Find the function you need

Choose a scale function below, then keep its read and control requests together as you work. For example, the tare section covers reading, applying and clearing tare. Each operation shows its HTTP method, endpoint and example. Platform selection belongs in the URL query string for both GET and POST requests.

WEIGHT
Read a weight and interpret its JSON fields.
Page 2

TARE
Read, apply and clear tare.
Page 3

UNITS
Read the current unit and toggle to the next.
Page 4

MODE & ZERO
Read or toggle gross/net mode; press ZERO.
Page 5

DATA DELIVERY
Choose application GET or scale-originated JSON.
Page 6

UNIT STRINGS
Look up unit codes and aliases.
Page 7

EXAMPLE INDICATOR API BASE URL

```
http://192.168.1.123:8080/api/1.0
```

Replace the example IP address and port with the indicator's configured settings. API release **1.0**; no bulk updates. Each request works with one object.

Shared rule	Use it this way
GET / POST	GET reads values. POST executes a supported control. A browser address-bar request uses GET.
Platform selection	Query string only: ?Platform=0 through ?Platform=9. Omit it, or use 0, for the displayed platform; 1 selects platform one.
Case rules	Paths are case-insensitive. Parameter names and JSON field names are case-sensitive: Platform is correct; platform is not.
Request bodies	Do not put Platform in a form-encoded or JSON body. Control examples in this reference send an empty body.

Reading the examples: JSON is structured data with named fields. Example values show the response format; actual values come from the indicator. Read-only URLs can also be opened in a browser.

Weight readings

Use GET /weight to read the scale. The response identifies the numeric Weight, its Unit and Platform, the Tare amount and the gross/net mode. Keep the Gross flag with the reading: true means gross weight; false means net weight. A missing Stable field does not establish stability.

Operation	Method and path	Purpose
Read weight	`GET /weight`	Get the reading for the displayed or selected platform.

READ WEIGHT

GET - PLATFORM CURRENTLY DISPLAYED

```
curl --get "http://192.168.1.123:8080/api/1.0/weight"
```

EXAMPLE JSON RESPONSE

```
{"Gross":false,"Platform":1,"Tare":102.4,"Unit":"oz","Weight":22.78}
```

GET - PLATFORM 1

```
curl --get "http://192.168.1.123:8080/api/1.0/weight?Platform=1"
```

EXAMPLE JSON RESPONSE

```
{"Gross":false,"Platform":1,"Tare":105.75,"Unit":"oz","Weight":22.819}
```

Understand the fields before storing the reading

Field	Type	How to interpret it
`Weight`	`number`	Numeric reading in Unit, interpreted according to Gross.
`Platform`	`integer`	Platform associated with the reading.
`Unit`	`string`	Reported unit, such as lb, g, kg or oz.
`Tare`	`number`	Tare amount in the reported unit.
`Gross`	`boolean`	true: Weight is gross. false: Weight is net. This flag is not a numeric gross-weight field.
`Stable`	`boolean`	Stability flag. If absent, stability is unknown; a workflow requiring stable readings must receive an explicit supported stable indication.

Tare: read, apply & clear

Keep all tare operations in one place. GET /tare reads the tare amount and its unit. A POST to the TARE click endpoint tares off the current load; a POST to the long-click endpoint clears the tare, corresponding to a long press of the on-screen TARE button.

Operation	Method and path	Effect
READ TARE	`GET /tare`	Return Platform, Unit and Tare.
APPLY TARE	`POST /button/tare/click`	Tare off the current load.
CLEAR TARE	`POST /button/tare/longclick`	Clear the tare amount.

1 READ THE TARE AMOUNT

GET - DISPLAYED PLATFORM, THEN PLATFORM 2

```
curl --get "http://192.168.1.123:8080/api/1.0/tare"
{"Platform":1,"Unit":"oz","Tare":100.123}

curl --get "http://192.168.1.123:8080/api/1.0/tare?Platform=2"
{"Platform":2,"Unit":"g","Tare":10.12}
```

2 APPLY TARE

This simulates pressing TARE for the platform currently displayed.

POST - APPLY TARE

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/tare/click"
```

3 CLEAR TARE

This simulates a long press of TARE and clears the tare amount.

POST - CLEAR TARE

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/tare/longclick"
```

Selecting a platform
For either POST action, append ?Platform=n to the endpoint to select the platform. Platform is read from the query string only. Omit it to use the platform currently displayed. The reference specifies no control response payload.

Units: read & toggle

Read the current unit with GET /unit. To change it, send POST to the UNIT button endpoint, which toggles to the next unit on the indicator. This action does not directly set an arbitrary unit string. The unit-code reference on page 7 lists the documented strings and aliases.

Operation	Method and path	Effect
READ UNIT	`GET /unit`	Return Platform and Unit.
NEXT UNIT	`POST /button/unit/click`	Toggle to the next unit.

1 READ THE CURRENT UNIT

GET - DISPLAYED PLATFORM, THEN PLATFORM 2

```
curl --get "http://192.168.1.123:8080/api/1.0/unit"
{"Platform":1,"Unit":"oz"}

curl --get "http://192.168.1.123:8080/api/1.0/unit?Platform=2"
{"Platform":2,"Unit":"g"}
```

2 TOGGLE TO THE NEXT UNIT

Send an explicit POST request. Opening this endpoint in a browser address bar uses GET instead.

POST - PLATFORM CURRENTLY DISPLAYED

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/unit/click"
```

POST - PLATFORM 2

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/unit/click?Platform=2"
```

Platform stays in the URL
The selected-platform example uses ?Platform=2 in the query string. It sends no request body. Do not move Platform into a form or JSON body.

Response: the control endpoint has no specified response payload. Use GET /unit to read the current unit.

Unit strings and aliases: page 7

Gross/net mode & zero

Use the gross/net functions to read or toggle the indicator's weight mode. GET /gross returns a boolean flag, while GET /weight returns the numeric reading in that mode. ZERO is a separate button action. The operations below keep the displayed or selected platform as their target.

Operation	Method and path	Effect
READ MODE	`GET /gross`	Return Platform and Gross.
TOGGLE MODE	`POST /button/gross/click`	Toggle gross/net mode.
ZERO	`POST /button/zero/click`	Simulate pressing ZERO.

GROSS / NET MODE

GET - DISPLAYED PLATFORM, THEN PLATFORM 2

```
curl --get "http://192.168.1.123:8080/api/1.0/gross"
{"Platform":1,"Gross":true}

curl --get "http://192.168.1.123:8080/api/1.0/gross?Platform=2"
{"Platform":2,"Gross":false}
```

Gross = true: gross mode. **Gross = false:** net mode. The flag itself is not a numeric weight.

POST - TOGGLE GROSS / NET

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/gross/click"
```

ZERO THE SCALE

Send POST to simulate pressing the on-screen ZERO button. This is separate from the tare and gross/net actions.

POST - ZERO

```
curl -X POST "http://192.168.1.123:8080/api/1.0/button/zero/click"
```

Platform selection and responses
Append ?Platform=n to a POST endpoint to select a platform, or omit it for the displayed platform. The parameter belongs in the query string only. These button endpoints have no specified response payload.

Collect readings: GET or JSON push

Choose the collection direction that fits the application. With GET, your software requests a reading from the indicator. With outbound JSON, the scale sends a reading to a configured receiver. Both remain supported. Remote controls are separate POST requests sent by the application to the indicator, rather than weight deliveries to the receiver.

APPLICATION REQUESTS THE READING - GET
The application sends GET to the indicator API address and port. The indicator returns JSON. Make an individual request when a reading is needed, or poll repeatedly.

SCALE SENDS THE READING - OUTBOUND JSON
The scale sends HTTP POST JSON to the configured receiver, periodically or through PRINT. The receiving application accepts the data and handles storage or its downstream workflow.

Question	Application GET	Scale-originated JSON
Who initiates it?	Application	Scale
Request destination	Indicator API address and port	Configured receiving URL
Trigger	On demand or repeated polling	Configured interval or PRINT
Where is the reading?	JSON response	JSON request body sent to receiver

Two different network destinations

The indicator API and the receiving application have separate addresses and listening ports. Use the configured settings for each. The address and port 8080 in the command examples are examples, not fixed defaults. Indicator API examples use HTTP.

Common integration questions

Can I continue using GET for weights?
Yes. The GET resources remain valid for individual requests and repeated polling.

Where does Platform go on POST controls?
In the URL query string only, for example button/unit/click?Platform=2. It is not read from the request body.

Does a control POST deliver a weight to my receiver?
No. Controls go to the indicator. Outbound JSON delivery goes from the scale to its configured receiver.

Setup and demonstration: For device topology, configuration and ScaleAPIMonitor, see arlynscales.com/industrial-scale-api/.

Unit strings & common terms

Use this list when interpreting the Unit field returned with a reading or tare amount. Unit strings are case-insensitive. These codes describe the reported units; they are not a separate command for choosing a unit. Read the current setting or toggle the UNIT button using the operations on page 4.

Group	Unit	Listed strings / aliases
Main default units	Pounds	`lb, pound, pounds`
Main default units	Kilograms	`kg, kilogram, kilograms`
Main default units	Grams	`g, gram, grams`
Main default units	Ounces	`oz, ounce, ounces`
Other units	Troy ounces	`otz`
Other units	Pennyweights	`pwt`
Other units	Grains	`gr`
Other units	Gallons	`gal, gallon, gallons`
Other units	Percentage	`perc`
Other units	Newtons	`n, N`
Other units	Jinn	`jn`

Terms used in the examples

Term	Meaning
Query string	The URL portion after ?, such as ?Platform=2. This is where platform selection belongs.
JSON	Structured data with named fields. Preserve field capitalization and value types.
Polling	The application repeatedly requests readings with GET.

Reference scope
API release 1.0: four read resources and five control actions, together with their response fields, platform-selection rules and unit strings. The integration guide provides the longer setup tutorial.