value	Required/Optional	Type	Description
DATATYPE	99	Required	Integer	-
TYPE	TANKINFO	Required	String	-

##### Data parameters

None

##### Success response

- Code: 200 OK
- Content

```
{
  "deviceName": "NXA_DEMO",
  "ipAddress": "10.56.53.33",
  "tanks": [
    {
      "tankId": 1,
      "tankName": "Tank-1",
      "tankStatus": 649
    },
    {
      "tankId": 2,
      "tankName": "Tank-2",
      "tankStatus": 649
    },
    ....
    {
      "tankId": 15,
      "tankName": "Tank-15",
```

```

        "tankStatus": 649
    }
}

```

Here,

Field	Description
deviceName	Tank Scanner NXA820 device name
ipAddress	IP address of Tank Scanner NXA820
tankId	Tank identifier 1 – First Tank, 2 – second tank etc.
tankName	Name of Tank
tankStatus	Status of Tank 646 – in Maintenance 647 – Locked 648 – Manual 649 – In Operation

#### Error response

- Code: 404 Not found
- Content  
{Access Error: 404 -- Not Found}

#### Example

<http://172.16.40.153/GWTHandler.esp?DATATYPE=99&TYPE=TANKINFO>

#### Notes

- Devices of type Tank Scanner NXA820 with Modbus protocol or WM550 protocol provide tank information for 15 tanks
- Devices of type Tank Scanner NXA820 with V1 protocol provide tank information for 10 tanks

### 4.1.2 Tank parameter

Provides tank parameter related information like value, status, timestamp, unit and assigned product information.

Tank Scanner NXA820 provides the following information via Rest interface:

- Total Observed Volume
- Available Volume
- Remaining Capacity
- Gross Observed Volume
- Gross Std Volume
- Net Std Volume
- Liquid Mass
- Mass in Vapor
- Total Mass
- Observed Density
- Product ID
- Product Name

#### URL

<GET /GWTHandler.esp?DATATYPE=99&TYPE=TANKPARAM&TANKID=<tankid>>

**URL parameters**

Key	Value	Required/Optional	Type	Description
DATATYPE	99	Required	Integer	-
TYPE	TANKPARAM	Required	String	-
TANKID	1-15	Required	Integer	V1 range of Tank Scanner NXA820 is 1-10

*Data parameters*

None

*Success response*

- Code: 200 OK
- Content

```
{
  "productName":"Petrol",
  "productId":"2460",
  "parameterlist": [
    {
      "id":"628",
      "value":"0.000",
      "unitType":"13",
      "statusId":"675",
      "timestamp":"14:56:09 04/04/2023"
    },
    {
      "id":"717",
      "value":"1010.000",
      "unitType":"9",
      "statusId":"657",
      "timestamp":"10:41:38 21/04/2023"
    },
    {
      "id":"718",
      "value":"0.000",
      "unitType":"9",
      "statusId":"657",
      "timestamp":"10:41:38 21/04/2023"
    },
    {
      "id":"719",
      "value":"1010.000",
      "unitType":"9",
      "statusId":"657",

```



```
"timestamp":"10:41:38 21/04/2023"
},
{
  "id":"726",
  "value":"1010.000",
  "unitType":"9",
  "statusId":"657",
  "timestamp":"10:41:38 21/04/2023"
},
{
  "id":"727",
  "value":"1010.000",
  "unitType":"9",
  "statusId":"676",
  "timestamp":"10:41:38 21/04/2023"
},
{
  "id":"728",
  "value":"1010.000",
  "unitType":"9",
  "statusId":"676",
  "timestamp":"10:41:38 21/04/2023"
},
{
  "id":"729",
  "value":"0.000",
  "unitType":"10",
  "statusId":"676",
  "timestamp":"10:41:38 21/04/2023"
},
{
  "id":"730",
  "value":"0.000",
  "unitType":"10",
  "statusId":"676",
  "timestamp":"10:41:38 21/04/2023"
},
{
  "id":"756",
  "value":"0.000",
  "unitType":"10",
  "statusId":"655",
```

```
"timestamp": "10:41:38 21/04/2023"
}}
}
```

Here,

Field	Description
productName	- Assigned product to given tank - 'No Product' when no product assigned to Tank.
productId	Product Identifier
Id	Parameter identifier Here, 717 – Total Observed Volume 718 – Remaining Capacity 719 – Available Volume 726 – Gross Observed Volume 727 – Gross Std Volume 728 – Net Std Volume 628 – Observed Density 730 – Total Mass 750 - Liquid Mass 756 – Mass in Vapor
value	Parameter value in float format.
unitType	Parameter unit in SI unit. Here, 13 – kg/m <sup>3</sup> 9 – m <sup>3</sup> 10 - kg
statusId	Parameter Status Here, 655 – Valid/OK 656 – Fail 657 – Manual 675 – Initialized 676 – No data 677 – Invalid Value 680 – Last valid value 706 – Not calibrated
timestamp	Parameter local time of Tank Scanner NXA820 in '24HH:MM:SS dd/mm/yyyy' format

#### Error response

- Code: 404 Not found
- Content  
{Access Error: 404 -- Not Found}

#### Example

<http://10.56.53.33/GWTHandler.esp?DATATYPE=99&TYPE=TANKPARAM&TANKID=1>

## 4.2 Use Case 2 – Reconciliation of inventories & warehouse management

This interface is similar Use case 1, here this interface provides group and product related information.

### 4.2.1 Group information

Before accessing the group data, the client application should know the available groups in the system and associated tanks on each group. Following URL can be used to receive group information from Tank Scanner NXA820.

#### URL

<GET /GWTHandler.esp?DATATYPE=99&TYPE=GROUPINFO>

#### URL parameters

Key	Value	Required/Optional	Type	Description
DATATYPE	99	Required	Integer	-
TYPE	GROUPINFO	Required	String	-

#### Data parameters

None

#### Success response

- Code: 200 OK
- Content

```
{ "groupTypes": [
  {
    "groupType": 3,
    "name": "Products",
    "groups": [
      {
        "groupId": 805331796,
        "groupName": "p4",
        "tanks": []
      },
      {
        "groupId": 805308828,
        "groupName": "Petrol",
        "tanks": [
          {
            "tankName": "Tank-1",
            "tankId": 1
          }
        ]
      }
    ]
  }
],
{
  "groupType": 1,
```

```
"name": "Static Groups",
"groups": [
  {
    "groupId": 268435458,
    "groupName": "Group 1",
    "tanks": [
      {
        "tankName": "Tank-1",
        "tankId": 1
      },
      {
        "tankName": "Tank-2",
        "tankId": 2
      },
      {
        "tankName": "Tank-3",
        "tankId": 3
      }
    ]
  },
  {
    "groupId": 268435457,
    "groupName": "Group 2",
    "tanks": [
      {
        "tankName": "Tank-1",
        "tankId": 1
      },
      {
        "tankName": "Tank-2",
        "tankId": 2
      }
    ]
  }
],
{
  "groupType": 2,
  "name": "Dynamic Groups",
  "groups": [
    {
      "groupId": 536870913,
```

```

    "groupName": "Dyn Group 3",
    "tanks": [
      {
        "tankName": "Tank-1",
        "tankId": 1
      }
    ]
  }
]
}

```

Here,

Field	Description
groupType	Type of group Here, 1 - Static Groups 2 - Dynamic Groups 3 - Products
name	Group name
groupId	Group identifier
groupName	Group name based on Group type
tankName	Tanks associated to the group
tankId	Tank Identifier

#### Error response

- Code: 404 Not found
- Content  
{Access Error: 404 -- Not Found}

#### Example

<http://10.56.53.33/GWTHandler.esp?DATATYPE=99&TYPE=GROUPINFO>

#### Notes

None

### 4.2.2 Group parameters

After receiving a group identifier, below URL can be called to get group parameter information like value, status, timestamp, unit and assigned product information.

#### URL

<GET /GWTHandler.esp?DATATYPE=99&TYPE=GROUPPARAM&GROUPID=<groupId>>

**URL parameters**

Key	Value	Required/Optional	Type	Description
DATATYPE	99	Required	Integer	-
TYPE	GROUPPARAM	Required	String	-
GROUPLD	<groupid>	Required	Integer	Group id

*Data parameters*

None

**Success response**

- Code: 200 OK
- Content

```
{
  "parameterlist": [
    {
      "id": "628",
      "value": "2323.000",
      "unitType": "13",
      "timestamp": "11:25:22 21/04/2023"
    },
    {
      "id": "717",
      "value": "2610.500",
      "unitType": "9",
      "timestamp": "11:25:22 21/04/2023"
    },
    {
      "id": "718",
      "value": "0.000",
      "unitType": "9",
      "timestamp": "11:25:22 21/04/2023"
    },
    {
      "id": "719",
      "value": "2610.500",
      "unitType": "9",
      "timestamp": "11:25:22 21/04/2023"
    },
    {
      "id": "726",
      "value": "2610.500",
      "unitType": "9",

```

```
    "timestamp": "11:25:22 21/04/2023"
  },
  {
    "id": "727",
    "value": "2610.500",
    "unitType": "9",
    "timestamp": "11:25:22 21/04/2023"
  },
  {
    "id": "728",
    "value": "2610.500",
    "unitType": "9",
    "timestamp": "11:25:22 21/04/2023"
  },
  {
    "id": "729",
    "value": "0.000",
    "unitType": "10",
    "timestamp": "11:25:22 21/04/2023"
  },
  {
    "id": "730",
    "value": "0.000",
    "unitType": "10",
    "timestamp": "11:25:22 21/04/2023"
  },
  {
    "id": "756",
    "value": "0.000",
    "unitType": "10",
    "timestamp": "11:25:22 21/04/2023"
  }
]
}
```

Here,

Field	Description
Id	Parameter identifier Here, 717 - Total Observed Volume 718 - Remaining Capacity 719 - Available Volume 726 - Gross Observed Volume 727 - Gross Std Volume 728 - Net Std Volume 628 - Observed Density 730 - Total Mass 750 - Liquid Mass 756 - Mass in Vapor
Value	- Parameter value in float format - Parameter value is the sum of parameter value for all Tanks in group
unitType	Parameter unit in SI unit. Here, 13 - kg/m <sup>3</sup> 9 - m <sup>3</sup> 10 - kg
statusId	Parameter status Here, 655 - Valid/OK 656 - Fail 657 - Manual 675 - Initialized 676 - No data 677 - Invalid Value 680 - Last valid value 706 - Not calibrated
timestamp	Parameter local time of Tank Scanner NXA820 in '24HH:MM:SS dd/mm/yyyy' format

#### Error response

- Code: 404 Not found
- Content  
{Access Error: 404 -- Not Found}

#### Example

<http://10.56.53.33/GWTHandler.esp?DATATYPE=99&TYPE=GROUPPARAM&GROUPID=268435458>

#### Notes

None



## 5 References

Here are some JSON samples for above mentioned interfaces.

### 5.1 TankInfo.json

```
{
  "deviceName": "TS4",
  "ipAddress": "172.16.40.153",
  "tanks": [{
    "tankId": 1,
    "tankName": "Tank-1",
    "tankStatus": 649
  },
  {
    "tankId": 2,
    "tankName": "Tank-2",
    "tankStatus": 649
  },
  {
    "tankId": 3,
    "tankName": "Tank-3",
    "tankStatus": 649
  },
  {
    "tankId": 4,
    "tankName": "Tank-4",
    "tankStatus": 649
  },
  {
    "tankId": 5,
    "tankName": "Tank-5",
    "tankStatus": 649
  },
  {
    "tankId": 6,
    "tankName": "Tank-6",
    "tankStatus": 649
  },
  {
    "tankId": 7,
    "tankName": "Tank-7",
    "tankStatus": 649
  }
}
```

```
    },  
    {  
      "tankId": 8,  
      "tankName": "Tank-8",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 9,  
      "tankName": "Tank-9",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 10,  
      "tankName": "Tank-10",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 11,  
      "tankName": "Tank-11",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 12,  
      "tankName": "Tank-12",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 13,  
      "tankName": "Tank-13",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 14,  
      "tankName": "Tank-14",  
      "tankStatus": 649  
    },  
    {  
      "tankId": 15,  
      "tankName": "Tank-15",  
      "tankStatus": 649  
    }  
  ]  
}
```

## 5.2 TankParam.json

```
{
  "productName": "No Product",
  "productId": "0",
  "parameterlist": [
    {
      "id": "628",
      "value": "900.000",
      "unitType": "13",
      "statusId": "657",
      "timestamp": "01:00:00 24/03/2020"
    },
    {
      "id": "717",
      "value": "1140.000",
      "unitType": "9",
      "statusId": "655",
      "timestamp": "16:14:10 24/03/2020"
    },
    {
      "id": "718",
      "value": "0.000",
      "unitType": "9",
      "statusId": "655",
      "timestamp": "16:14:10 24/03/2020"
    },
    {
      "id": "719",
      "value": "1140.000",
      "unitType": "9",
      "statusId": "655",
      "timestamp": "16:14:10 24/03/2020"
    },
    {
      "id": "726",
      "value": "1140.000",
      "unitType": "9",
      "statusId": "655",
      "timestamp": "16:14:10 24/03/2020"
    },
    {
      "id": "727",
```

```
    "value": "1140.000",
    "unitType": "9",
    "statusId": "676",
    "timestamp": "16:14:10 24/03/2020"
  },
  {
    "id": "728",
    "value": "1140.000",
    "unitType": "9",
    "statusId": "676",
    "timestamp": "16:14:10 24/03/2020"
  },
  {
    "id": "729",
    "value": "0.000",
    "unitType": "10",
    "statusId": "676",
    "timestamp": "16:14:10 24/03/2020"
  },
  {
    "id": "730",
    "value": "0.000",
    "unitType": "10",
    "statusId": "676",
    "timestamp": "16:14:10 24/03/2020"
  },
  {
    "id": "756",
    "value": "0.000",
    "unitType": "10",
    "statusId": "655",
    "timestamp": "16:14:10 24/03/2020"
  }
]
```

### 5.3 GroupInfo.json

```
{
  "groupTypes": [{
    "groupType": 3,
    "name": "Products",
    "groups": [{
      "groupId": 805359659,
      "groupName": "p1",
      "tanks": [{
        "tankName": "Tank-1",
        "tankId": 1
      }]
    }]
  },
  {
    "groupType": 1,
    "name": "Static Groups",
    "groups": [{
      "groupId": 268435457,
      "groupName": "group1",
      "tanks": [{
        "tankId": 1,
        "tankName": "Tank-1"
      }],
      {
        "tankId": 2,
        "tankName": "Tank-2"
      },
      {
        "tankId": 3,
        "tankName": "Tank-3"
      },
      {
        "tankId": 4,
        "tankName": "Tank-4"
      },
      {
        "tankId": 5,
        "tankName": "Tank-5"
      }
    ]
  }
],
```

```
{
  "groupType": 2,
  "name": "Dynamic Groups",
  "groups": [{
    "groupId": 536870913,
    "groupName": "dg",
    "tanks": [{
      "tankId": 1,
      "tankName": "Tank-1"
    },
    {
      "tankId": 2,
      "tankName": "Tank-2"
    },
    {
      "tankId": 3,
      "tankName": "Tank-3"
    },
    {
      "tankId": 4,
      "tankName": "Tank-4"
    }
  ]
},
{
  "groupId": 536870914,
  "groupName": "dg1",
  "tanks": [{
    "tankId": 1,
    "tankName": "Tank-1"
  },
  {
    "tankId": 2,
    "tankName": "Tank-2"
  }
  ]
}
}
```

## 5.4 GroupParam.json

```
{
  "parameterlist": [
    {
      "id": "628",
      "value": "960.000",
      "unitType": "13",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "717",
      "value": "8090.000",
      "unitType": "9",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "718",
      "value": "0.000",
      "unitType": "9",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "719",
      "value": "8090.000",
      "unitType": "9",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "726",
      "value": "8090.000",
      "unitType": "9",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "727",
      "value": "7764.199",
      "unitType": "9",
      "timestamp": "16:17:56 24/03/2020"
    },
    {
      "id": "728",
      "value": "7764.199",
```

```
    "unitType": "9",
    "timestamp": "16:17:56 24/03/2020"
  },
  {
    "id": "729",
    "value": "7766400.000",
    "unitType": "10",
    "timestamp": "16:17:56 24/03/2020"
  },
  {
    "id": "730",
    "value": "7766400.000",
    "unitType": "10",
    "timestamp": "16:17:56 24/03/2020"
  },
  {
    "id": "756",
    "value": "0.000",
    "unitType": "10",
    "timestamp": "16:17:56 24/03/2020"
  }
]
```











[www.addresses.endress.com](http://www.addresses.endress.com)

---