

TALLO

API REST

Guía de integración para desarrolladores

Control de válvulas y consulta de telemetría de dispositivos

Versión de la API	v1.0
URL base	https://portal.tallo.tech/api/v1
Autenticación	HTTP Basic Authentication sobre HTTPS
Fecha del documento	27 de julio de 2026

Documentación técnica de integración

Control del documento

Elemento	Valor
Documento	Tallo API REST - Guía de integración para desarrolladores
Versión	1.0
Versión de la API	v1
Estado	Guía de integración
URL base	https://portal.tallo.tech/api/v1
Autenticación	HTTP Basic Authentication
Formato de respuesta	JSON, UTF-8

Objetivo

Este documento describe cómo integrar aplicaciones y servicios con la API REST de dispositivos Tallo. Está dirigido a desarrolladores backend, frontend, aplicaciones móviles, integradores IoT y equipos de QA que necesiten consultar el estado de un dispositivo o enviar órdenes a sus válvulas.

Requisito de seguridad: Todas las peticiones en producción deben realizarse mediante HTTPS. HTTP Basic Authentication depende de TLS/HTTPS para proteger las credenciales durante el transporte.

Contenido

- 1. Descripción general de la API
- 2. Autenticación
- 3. Convenciones de petición y respuesta
- 4. Referencia de endpoints
- 5. Consultar estado de válvula - GET
- 6. Enviar orden a válvula - PUT
- 7. Consultar contador - GET
- 8. Consultar batería - GET
- 9. Errores y códigos HTTP
- 10. Ejemplos de integración
- 11. Recomendaciones y checklist de integración

1. Descripción general de la API

La API expone recursos asociados a cada dispositivo mediante rutas REST versionadas. Cada petición identifica el dispositivo objetivo por su número de serie (SN). La versión actual permite consultar y controlar hasta cuatro canales de válvula, consultar el contador acumulado y obtener el nivel de batería.

Operación	Método	Endpoint	Descripción
Consultar válvula	GET	/devices/{SN}/valves/{channel}	Obtiene el estado actual de un canal
Controlar válvula	PUT	/devices/{SN}/valves/{channel}	Solicita el estado on/off de un canal
Consultar contador	GET	/devices/{SN}/counter	Obtiene el valor acumulado del contador
Consultar batería	GET	/devices/{SN}/battery	Obtiene el nivel de batería en porcentaje

1.1 URL base

```
https://portal.tallo.tech/api/v1
```

Todos los endpoints descritos en esta guía son relativos a esta URL base.

1.2 Identificador del dispositivo

El parámetro de ruta {SN} identifica el dispositivo. Debe estar formado únicamente por dígitos. Ejemplo:

```
868333034802583
```

1.3 Canales de válvula

Los endpoints de válvula admiten los canales 1, 2, 3 y 4. Cualquier otro valor será rechazado con HTTP 400.

2. Autenticación

Todas las peticiones requieren HTTP Basic Authentication. El usuario y la contraseña deben ser credenciales válidas para el dispositivo indicado mediante el SN de la URL.

```
Authorization: Basic base64(username:password)
```

La mayoría de clientes HTTP generan esta cabecera automáticamente. Por ejemplo, con cURL se utiliza la opción -u:

```
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/868333034802583/counter"
```

Campo	Descripción
username	Usuario autorizado para acceder al dispositivo.
password	Contraseña asociada al usuario.
SN	Número de serie del dispositivo solicitado en la URL.

Ámbito de autenticación: Las credenciales deben ser válidas para el dispositivo indicado por el SN. Si no son correctas, la API devuelve HTTP 401.

3. Convenciones de petición y respuesta

3.1 Tipo de contenido

Las respuestas se devuelven en JSON codificado en UTF-8. Las peticiones PUT que incluyen un cuerpo JSON deben enviar Content-Type: application/json.

```
Content-Type: application/json
Accept: application/json
```

3.2 Estructura de una respuesta correcta

```
{
  "ok": true,
  "data": {
    "...": "campos específicos del endpoint"
  }
}
```

3.3 Estructura de una respuesta de error

```
{
  "ok": false,
  "error": {
    "code": "ERROR_CODE",
    "message": "Descripción del error",
    "details": {}
  }
}
```

El campo details es opcional y únicamente aparece cuando la API proporciona información adicional de validación.

3.4 Tipos de datos y nomenclatura

Campo	Tipo	Descripción
ok	boolean	true si la operación fue correcta; false si se produjo un error
SN	string	Número de serie del dispositivo
channel	integer	Canal de válvula: 1, 2, 3 o 4
status	string	Estado de válvula: on u off
value	number/string	Valor devuelto o solicitado, según el endpoint
unit	string	m3 para contador; % para batería
command	string	Nombre lógico de la operación realizada

3.5 Caché

La API envía Cache-Control: no-store. Las aplicaciones cliente deben tratar las respuestas como datos operativos actuales y no depender de cachés intermedias.

4. Referencia de endpoints

Convención de ejemplos: Los ejemplos utilizan el SN 868333034802583 únicamente a modo ilustrativo. Debe sustituirse por el número de serie real del dispositivo.

5. Consultar estado de válvula

GET /devices/{SN}/valves/{channel}

Devuelve el estado actual del canal de válvula solicitado.

Parámetro	Ubicación	Obligatorio	Valores admitidos
SN	Ruta	Sí	Solo dígitos
channel	Ruta	Sí	1, 2, 3, 4

Ejemplo de petición

```
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/868333034802583/valves/2"
```

Respuesta correcta - 200 OK

```
{
  "ok": true,
  "data": {
    "command": "get_valve",
    "SN": "868333034802583",
    "channel": 2,
    "status": "on"
  }
}
```

Campo	Descripción
status	Estado actual de la válvula. Puede ser on u off.

6. Enviar orden a una válvula

PUT /devices/{SN}/valves/{channel}

Solicita un nuevo estado para el canal de válvula indicado.

Parámetro	Ubicación	Obligatorio	Valores admitidos
SN	Ruta	Sí	Solo dígitos
channel	Ruta	Sí	1, 2, 3, 4
value	Cuerpo JSON	Sí	on, off

Cuerpo de la petición

```
{
  "value": "on"
}
```

Ejemplo con cURL

```
curl -u "USUARIO:CONTRASEÑA" \
-X PUT \
-H "Content-Type: application/json" \
-d '{"value":"on"}' \
"https://portal.tallo.tech/api/v1/devices/868333034802583/valves/2"
```

Respuesta correcta - 200 OK

```
{
  "ok": true,
  "data": {
    "command": "set_valve",
    "SN": "868333034802583",
    "channel": 2,
    "value": "on"
  }
}
```

Confirmación de estado: Una respuesta correcta confirma que la API ha aceptado la orden. Si la aplicación necesita confirmar el estado real de la válvula, debe consultar posteriormente el endpoint GET de ese canal.

7. Consultar contador

GET /devices/{SN}/counter

Devuelve el valor acumulado del contador asociado al dispositivo, expresado en metros cúbicos.

Ejemplo de petición

```
curl -u "USUARIO:CONTRASEÑA" \
"https://portal.tallo.tech/api/v1/devices/868333034802583/counter"
```

Respuesta correcta - 200 OK

```
{
  "ok": true,
  "data": {
    "command": "get_counter",
    "SN": "868333034802583",
    "value": 123.456,
    "unit": "m3"
  }
}
```

Campo	Descripción
value	Lectura acumulada del contador.
unit	Unidad del valor. En API v1: m3.

8. Consultar batería

GET /devices/{SN}/battery

Devuelve el nivel de batería del dispositivo expresado como porcentaje entero entre 0 y 100.

Ejemplo de petición

```
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/868333034802583/battery"
```

Respuesta correcta - 200 OK

```
{
  "ok": true,
  "data": {
    "command": "get_battery",
    "SN": "868333034802583",
    "value": 78,
    "unit": "%"
  }
}
```

Campo	Descripción
value	Nivel de batería, entre 0 y 100.
unit	Unidad del valor. Siempre %.

9. Errores y códigos HTTP

La aplicación cliente debe evaluar tanto el código de estado HTTP como el valor JSON error.code. Se recomienda utilizar error.code para la lógica de programa y reservar error.message para diagnóstico o presentación al usuario.

HTTP	Código de error	Descripción
400	INVALID_SERIAL_NUMBER	El SN está vacío o contiene caracteres no numéricos.
400	INVALID_CHANNEL	El canal no es un entero entre 1 y 4.
400	INVALID_JSON	El cuerpo de una petición PUT no contiene JSON válido.
400	MISSING_PARAMETER	Falta un parámetro obligatorio.
400	INVALID_VALUE	El valor de válvula no es on ni off.
401	AUTHENTICATION_REQUIRED	No se han enviado credenciales HTTP Basic.
401	INVALID_CREDENTIALS	El usuario o la contraseña no son válidos para el dispositivo.
404	DEVICE_NOT_FOUND	No existe un dispositivo con el SN solicitado.
404	ENDPOINT_NOT_FOUND	El endpoint solicitado no existe.
405	METHOD_NOT_ALLOWED	El método HTTP no está permitido para ese endpoint.
500	DATABASE_CONNECTION_ERROR	No ha sido posible acceder al servicio de datos.
500	DATABASE_ERROR	Se ha producido un error interno al procesar los datos.
500	INVALID_DATABASE_VALUE	El dispositivo contiene un valor interno no válido.

9.1 Autenticación requerida

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Valve API", charset="UTF-8"

{
  "ok": false,
  "error": {
    "code": "AUTHENTICATION_REQUIRED",
    "message": "Se requiere autenticación HTTP Basic."
  }
}
```

9.2 Canal incorrecto

```
{
  "ok": false,
  "error": {
    "code": "INVALID_CHANNEL",
    "message": "El parámetro channel debe estar entre 1 y 4.",
    "details": {
      "allowed_values": [1, 2, 3, 4]
    }
  }
}
```

9.3 Dispositivo no encontrado

```
{
  "ok": false,
  "error": {
    "code": "DEVICE_NOT_FOUND",
    "message": "No se ha encontrado ningún dispositivo con el número de serie indicado."
  }
}
```


10. Ejemplos de integración

Los ejemplos siguientes utilizan marcadores para las credenciales. No se recomienda incluir contraseñas de producción directamente en el código fuente. Utilice variables de entorno, gestores de secretos o mecanismos equivalentes de almacenamiento seguro.

10.1 JavaScript - fetch

```
const baseUrl = "https://portal.tallo.tech/api/v1";
const sn = "868333034802583";
const username = process.env.API_USER;
const password = process.env.API_PASSWORD;
const auth = btoa(`${username}:${password}`);

const response = await fetch(`${baseUrl}/devices/${sn}/battery`, {
  headers: {
    "Authorization": `Basic ${auth}`,
    "Accept": "application/json"
  }
});

const payload = await response.json();
if (!response.ok) {
  throw new Error(`${payload.error.code}: ${payload.error.message}`);
}
console.log(payload.data.value, payload.data.unit);
```

10.2 Python - requests

```
import os
import requests

base_url = "https://portal.tallo.tech/api/v1"
sn = "868333034802583"
auth = (os.environ["API_USER"], os.environ["API_PASSWORD"])

r = requests.get(f"{base_url}/devices/{sn}/counter", auth=auth, timeout=10)
r.raise_for_status()
data = r.json()["data"]
print(data["value"], data["unit"])
```

10.3 PHP - cURL

```
<?php
$baseUrl = 'https://portal.tallo.tech/api/v1';
$sn = '868333034802583';
$user = getenv('API_USER');
$pass = getenv('API_PASSWORD');

$ch = curl_init($baseUrl . '/devices/' . $sn . '/valves/1');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERPWD, $user . ':' . $pass);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Accept: application/json'));

$body = curl_exec($ch);
$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

$data = json_decode($body, true);
?>
```

10.4 PUT de válvula - JavaScript

```
const response = await fetch(
  `${baseUrl}/devices/${sn}/valves/2`,
  {
    method: "PUT",
    headers: {
      "Authorization": `Basic ${auth}`,
      "Content-Type": "application/json",
      "Accept": "application/json"
    },
    body: JSON.stringify({ value: "on" })
  }
);
```

11. Recomendaciones y checklist de integración

11.1 Comportamiento recomendado del cliente

- Utilizar HTTPS en todas las peticiones.
- Configurar tiempos de espera razonables y gestionar posibles fallos de red.
- Tratar las respuestas 4xx como errores de autenticación, validación o petición, y las 5xx como errores del servicio.
- No asumir que un PUT correcto implica que la válvula física ya ha cambiado de estado; consultar GET cuando sea necesaria confirmación.
- No almacenar en caché las consultas operativas salvo que la aplicación acepte explícitamente datos no actualizados.
- Mantener las credenciales fuera del código fuente y de los logs.
- Validar el SN y el canal antes de realizar la petición.
- Utilizar error.code para gestionar errores de forma programática.

11.2 Checklist rápida de integración

- URL base configurada: <https://portal.tallo.tech/api/v1>
- Usuario y contraseña HTTP Basic almacenados de forma segura
- SN del dispositivo conocido y numérico
- Consulta GET de válvula validada para los canales utilizados
- Orden PUT validada con valores on y off
- Respuesta del contador interpretada con unit = m3
- Respuesta de batería interpretada con unit = %
- Errores HTTP 400, 401 y 404 gestionados
- Política de timeout/reintentos definida por la aplicación integradora

Versionado: Las integraciones deben incluir /v1 en la URL base. De este modo, futuras versiones incompatibles podrán publicarse bajo una nueva ruta sin modificar el comportamiento documentado aquí.

Anexo A - Referencia rápida

Método	Ruta	Cuerpo	Datos de respuesta
GET	/devices/{SN}/valves/{channel}	-	command, SN, channel, status
PUT	/devices/{SN}/valves/{channel}	{"value":"on off"}	command, SN, channel, value
GET	/devices/{SN}/counter	-	command, SN, value, unit=m3
GET	/devices/{SN}/battery	-	command, SN, value, unit=%

Anexo B - Prueba rápida

Sustituya USUARIO, CONTRASEÑA y SN por los valores correspondientes antes de ejecutar los comandos.

```
# 1. Batería
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/SN/battery"

# 2. Contador
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/SN/counter"

# 3. Estado de válvula
curl -u "USUARIO:CONTRASEÑA" \
  "https://portal.tallo.tech/api/v1/devices/SN/valves/1"

# 4. Orden de válvula
curl -u "USUARIO:CONTRASEÑA" -X PUT \
  -H "Content-Type: application/json" \
  -d '{"value":"on"}' \
  "https://portal.tallo.tech/api/v1/devices/SN/valves/1"
```

Fin del documento